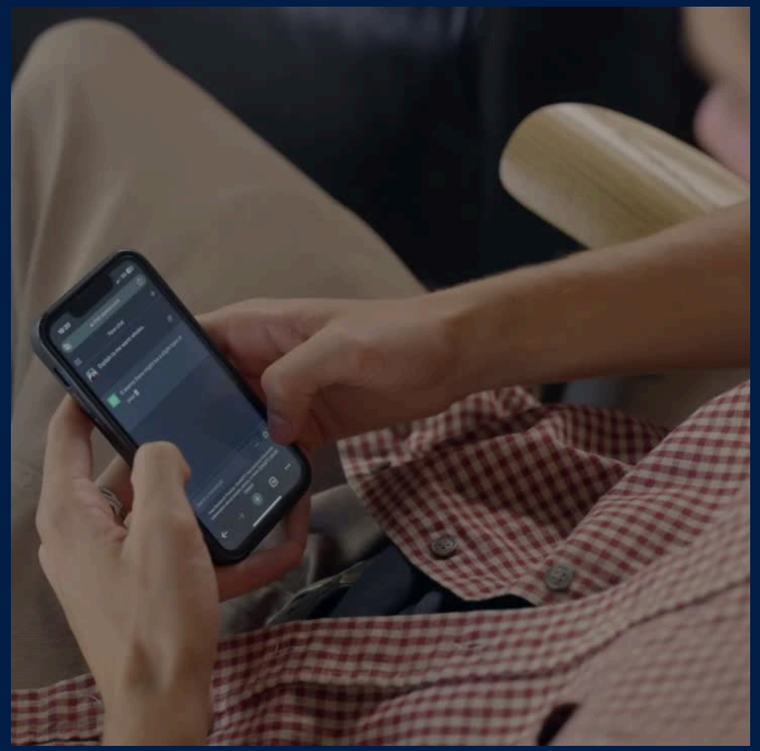
LANE, OBSTACLE AND MINE DETECTION ROBOT

Anushka Sethia 1BM22ET009
Aprajita Gupta 1BM22ET010
Palasha Kare 1BM22ET041
Ungarala Deeraj 1BM22ET059

PROJECT OVERVIEW

The Arduino-based Robotic Car with Ultrasonic and IR Sensors and Mine Detector project aims to showcase the integration of various sensors and actuators to create an autonomous vehicle capable of navigating through its environment. In this project, we utilize an Arduino microcontroller as the central processing unit, incorporating ultrasonic sensors for distance measurement and obstacle detection, as well as infrared (IR) sensors for line following or additional obstacle detection and specialized mine detection sensors to identify hazardous objects or areas.

02

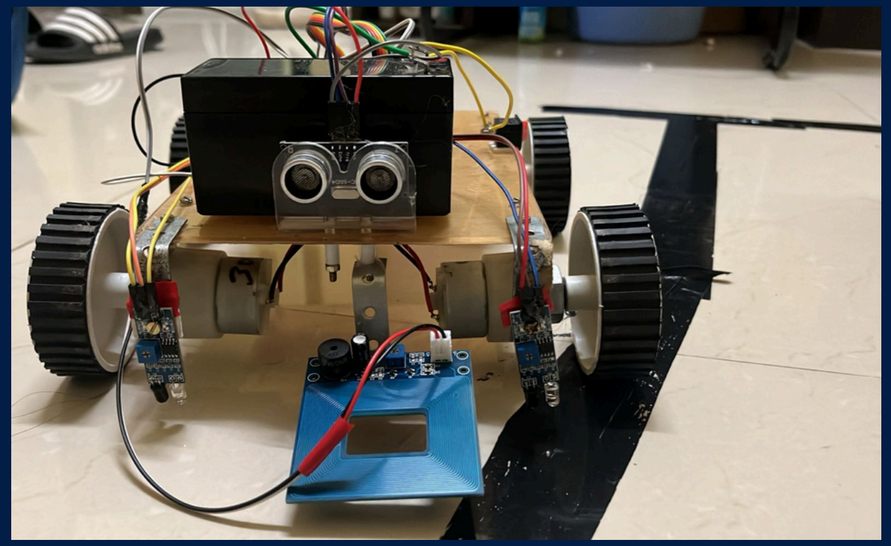


PROBLEM STATEMENT

- Lane Drifting Accidents
- Collisions with Moving or Static Objects
- Human Risk in Minefields
- Inaccessible Locations

03

SOLUTION

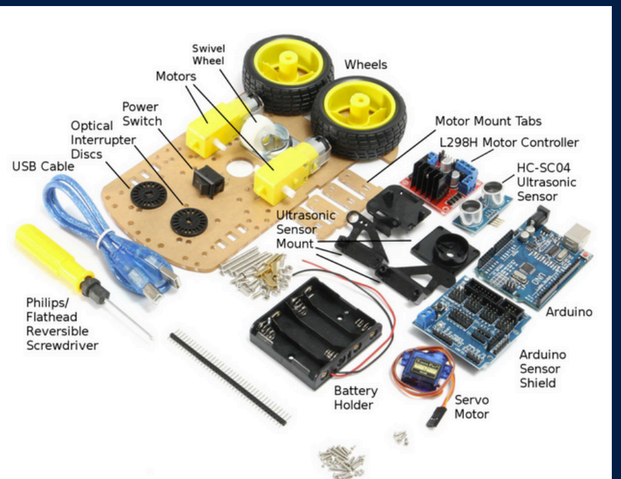


- Accurate Lane Tracking
- Mine Detection by Automation
- Obstacle Detection in Real Time
- Collision-avoidance systems

METHODOLOGY 04

COMPONENTS REQUIRED:

- 1 X CAR CHASSIS
- 4 X CAR WHEELS
- 2 X DC GEAR MOTOR
- 4 X FASTENERS (HIGH INTENSITY BLACK ACRYLIC)
- 1 X 12V BATTERY BOX
- 1 X QUALITY ROCKER SWITCH
- 2 X IR SENSORS
- 1 X UNO R3
- 1 X L298N
- 1 X MINE DETECTOR
- 1 X ULTRASONIC SENSOR

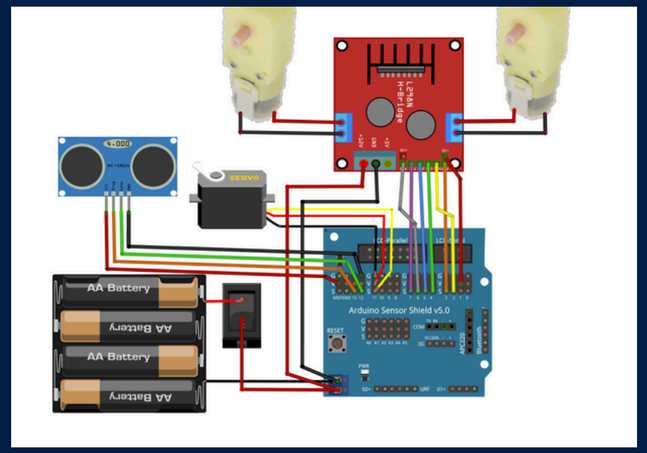


STEPS FOLLOWED:

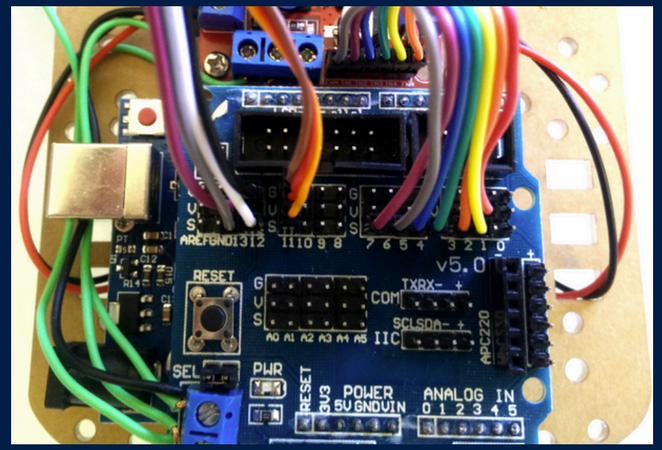
- SWIVEL WHEEL: THE SWIVEL WHEEL WAS MOUNTED ON THE BASEPLATE USING M3 SCREWS AND NUTS.
- ARDUINO AND SHIELD: THE ARDUINO AND SENSOR SHIELD WERE ATTACHED USING SPACERS AND SCREWS, ENSURING PROPER PIN ALIGNMENT.
- MOTORS AND WHEELS: MOTORS WERE SOLDERED WITH WIRES, MOUNTED USING ACRYLIC TABS AND SCREWS, AND FITTED WITH WHEELS.
- POWER AND CONTROLLER: THE SWITCH, BATTERY HOLDER, AND L298N BOARD WERE INSTALLED AND WIRED. JUMPERS WERE REMOVED FROM THE L298N BOARD FOR SETUP.
- FIRMWARE: THE SERVO WAS CENTERED BY UPLOADING THE “ROBOT-CAR“ SKETCH THROUGH THE ARDUINO IDE.
- ULTRASONIC SENSOR: THE ULTRASONIC SENSOR WAS MOUNTED ONTO A SERVO MOTOR AND ATTACHED TO THE BASEPLATE.
- CONNECTIONS: ALL SENSORS, SERVOS, AND MOTORS WERE WIRED TO THE SENSOR SHIELD AND L298N BOARD.
- TESTING: THE ROBOT WAS POWERED AND TESTED, ENSURING PROPER SERVO CENTERING AND WHEEL DIRECTIONS. ADJUSTMENTS WERE MADE WHERE NECESSARY.

CIRCUIT: PROGRESS:

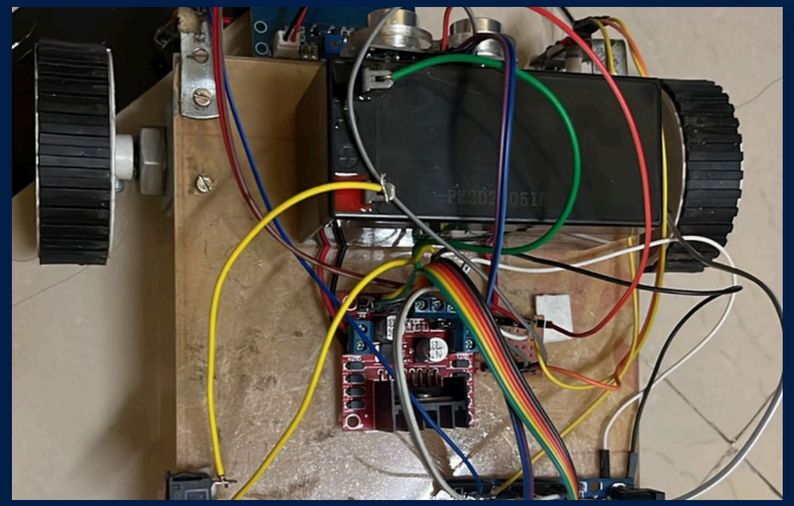
05



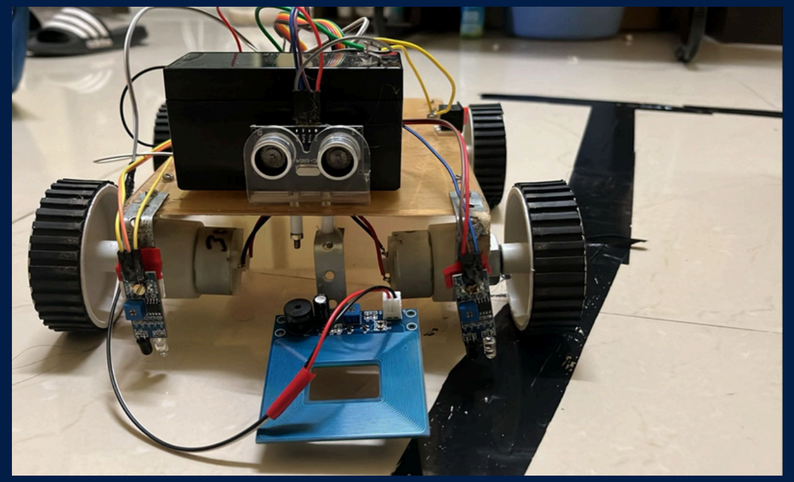
a) Circuit diagram



b) Circuit diagram



c)Top view



d) Front view

STRUCTURE OF WORK

Lane Detection Module (using Infrared Sensors)

In a lane detection module using infrared (IR) sensors, the sensors emit infrared light and detect its reflection from the road surface. Dark lines (lane markings) reflect less IR light than lighter surfaces, allowing the sensor to distinguish lane boundaries by measuring the reflected IR intensity.

07

Lane Detection Module (using UltraSonic Sensors)

The sensors emit high-frequency sound waves and measure the time it takes for the waves to bounce back from nearby surfaces. They help detect obstacles or curbs rather than lane markings by calculating the distance to objects alongside the vehicle.

Lane Detection Module (using Mine Detector Sensors)

A mine detector sensor works by generating an electromagnetic field and detecting disturbances caused by the presence of metal objects. When a metallic object, like a mine, disrupts the field, the sensor signals its presence, helping to locate buried explosives or objects.

IMPLEMENTATION:

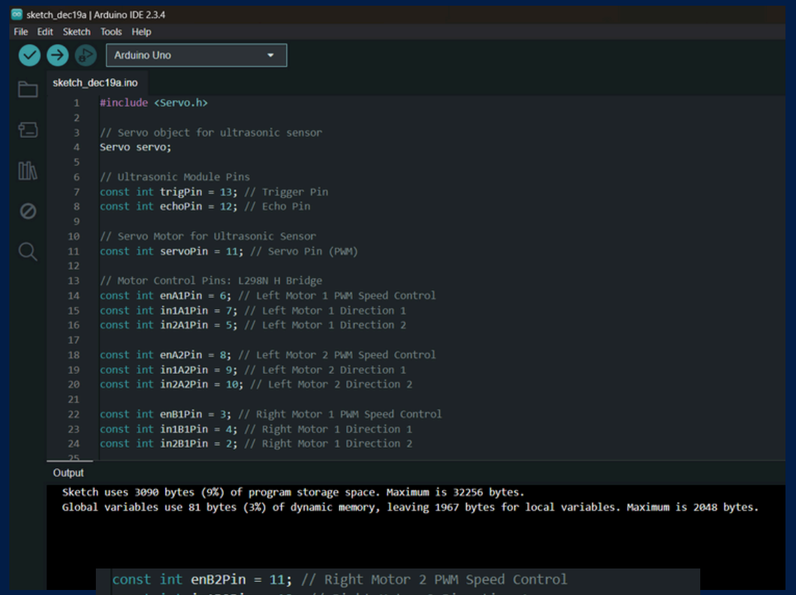
08

The mine detection RC car has the following uses:

1. Mine Detection: Locates buried mines using a metal detection coil, enhancing safety in hazardous areas.
2. Surveillance: Navigates challenging terrains for remote monitoring and exploration.
3. Rescue Operations: Assists in identifying dangerous zones during rescue missions.
4. Educational Tool: Demonstrates robotics, sensor integration, and automation concepts.
5. Military Applications: Supports demining operations in conflict zones.

SOFTWARE(ARDUINO IDE):

09




```ino
#include <Servo.h>

// Servo object for ultrasonic sensor
Servo servo;

// Ultrasonic Module Pins
const int trigPin = 13; // Trigger Pin
const int echoPin = 12; // Echo Pin

// Servo Motor for Ultrasonic Sensor
const int servoPin = 11; // Servo Pin (PWM)

// Motor Control Pins: L298N H Bridge
const int enA1Pin = 6; // Left Motor 1 PWM Speed Control
const int in1A1Pin = 7; // Left Motor 1 Direction 1
const int in2A1Pin = 5; // Left Motor 1 Direction 2

const int enA2Pin = 8; // Left Motor 2 PWM Speed Control
const int in1A2Pin = 9; // Left Motor 2 Direction 1
const int in2A2Pin = 10; // Left Motor 2 Direction 2

const int enB1Pin = 3; // Right Motor 1 PWM Speed Control
const int in1B1Pin = 4; // Right Motor 1 Direction 1
const int in2B1Pin = 2; // Right Motor 1 Direction 2
```

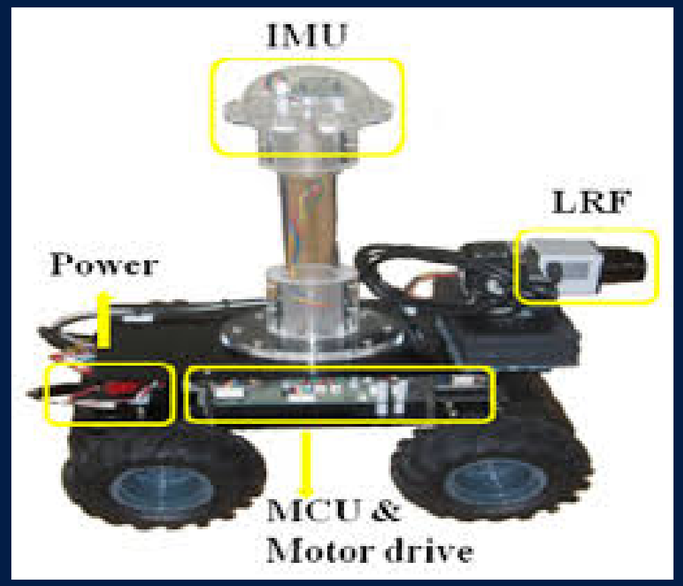
```ino
const int enB2Pin = 11; // Right Motor 2 PWM Speed Control
const int in1B2Pin = 12; // Right Motor 2 Direction 1
const int in2B2Pin = 13; // Right Motor 2 Direction 2

// Speed of sound in air: 343 m/s
#define SOUND_SPEED 343

// Servo Angles and Distance Arrays
#define NUM_ANGLES 7
unsigned char sensorAngle[NUM_ANGLES] = {60, 70, 80, 90, 100, 110, 120};
unsigned int distance[NUM_ANGLES];

// Motor Enumeration
enum Motor { LEFT, RIGHT };

// Function to control a motor group (LEFT or RIGHT)
void go(enum Motor side, int speed) {
  if (side == LEFT) {
    digitalWrite(in1A1Pin, speed > 0 ? HIGH : LOW);
    digitalWrite(in2A1Pin, speed <= 0 ? HIGH : LOW);
    analogWrite(enA1Pin, abs(speed));

    digitalWrite(in1A2Pin, speed > 0 ? HIGH : LOW);
    digitalWrite(in2A2Pin, speed <= 0 ? HIGH : LOW);
    analogWrite(enA2Pin, abs(speed));
  } else {
    digitalWrite(in1B1Pin, speed > 0 ? HIGH : LOW);
    digitalWrite(in2B1Pin, speed <= 0 ? HIGH : LOW);
    analogWrite(enB1Pin, abs(speed));

    digitalWrite(in1B2Pin, speed > 0 ? HIGH : LOW);
    digitalWrite(in2B2Pin, speed <= 0 ? HIGH : LOW);
    analogWrite(enB2Pin, abs(speed));
  }
}
```

```ino
// Main loop: Avoid obstacles and move
void loop() {
  readNextDistance();

  // Check if anything is too close
  unsigned char tooClose = 0;
  for (unsigned char i = 0; i < NUM_ANGLES; i++) {
    if (distance[i] < 300) { // Object within 300 mm
      tooClose = 1;
      break;
    }
  }

  if (tooClose) {
    // Something is nearby: back up left
    go(LEFT, -180);
    go(RIGHT, -80);
  } else {
    // Nothing in the way: move forward
    go(LEFT, 255);
    go(RIGHT, 255);
  }

  // Delay before the next reading
  delay(50);
}
```

```ino
// Test Motors: Forward/backward for all motors
void testMotors() {
  static int speed[4] = {128, 255, -128, -255};
  for (int i = 0; i < 4; i++) {
    go(LEFT, speed[i]);
    go(RIGHT, speed[i]);
    delay(200);
  }
  go(LEFT, 0);
  go(RIGHT, 0);
}

// Read distance from the ultrasonic sensor in mm
unsigned int readDistance() {
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10); // Send a 10-microsecond pulse
  digitalWrite(trigPin, LOW);

  unsigned long duration = pulseIn(echoPin, HIGH); // Measure high pulse
  return duration * SOUND_SPEED / 2000;          // Calculate distance in mm
}

// Sweep ultrasonic sensor and record distances
void readNextDistance() {
  static unsigned char angleIndex = 0;
  static signed char step = 1;

  distance[angleIndex] = readDistance();
  angleIndex += step;

  if (angleIndex == NUM_ANGLES - 1) step = -1;
  else if (angleIndex == 0) step = 1;

  servo.write(sensorAngle[angleIndex]);
}
```

```ino
// Initial setup
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  digitalWrite(trigPin, LOW);

  pinMode(enA1Pin, OUTPUT);
  pinMode(in1A1Pin, OUTPUT);
  pinMode(in2A1Pin, OUTPUT);

  pinMode(enA2Pin, OUTPUT);
  pinMode(in1A2Pin, OUTPUT);
  pinMode(in2A2Pin, OUTPUT);

  pinMode(enB1Pin, OUTPUT);
  pinMode(in1B1Pin, OUTPUT);
  pinMode(in2B1Pin, OUTPUT);

  pinMode(enB2Pin, OUTPUT);
  pinMode(in1B2Pin, OUTPUT);
  pinMode(in2B2Pin, OUTPUT);

  servo.attach(servoPin);
  servo.write(90); // Center the servo

  // Turn off motors
  go(LEFT, 0);
  go(RIGHT, 0);

  // Test motors
  testMotors();

  // Pre-scan surroundings
  servo.write(sensorAngle[0]);
  delay(200);
  for (unsigned char i = 0; i < NUM_ANGLES; i++) {
    readNextDistance();
    delay(200);
```

Limitations and Solutions



- Power Struggles
- Sensor Mood Swings
- Chassis Chaos
- Overheating
- Real-World curveballs
- Components clashes

CONCLUSION:

The Arduino-based Robotic Car integrates ultrasonic, IR, and mine detection sensors for autonomous navigation. It’s a low-cost, adaptable solution for tasks like obstacle avoidance, line following, and hazard detection. Ideal for search and rescue, industrial, and hazardous environment applications, the robot's modular design allows for easy customization. Future improvements in AI and sensor integration will expand its capabilities, offering safer and more efficient solutions across various fields.