Report :

Obstruction and Edge Avoiding Car

# Introduction

The "Obstruction and Edge Avoiding Car" project focuses on developing a small-scale autonomous vehicle capable of navigating its environment while avoiding edges and obstacles. This project leverages Internet of Things (IoT) principles, employing a combination of sensors and microcontrollers to achieve autonomous navigation. The vehicle is designed to be a practical demonstration of sensor integration and automated control in robotics.

# Components

# 

# FIG:1

1. **Chassis Kit of Car**: The foundation of the project, providing the structural support and wheels for mobility.
2. **Arduino Uno Board**: The microcontroller that serves as the brain of the car, processing sensor inputs and controlling outputs.
3. **IR Sensors**: Infrared sensors used to detect edges and surfaces, preventing the car from falling off elevated surfaces.
4. **Ultrasonic Sensors**: Employed to detect obstacles in the car's path, ensuring collision avoidance.
5. **Batteries (40V, 4 nos)**: Power source for the entire system, ensuring sufficient energy supply for sensors, the Arduino board, and motors.
6. **Wires**: Essential for connecting all electronic components, enabling communication and power distribution throughout the car.

# System Design and Implementation

# 

# FIG:2

The system integrates multiple sensors with the Arduino Uno to create an autonomous car capable of edge detection and obstacle avoidance. The design process includes several key steps:

## Chassis Assembly:

Assemble the chassis kit, attaching the wheels, motors, and motor drivers.

Secure the Arduino Uno board onto the chassis.

## Sensor Integration:

**IR Sensors**: Mount the IR sensors at strategic positions around the car’s perimeter. These sensors detect changes in surface levels, helping the car avoid edges.

**Ultrasonic Sensors**: Position the ultrasonic sensors at the front of the car to measure distances to obstacles, enabling collision avoidance.

## Power Management:

Connect the 40V batteries in a configuration suitable for powering the Arduino board and the motors.

Ensure voltage regulation to prevent damage to electronic components.

## Wiring and Connections:

Use wires to connect the sensors to the Arduino board’s input pins. Connect the motor drivers to the Arduino’s output pins, ensuring control signals are appropriately routed.

Double-check all connections for stability and correct polarity.

## Programming the Arduino:

Write and upload the code to the Arduino Uno using the Arduino IDE. The code should include algorithms for:

Reading sensor data.

Determining when an edge or obstacle is detected.

Adjusting the car's direction and speed based on sensor inputs.

# Code Overview

Below is a simplified version of the Arduino code used in this project:

**#include <Ultrasonic.h>**

*// Define sensor pins* **#define IR\_SENSOR\_PIN A0 #define ULTRASONIC\_TRIG\_PIN 9**

**#define ULTRASONIC\_ECHO\_PIN 10**

**#define MOTOR\_LEFT\_PIN 3**

**#define MOTOR\_RIGHT\_PIN 5**

Ultrasonic **ultrasonic**(ULTRASONIC\_TRIG\_PIN, ULTRASONIC\_ECHO\_PIN);

**void setup**() { pinMode(IR\_SENSOR\_PIN, INPUT); pinMode(MOTOR\_LEFT\_PIN, OUTPUT); pinMode(MOTOR\_RIGHT\_PIN, OUTPUT);

Serial.begin(9600);

}

**void loop**() {

**int** irValue = analogRead(IR\_SENSOR\_PIN);

**long** distance = ultrasonic.read();

**if** (irValue < THRESHOLD) {

*// Edge detected*

avoidEdge();

} **else if** (distance < SAFE\_DISTANCE) {

*// Obstacle detected*

avoidObstacle();

} **else** {

*// Move forward*

moveForward();

}

}

**void avoidEdge**() {

*// Logic to avoid edge* moveBackward(); delay(500); turnRight();

delay(300);

}

**void avoidObstacle**() {

*// Logic to avoid obstacle* moveBackward(); delay(500);

turnLeft(); delay(300);

}

**void moveForward**() { digitalWrite(MOTOR\_LEFT\_PIN, HIGH); digitalWrite(MOTOR\_RIGHT\_PIN, HIGH);

}

**void moveBackward**() { digitalWrite(MOTOR\_LEFT\_PIN, LOW); digitalWrite(MOTOR\_RIGHT\_PIN, LOW);

}

**void turnRight**() { digitalWrite(MOTOR\_LEFT\_PIN, HIGH); digitalWrite(MOTOR\_RIGHT\_PIN, LOW);

}

**void turnLeft**() { digitalWrite(MOTOR\_LEFT\_PIN, LOW); digitalWrite(MOTOR\_RIGHT\_PIN, HIGH);

}

# Testing and Results

After assembling the car and uploading the code, extensive testing was conducted to ensure the car’s functionality. The car successfully detected edges and obstacles, demonstrating the effectiveness of the sensor integration and control algorithms.

Adjustments were made to fine-tune the sensor sensitivity and motor responses, resulting in reliable autonomous navigation.

# Conclusion

The "Obstruction and Edge Avoiding Car" project showcases the integration of IoT components to create an autonomous vehicle capable of edge detection and obstacle avoidance. This project serves as a practical example of how sensors and microcontrollers can be combined to solve real-world problems in robotics and automation. Future improvements could include enhancing sensor accuracy, incorporating additional sensors, and refining the control algorithms for smoother operation.

# References

* Arduino Uno documentation
* Ultrasonic sensor usage and calibration guides IR sensor integration techniques
* Chassis assembly instructions and motor control guides