**Motorized Radar Exploration Bot Project Report**

**Name:** Hari and Team  
**Class / Section:** 11  
**Date:**

## ****Project Title:****

**Motorized Radar Exploration Bot**

## ****Objective:****

To design and construct a motorized bot that autonomously explores its surroundings, continuously detecting obstacles using a servo-mounted ultrasonic sensor, and visualizing obstacle data on a PC using Processing software. The bot demonstrates motor control, sensor integration, and Arduino-based electronics.

## ****Components Needed:****

* Arduino Uno -(Having)
* L298N motor driver module - 1
* Bo geared motor and wheel - 4
* Ultrasonic sensor (HC-SR04) - 1
* Servo motor - 1
* Lever switch - 1
* OLED / LCD display (Optional)
* 2S Li-ion battery (7.4V) or suitable powerbank
* Acrylic sheet for basement and body
* Wires, resistors, and switches

## ****Working Principle:****

**1. Bot Movement**

* **Bot moves forward for a fixed duration.**
* **Performs a U-turn (or partial turn) after moving forward.**
* **Loops this movement indefinitely.**
* **No obstacle avoidance: the bot is for detection/exploration, not for navigating around obstacles.**

**2. Radar System**

* **Servo-mounted ultrasonic sensor sweeps from 15° to 190° continuously.**
* **Measures distance at each angle and sends data over Serial to a PC.**
* **Processing software visualizes the radar sweep in real time:**
* **Sweep line shows servo angle**
* **Red dots indicate detected obstacles and their distance**

**3. Lever Switch Control**

* **Lever switch acts as start/stop toggle:**
* **ON → bot runs movement loop and radar sweep**
* **OFF → bot stops all motors and radar servo**

**4. Display Integration**

* **OLED can show Distance of nearest object (optional)**
* **Status: “Running” / “Stopped”**
* **OLED does not show radar sweep, only front/nearest distance info.**

**5. Power Supply**

* **2S Li-ion battery (7.4V) powers Arduino and motors.**

**Common GND must be shared across all components.**

## ****Key Features / Notes:****

* Fully autonomous exploration bot
* Servo-mounted radar sensor for continuous obstacle detection
* U-turn movement loop for full area coverage
* Lever switch for manual start/stop control
* Processing software provides live radar visualization
* Modular design allows future upgrades (additional sensors, remote control, etc.)

## ****Arduino Code:****

#include <Servo.h>

// Motor pins

#define IN1 5

#define IN2 4

#define IN3 8

#define IN4 12

#define ENA 3

#define ENB 11

// Radar ultrasonic pins

#define RADAR\_TRIG 9

#define RADAR\_ECHO 10

#define SERVO\_PIN 6

// Lever switch

#define LEVER\_PIN 2

Servo radarServo;

long duration;

int distance;

int angle;

unsigned long moveTime = 3000; // move forward 3 sec

unsigned long turnTime = 1000; // U-turn 1 sec (adjust based on motors)

int turnAngle = 180; // U-turn degrees

void setup() {

Serial.begin(9600);

// Motor setup

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

pinMode(ENA, OUTPUT);

pinMode(ENB, OUTPUT);

analogWrite(ENA, 180); // motor speed

analogWrite(ENB, 180);

// Lever switch

pinMode(LEVER\_PIN, INPUT\_PULLUP); // active LOW

// Radar sensor + servo

pinMode(RADAR\_TRIG, OUTPUT);

pinMode(RADAR\_ECHO, INPUT);

radarServo.attach(SERVO\_PIN);

}

void loop() {

// Check lever switch

if (digitalRead(LEVER\_PIN) == HIGH) { // OFF -> stop everything

stopBot();

return;

}

// Move forward

moveForward();

delay(moveTime);

// U-turn

uTurn();

}

// =================== Motor Functions ===================

void moveForward() {

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

}

void stopBot() {

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

}

void uTurn() {

// Spin in place (left/right motors opposite)

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

delay(turnTime);

stopBot();

}

// =================== Radar Sensor Functions ===================

void sweepRadar() {

// Sweep from 15° to 190° and back

for (angle = 15; angle <= 190; angle += 5) {

radarServo.write(angle);

delay(30);

sendRadarData(angle);

}

for (angle = 190; angle >= 15; angle -= 5) {

radarServo.write(angle);

delay(30);

sendRadarData(angle);

}

}

void sendRadarData(int ang) {

distance = calculateDistance();

Serial.print(ang);

Serial.print(",");

Serial.print(distance);

Serial.print(".");

}

int calculateDistance() {

digitalWrite(RADAR\_TRIG, LOW);

delayMicroseconds(2);

digitalWrite(RADAR\_TRIG, HIGH);

delayMicroseconds(10);

digitalWrite(RADAR\_TRIG, LOW);

duration = pulseIn(RADAR\_ECHO, HIGH);

return duration \* 0.034 / 2; // cm

}

**Project Display & Visualization**

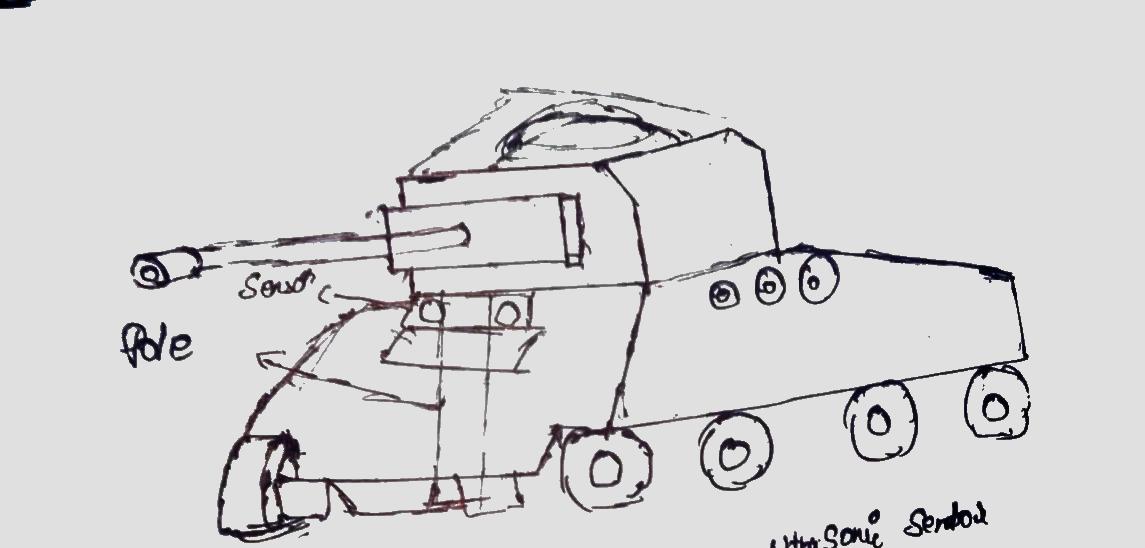
1. Front Sensor + OLED (On the Tank)

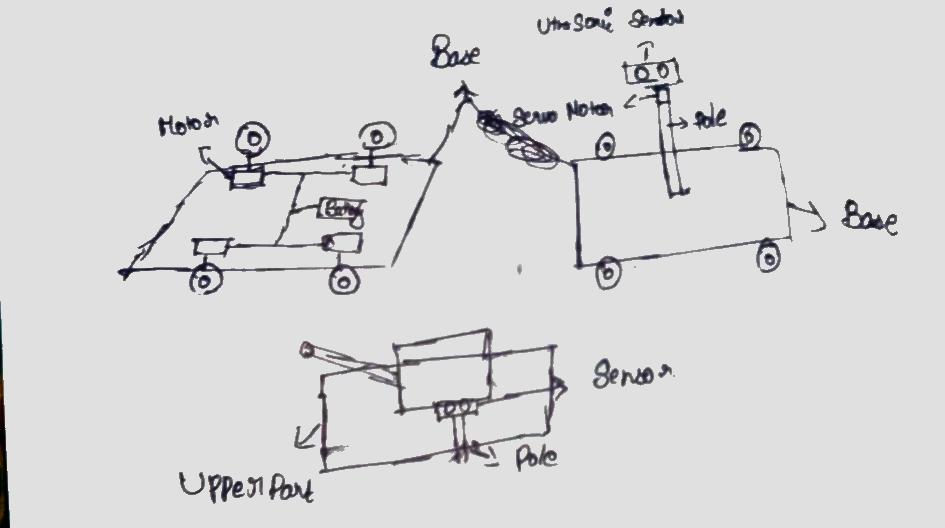
* The front ultrasonic sensor is mounted fixed on the tank.
* It constantly measures distance directly ahead.
* The OLED display on the tank shows:
* Distance in cm
* Status: “Clear”, “Obstacle!”, or “Turning…”
* This works fully independently, no PC needed.

1. Radar Sensor + Processing Software (PC Visualization)

* The radar ultrasonic sensor is mounted on a servo and sweeps from 15° → 190°.
* Arduino sends angle + distance data over the USB Serial connection.
* Processing software on a PC reads this data and draws a radar visualization:
* Sweep line shows servo angle
* Red marks show detected objects and distances
* Without a PC, this radar visualization cannot appear on the OLED — OLED only shows front distance and status.

Sample design





## ****Conclusion:****

The motorized radar exploration bot successfully moves in a loop, performs U-turns, and continuously scans its surroundings using a servo-mounted ultrasonic sensor. Obstacle data is sent to a PC for real-time visualization using Processing software. This project demonstrates practical applications of Arduino programming, motor control, and sensor integration in a mobile robotics platform.