TEAM 2025105

Index:

Problem Statement:

In today's automobile-centric world, parking can be a real headache. Whether in crowded urban areas or busy parking lots, finding enough space to maneuver a vehicle into a spot often becomes a frustrating challenge. Traditional parallel parking requires a series of precise forward and backward movements, which can be both time-consuming and stressful.

Once parked, things don't necessarily get easier. In tight, crowded lots, simply getting in and out of the car can feel like a gymnastic feat. Squeezing through narrow gaps between vehicles is not just inconvenient—it can also lead to accidental scrapes and dents. Plus, the risk of collisions when pulling out of a tight spot is all too common.

Existing Solutions:

Crab-walking has been implemented in various cars which allow diagonal maneuvering.
However, most of the currently existing solutions rely on sensors which may prove to be erroneous in detecting obstacles and may malfunction on cloudy days. Sensors will also increase the latency of the system. The system's slow operation and close proximity to other vehicles during parking maneuvers can be unsettling. Also, the steps involved in switching to Parallel Parking mode are not that user friendly and may prove to be a hassle for people who are not tech savvy. Additionally, the expensive engineering involved increases the production cost and makes it unsuitable for mass production.

Our Solution:

Our solution is an implementation of a four-wheel drive car. The wheels are controlled using the DC motors. The DC motor mounts are connected to the servos mounted on the car. The servos provide direction to the movement of the car. When operated in 'Normal Mode', the model functions just like a normal car would. When switched to 'Parallel Parking' mode, the model seamlessly transitions into the parallel parking setup, the alignment being controlled by the servos. In 'Parallel Parking' mode, the model can move parallelly from side to side, making parallel parking much more efficient. This feature enables the driver to park in tight spaces very easily, something that can prove to be very difficult or sometimes completely impossible.

Components Used:

1. **Arduino Uno Microcontroller**

- Microcontroller**:** ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Digital I/O Pins: 14 (6 can be used as PWM outputs)
- Analog Input Pins: 6
- PWM Output Pins: 6
- Clock Speed: 16 MHz

One Arduino Uno is mounted on the car and acts as the receiver. It controls all the functionalities of the car such as controlling the servos and DC motors.

The other Arduino Uno is used to build a handheld transmitter to transmit different signals for the operations we want to perform.

2. **nRF24L01 transceiver module**

The nRF24L01 is a low-power, 2.4 GHz ISM band wireless transceiver module that enables communication between devices in wireless networks.

- Operating Voltage: 1.9V to 3.6V
- Operating Frequency: 2.4 GHz ISM band
- Data Rate: 250 Kbps, 1 Mbps, and 2 Mbps
- Range: Up to 100 meters
- SPI Interface: For communication with microcontrollers.

One nRF24L01 transceiver module is mounted on the car and acts as a receiver while the other is present in the handheld controller and acts as a transmitter.
The transmitter transmits different signals wirelessly, based on the input and the receiver captures these signals to retrieve the given input.

3. 12 V DC Motor

The 4 DC motors rotate the 4 wheels of the car.

- Speed: 200 RPM
- Voltage: 12V
- Torque: 0.5 kg-cm

4. Servo MG946R

The MG946R is a high-torque, metal-gear servo motor commonly used in robotics, remote-controlled vehicles, and other mechanical systems requiring precise and strong actuation.

- Operating Voltage: 4.8V to 6.0V
- Operating Current: 250 mA (at 4.8V)
- Dimensions: 40.7 mm x 19.7 mm x 42.9 mm
- Control Signal: PWM (Pulse Width Modulation)

The 4 servos control the alignment of the 4 DC motors. This helps us effectively implement normal steering operations as seen in a car as well as the feature of parallel parking.

5. **L298n Motor Driver**

- Operating Voltage: 4.5V to 46V
- Motor Output Voltage: 4.5V to 46V
- Motor Output Current: Up to 2A per channel (continuous)
- Peak Current: 3A per channel
- Logic Voltage: 5V

The L298N is a dual H-bridge motor driver IC, widely used to control DC motors, stepper motors, and other inductive loads. It allows for bidirectional control and is commonly used in robotics and automation.
The L298n motor driver is mounted on the car and controls the speed and direction of rotation of the 4 DC motors.

6. **Potentiometer**

The potentiometer effectively acts as a steering mechanism for the car. It is mounted on the handheld transmitter and helps us to steer the car in driving mode.

7. **Push buttons**

The 3 push buttons are mounted on the transmitter and help us either to move the car in forward(left) and backward(right) directions or to switch between the driving and parallel parking modes.

8. **LM2596 Buck Converter**

- Input voltage: 3-40V
- Output voltage: 1.5-35V(Adjustable)
- Output current: Rated current is 2A, maximum 3A

9. Battery Holder

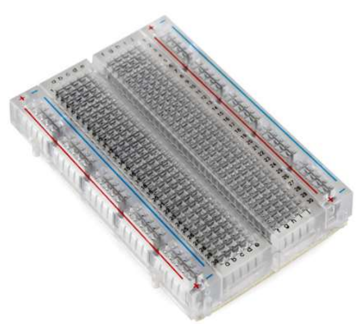
The battery holder combines the Li ion cells into a single power unit that helps manage space on the car.

10. 18650 Li ion cells

- Minimum capacity: 2200mah
- Typical capacity: 2000-2200mah
- Internal impedance: <1 ohm
- Standard voltage: 3.7V

4 Li ion cells are used to power the different components on the car. The cells, in conjunction with the buck convertor, provide different DC voltage levels required for the proper functioning of the different components of the car such as the Arduino Uno(microcontroller on receiver side), the L298n motor driver, the nRF24L01 transceiver module(receiver) and the 4 servos.
2 Li ion cells are used in the handheld transmitter to power the different transmitter components such as the Arduino Uno(microcontroller on the transmitter side), the nRF24L01 transceiver module(transmitter) and the potentiometer.

11. Breadboard



The breadboard serves as a base to integrate the different components required for the handheld transmitter.

12. Connector Terminal

Terminals are used to connect wires between breadboard(on the vehicle) and components like the Arduino Uno, servo motors and buck converter.

13. OLED Display

- Driver IC: SSD1306
- Resolution: 128 x 64
- Visual Angle: >160°
- Input Voltage: 3.3V ~ 6V

An OLED Display module is mounted on the handheld transmitter to show the transmitted information like Pot values, Forward/Backward movement, Left/Right movement and mode of operation.

14. Wheels

10mm diameter wheels have been used for the car. They have been attached to the 4 DC motors.

15. Wires

Jumper wires, breadboard wires and single stranded aluminum wires are used to connect various components.

16. 6mm Plywood

6mm thick plywood has been used to build the chassis for the model.

17. Springs

- Length (Hole to Hole): 60mm
- Width: 8mm

The springs are the main component of the suspension system which provides a smoother ride for the car. The springs are light, yet strong, providing adequate strength, and also damping, perfect for our application. Another benefit of our chosen spring is that it allowed for pre tensioning, allowing us to adjust the initial tension as per our requirement.

Mechanical Design:

The mechanical assembly of the car consisted mainly of two components, the wheel assembly and the chassis.

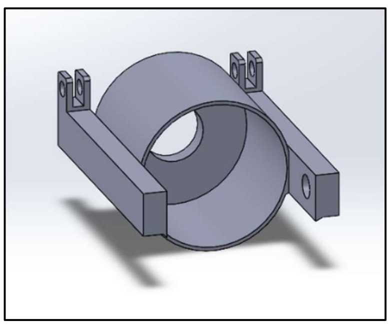
Servo motor - DC motor attachment:

This is the main component which differentiates our wheel assembly from that of a normal car. It comprised a total of 3 parts. The DC motor attachment, the servo motor attachment, and a connecting cylinder. All the components of this assembly were manufactured using 3D printing.

3D printing was chosen as the choice for the manufacturing of this material because of the following reasons:

- Good strength to weight ratio
- Easy to manufacture complex parts and geometries
- Allows for rapid prototyping

DC motor attachment:

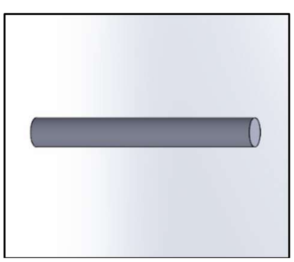


This part's use is to act as a mount for the DC motor. The motor is inserted into the piece, a hole on the opposite side allows the shaft of the motor to come out. The motor is fastened using a nut attached to the part. Further, one end of the suspension system is attached to this part.

Design Considerations:

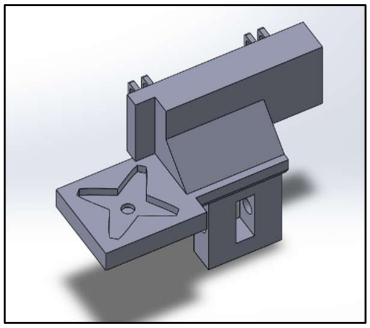
- The part should allow easy attachment and fastening of the DC motor
- It should have strong spring attachment points to handle the springs forces
- The weight of the car is transferred to the wheels using this part and the suspension, so it has to be strong enough to handle these forces.

Connecting Cylinder:



This was a simple component to connect the DC motor attachment and servo motor attachment. A cylinder was chosen to allow free rotation of the DC motor attachment about that axis, which is required for a proper suspension system.

Servo Motor Attachment:

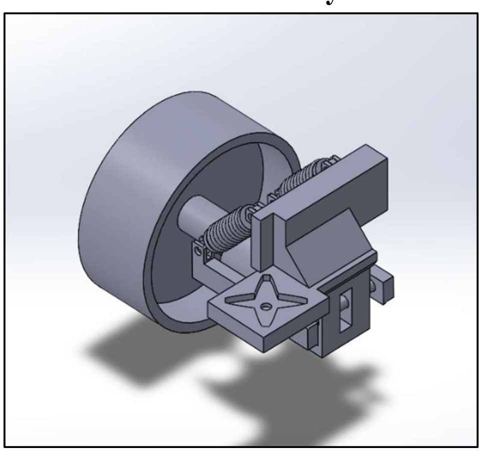


This part serves multiple purposes. Firstly, it contains a cross shaped hole for the attachment of the servo motor's horn to the rest of the wheel assembly. Secondly, the other end of the suspension system is attached to it. The geometry of this part was extremely complex, and it would be virtually impossible to manufacture without 3D printing.

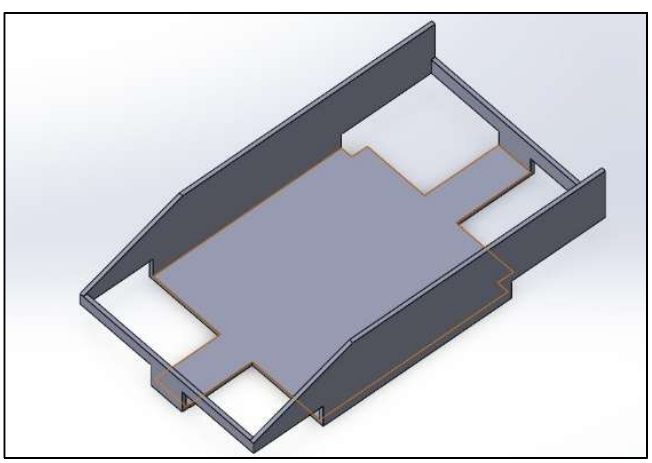
Design considerations:

- The part should allow easy attachment and fastening of the servo motor.
- It should have strong spring attachment points to handle the springs forces.
- It needs to be able to handle the torsional forces caused due to the servo motor.

The entire wheel assembly once assembled:



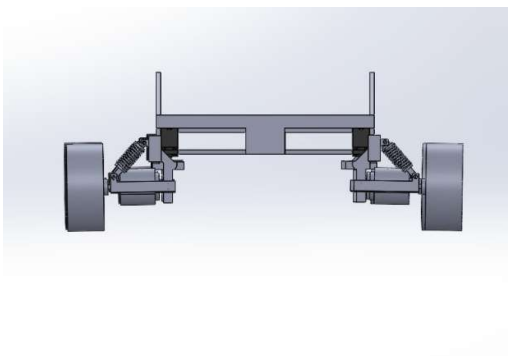
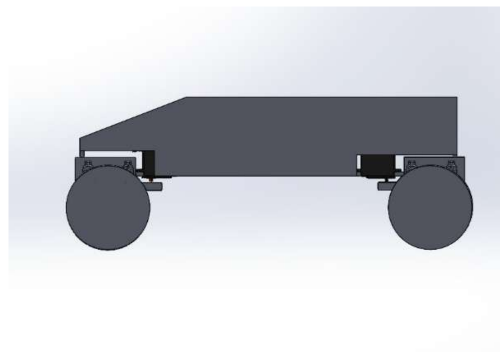
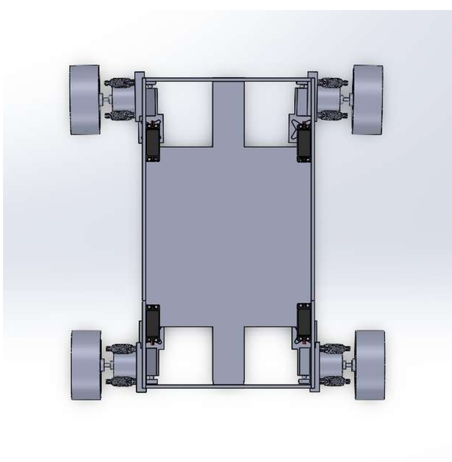
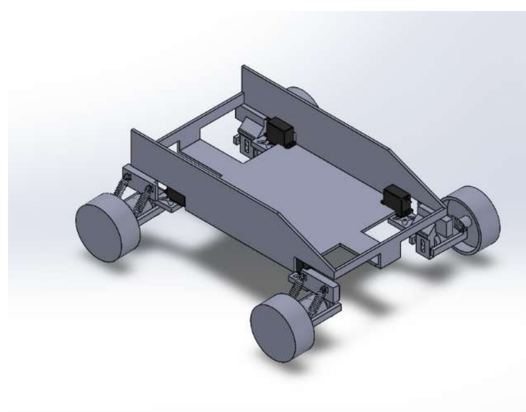
Chassis:



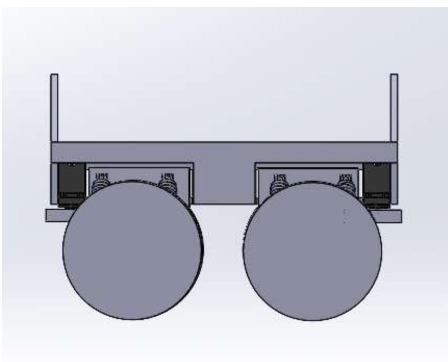
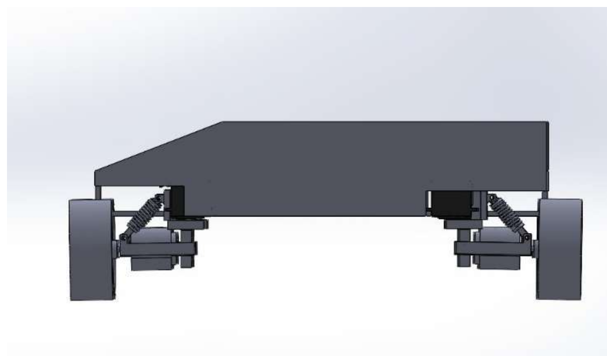
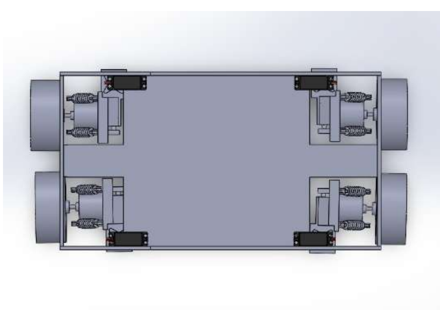
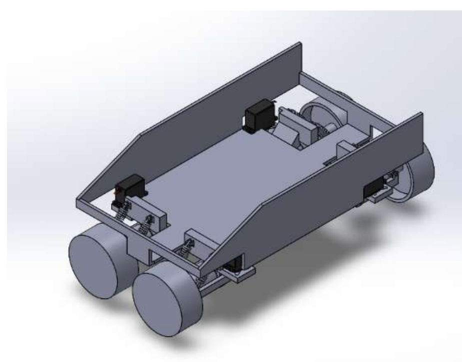
The chassis of the car was made using 6mm thick plywood. To try and be as accurate as possible the dimensions were scaled down like wheelbase, track width etc. of a Maruti Alto, to get the dimensions of our model. We achieved optimal weight distribution by placing components as symmetrically as possible, especially the heavier components like the battery pack.

Full CAD Design:

Normal Mode:

 Front View	 Side View
 Top View	 Bird View

Parallel Parking Mode:

 Front View	 Side View
 Top View	 Bird View

Electronic Architecture:

1. **Four Wheel Drive:**
   A four-wheel drive system has been opted for to ensure consistent power delivery to all wheels, essential for both normal driving and parallel parking. In regular mode, traction and stability have been enhanced, allowing the vehicle to handle diverse terrains.
   In parallel parking mode, smooth lateral movement has been enabled by the 4WD system, as all four wheels have been made to work together to create a crab-like motion. This allows maneuvering in tight spaces to be carried out more quickly and easily.
2. **Ackermann Steering:**
   Ackermann steering has been implemented for the normal driving mode because maneuverability and control during typical driving scenarios have been significantly enhanced by it. Since efficient operation on roads is required when the vehicle is not in parallel parking mode, smoother and more precise turns have been provided by Ackermann steering. This is crucial for ensuring stability and minimizing tire friction during standard movement.
   While Ackermann steering is not utilized during the parallel parking mode (where lateral, crab-like motion is used), it has been retained as an essential component for regular driving to allow efficient navigation on roads and in non-parking situations.
3. **Mode Switching:**
   Two modes have been incorporated into the system – normal driving mode and parallel parking mode. In normal driving mode, steering input is taken from a potentiometer, and direction input (forward or backward) is taken from two buttons, all of which have been mounted on the handheld transmitter.
   A third button has been provided to toggle the system into "parallel parking" mode. In this mode, the wheels are rotated and fixed in a parallel orientation using servo motors. Steering input from the potentiometer is no longer used, and the same two buttons are utilized to move the vehicle left or right.
   The transition between the two modes has been made smooth and slow to prevent stress or damage to the components from prolonged use.
4. **Wireless Control Communication:**
   The system has been divided into a transmitter side and a receiver side, between which communication is carried out wirelessly via an nRF24L01 module on either side.

a. Transmitter:

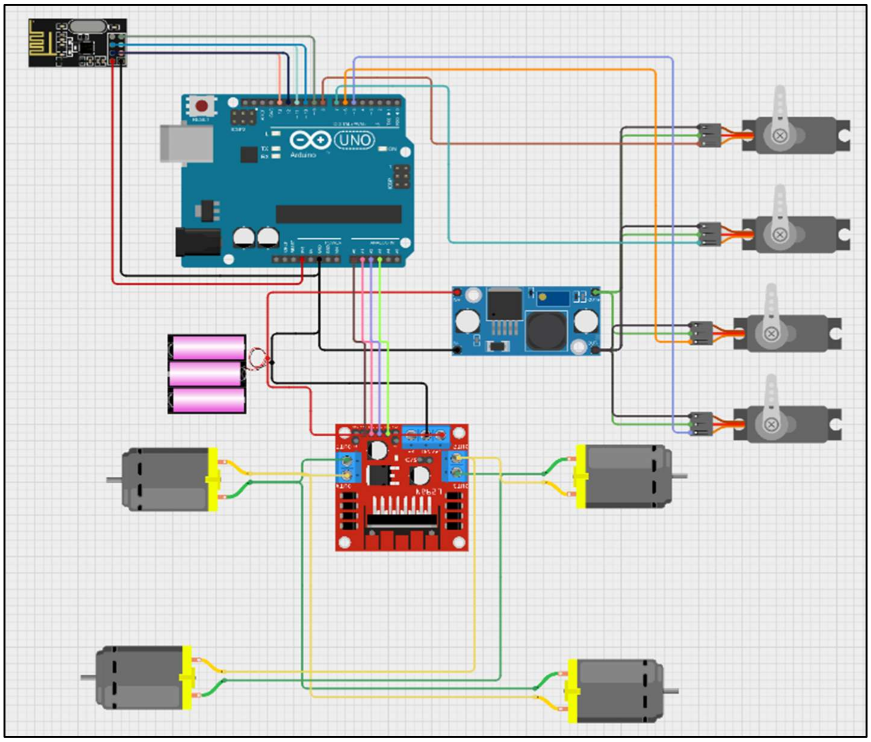
Inputs are taken on this side and have been communicated wirelessly to the receiver side through a radio channel. A potentiometer has been used to indicate the steering angle. Three buttons have been provided – one for changing the driving mode and two for powering the DC motors (for forward/backward or left/right movement). When the buttons are pressed, the required signals are sent by the Arduino and nRF module.

b. Receiver:

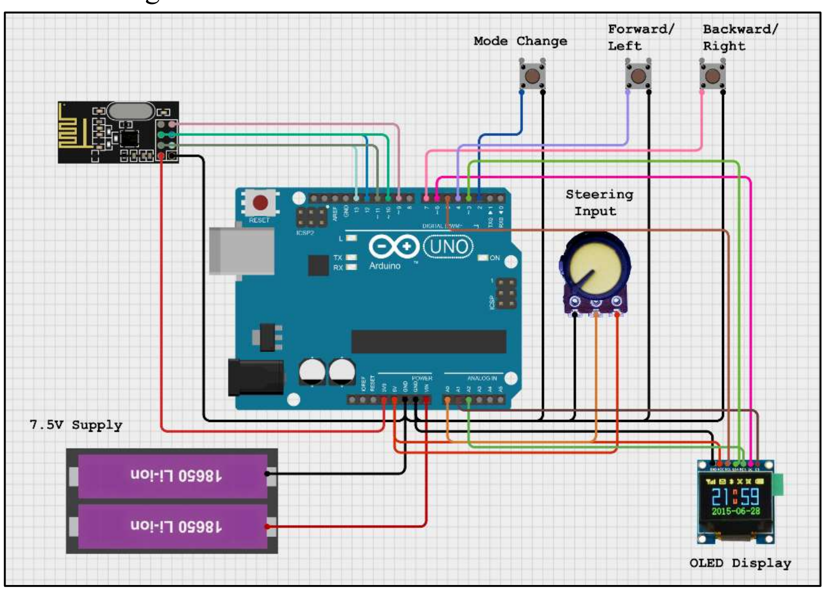
Encoded messages from the transmitter are received by the receiver Arduino and nRF module. The Arduino has been connected to an L298N motor driver and four servo motors. A buck converter, which has been connected to the 12V battery, is used to scale the voltage down to 5V. The motor driver is powered using the 12V supply, and the servo motors are powered using the 5V supply.

Circuit Diagrams:

1. Receiver Circuit Diagram:



2. Transmitter circuit diagram:



Future Applications:

Enhanced U-Turn Capability

Traditional vehicles rely solely on front-wheel steering for U-turns, which requires a small radius of curvature and significant road space. This makes U-turns challenging in narrow or congested areas.

In our design, all four wheels can be turned to specific angles and powered independently. This allows the car to rotate about its own center, effectively spinning in place. The result is a true zero-radius turn, enabling the vehicle to make U-turns with minimal space and time, significantly improving maneuverability.

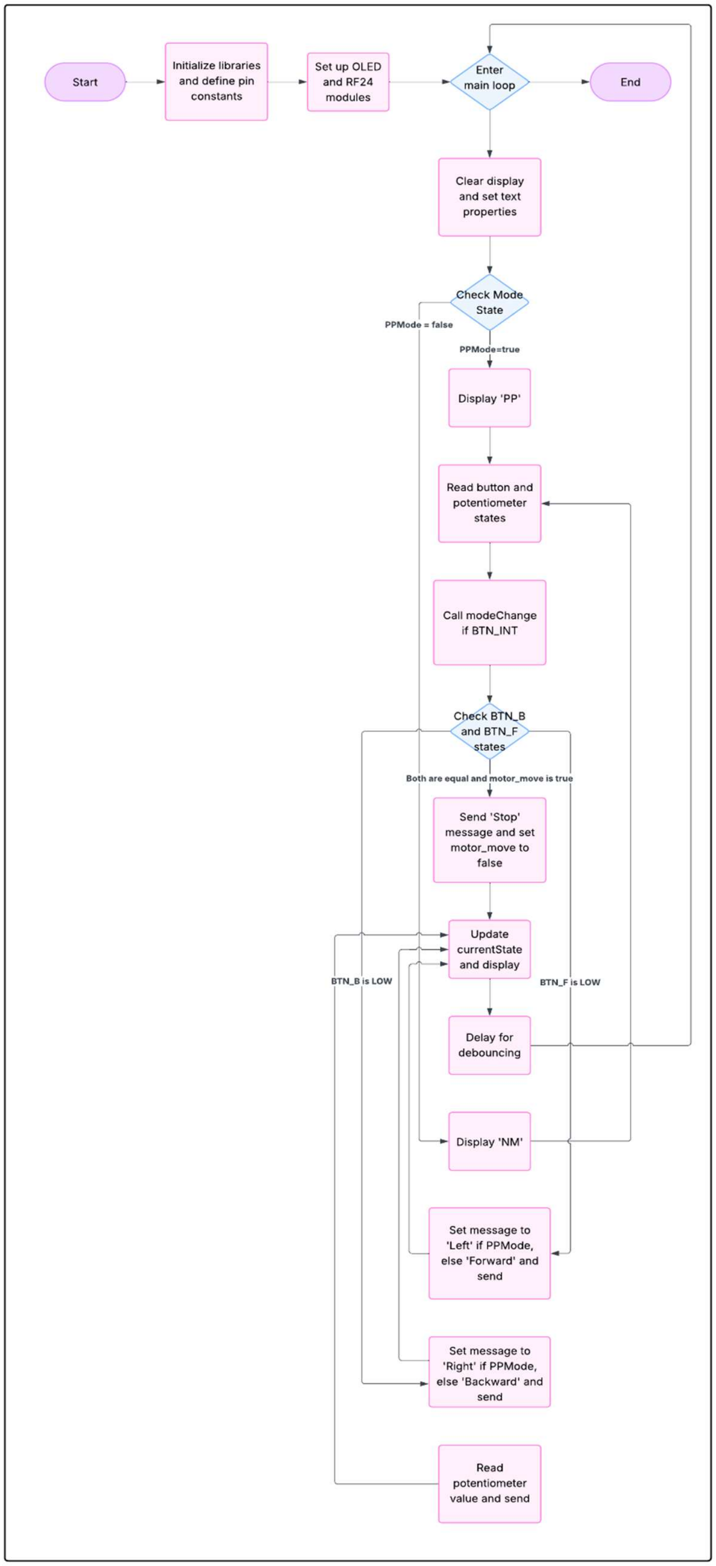
Efficient Diagonal Lane Changing

Instead of performing curved lane shifts like standard vehicles, our system enables the car to move diagonally across lanes. In this mode, all four wheels are aligned to a fixed steering angle, allowing the car to drive at an angle with respect to its body orientation.

This diagonal motion enables a quick, controlled lane change without requiring large turning arcs. Once the maneuver is completed, the car can switch back to Normal Mode for regular movement. This approach is especially useful in tight traffic situations or autonomous path planning, where precision and responsiveness are crucial.

ANNEXURE

Flowchart for the Transmitter End:



Flowchart for the Receiver End: