Robotic Application Development

COHDSE21.2F

Course Work

Project Report

National Institute of Business Management

Colombo - 07

Robot Car

Robert Programming

Group members:

1. Yasiru Athukorala (COHDSE21.2F-050)
2. Pruthuvi De Silva (COHDSE21.2F-032)
3. Janith Dissanayake (COHDSE21.2F-004)
4. Tharusha Perera (COHDSE21.2F-028)
5. Tashiya Fernando (COHDSE21.2F-010)

Content

Introduction

A Robot car is any machine which is completely automatic, i.e., it starts on its own, decides its own way of work and stops on its own.

To build up this robot car we choose Arduino programming language and IDE to code self-automatic robot car. Actually, this robot car is a line following robot car. In nowadays mainly in hotels, roads, and most of the places these line following robots are used.

So, we developed simple line following robot car to demonstrate the activation of all the line following robots.

This project is done by as a course work of our Robotic Application Development Module

Goal

The goal of project to build a line following and obstacle avoiding robot car that can work autonomously.

Motivation

The line following and obstacle avoiding robot car is commonly used for:

- Industrial Applications:

  These robots can be used as automated equipment carriers in industries replacing traditional conveyer belts.

- Automobile applications:

  These robots can also be used as automatic cars running on roads with embedded magnets.

- Domestic applications:

  These can also be used at homes for domestic purposes like floor cleaning etc.

- Guidance applications:

  These can be used in public places like shopping malls, museums etc. to provide path guidance.

Problems and Solving

Question no. 01:

How will the robot follow the line?

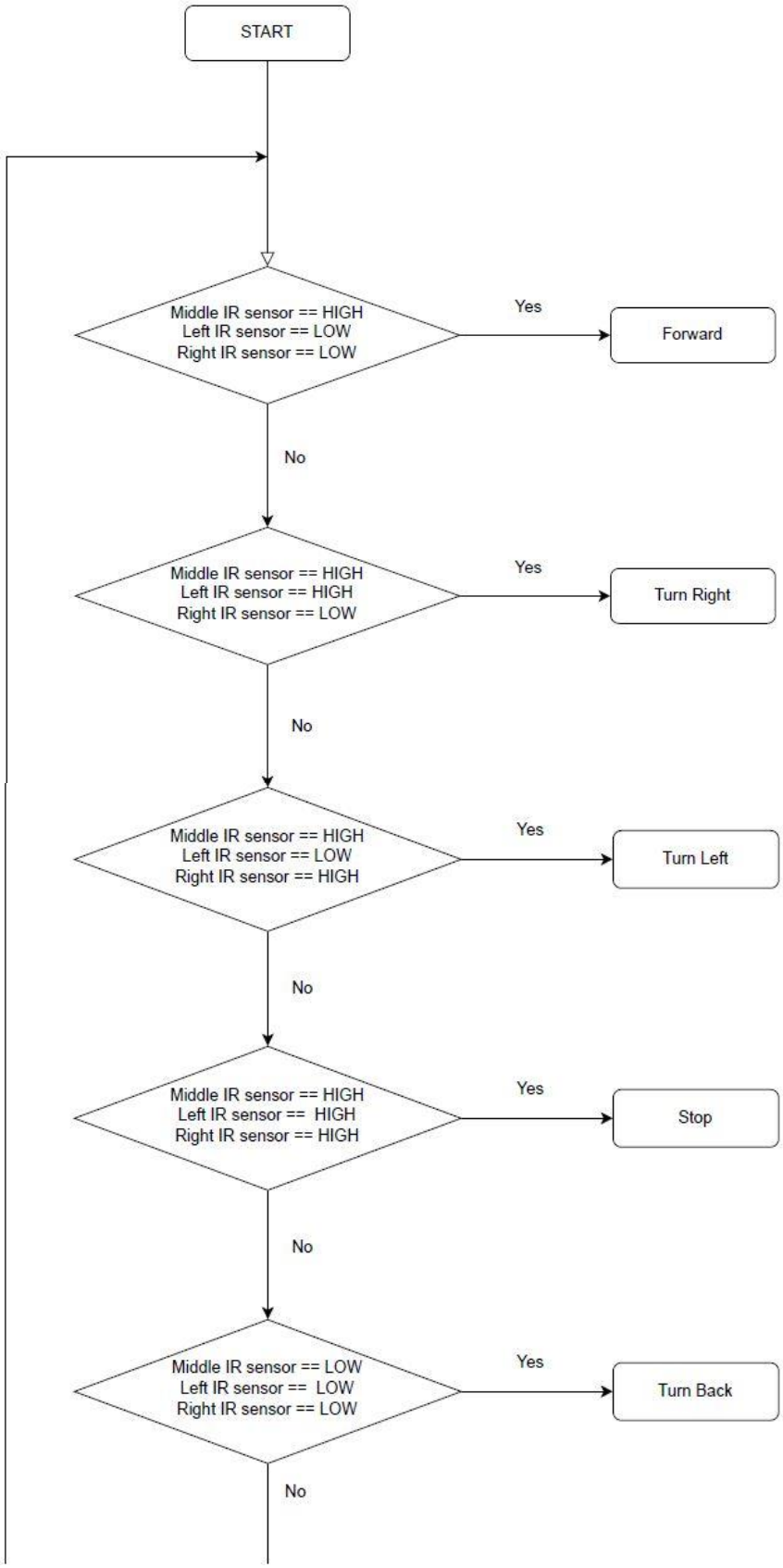
- Using infrared sensors!

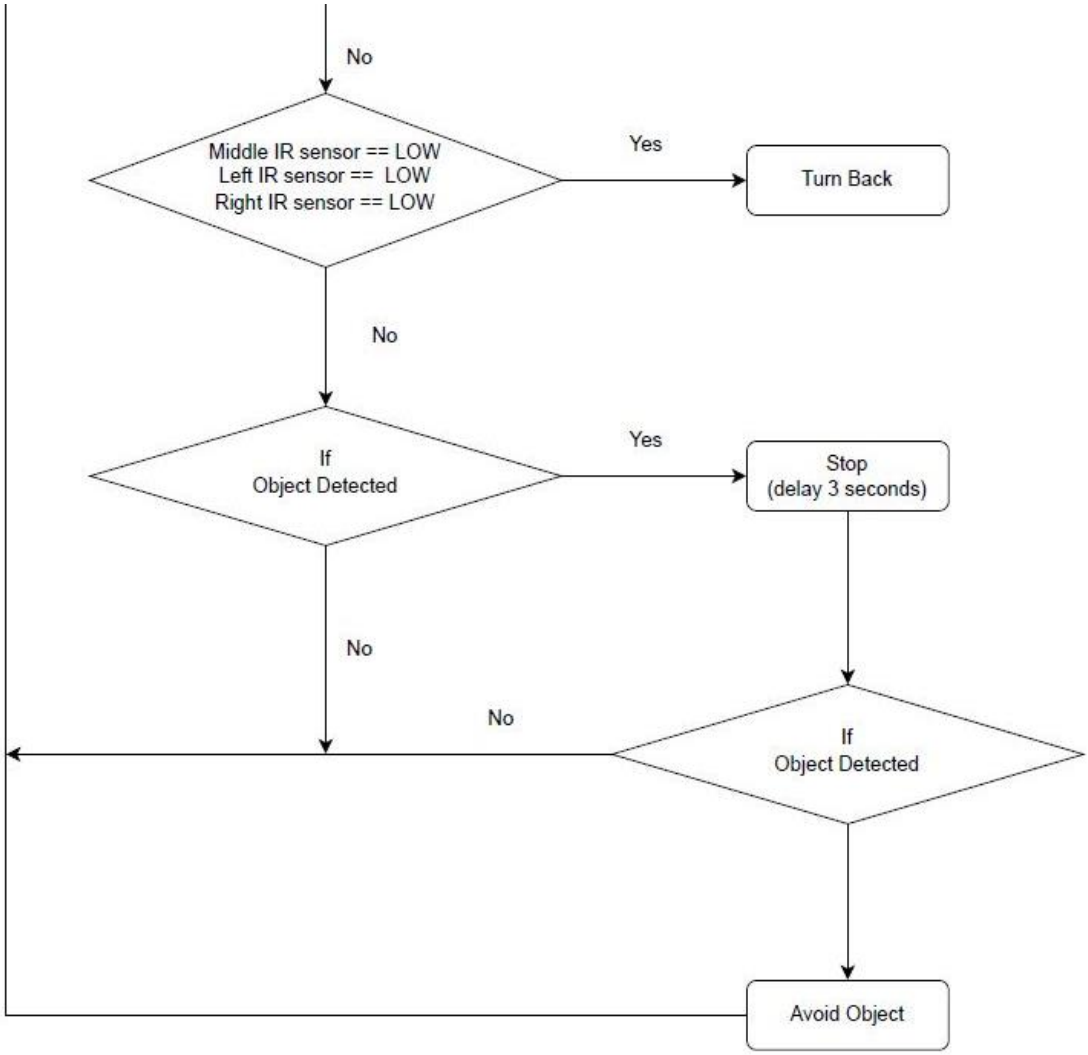
Question no. 02:

How to let a robot car avoid an obstacle during the line following?

- Using an ultrasonic sensor to detect the obstacle and then circumnavigate it.

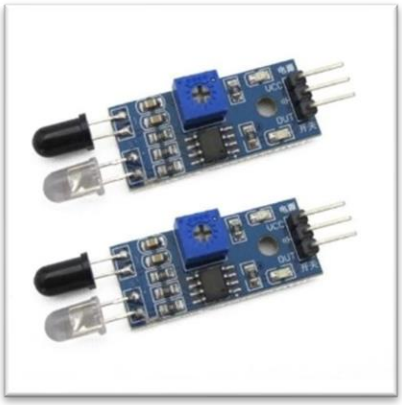
How will it work? (Flow Diagram)

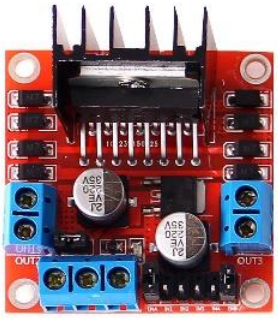



No
Middle IR sensor == LOW
Left IR sensor == LOW
Right IR sensor == LOW
Yes
Turn Back
No
If
Object Detected
Yes
Stop
(delay 3 seconds)
No
No
If
Object Detected
Avoid Object

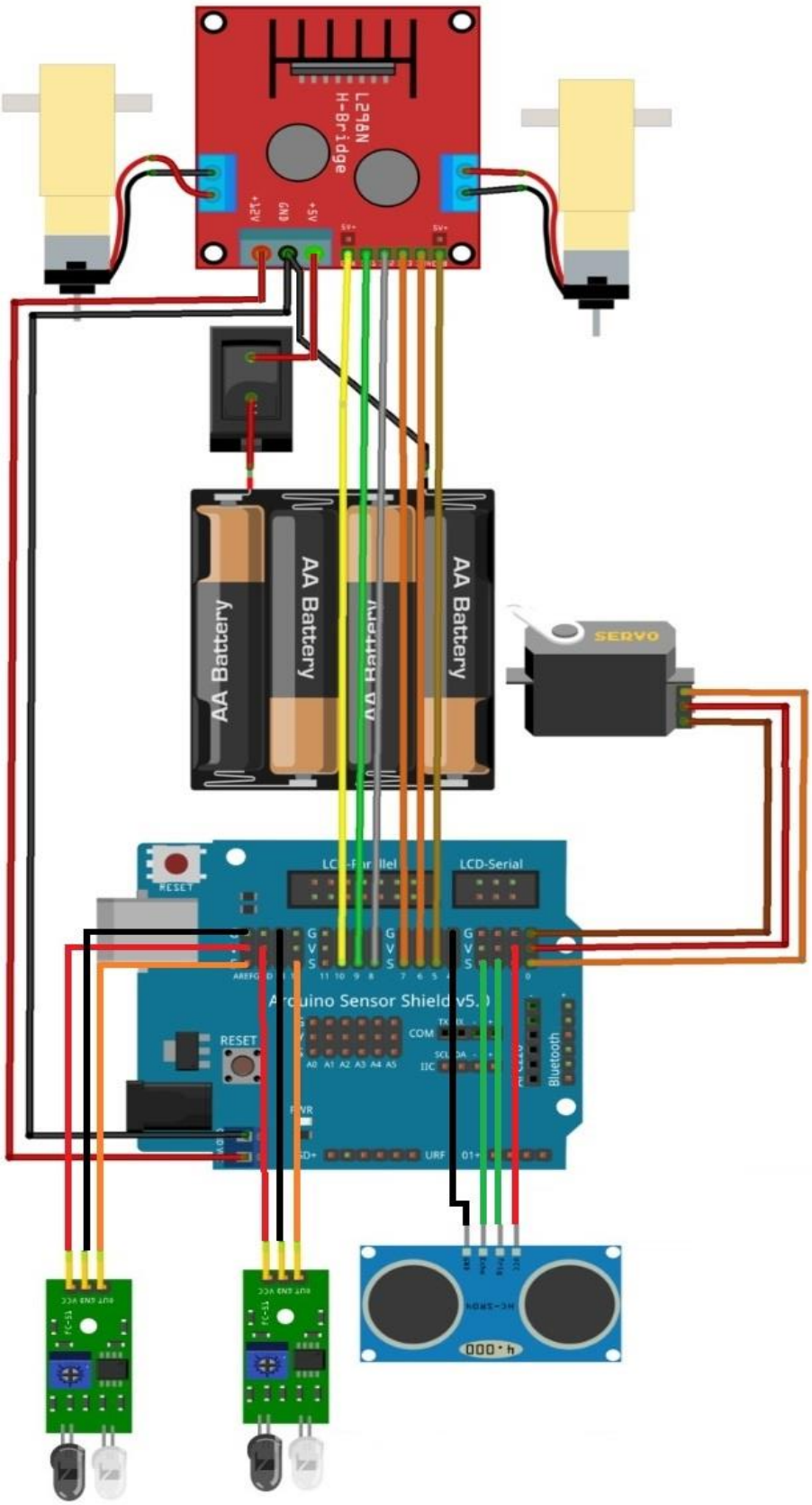
What we used

1. Arduino Uno
2. Ultrasonic Sensor
3. Infrared sensors
4. Servo motor
5. L298 Motor drive
6. Batteries
7. Wheels
8. Car chassis pack

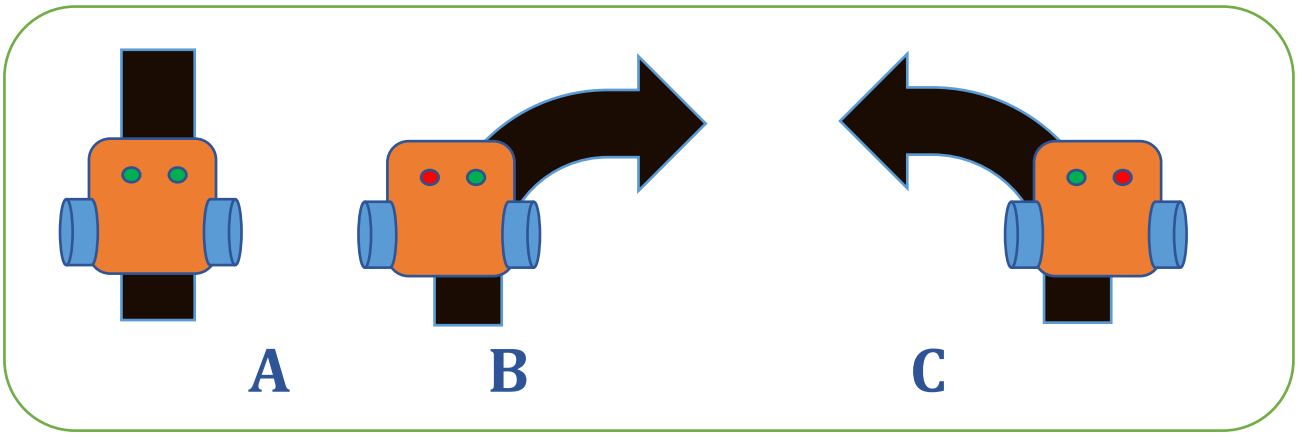




Circuit Diagram



How it's following the line



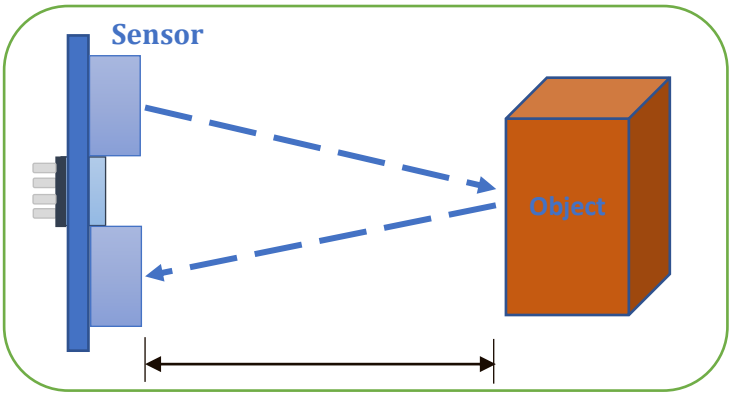
A. Both IR LEDs are on the line. Robot car should move forward.

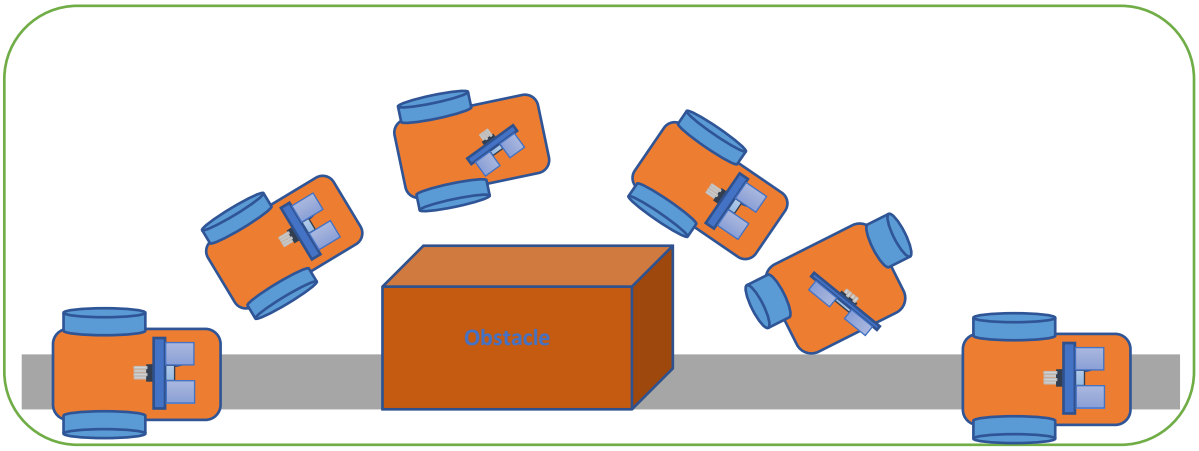
B. The Right IR LED is on the line while the left deviates to the left. Robot car should turn to the right.

C. The Left IR LED is on the line while the right deviates to the right. Robot car should turn to the left.

How it's avoiding obstacles

The basic principle behind the working of ultrasonic sensor is to note down the time taken by sensor to transmit ultrasonic beams and receiving the ultrasonic beams after hitting the surface. Then further the distance is calculated.





CODE

```
#include <Servo.h>
#include <NewPing.h>

#define speedPinR 5
#define RightMotorDirPin1 4
#define RightMotorDirPin2 7
#define speedPinL 6
#define LeftMotorDirPin1 8
#define LeftMotorDirPin2 12

#define SERVO_PIN 10

#define Echo_PIN 2
#define Trig_PIN 3

#define SPEED1 130
#define SPEED2 110
#define SPEED3 100

#define max_dist 150
#define stop_range 10
#define stop_wait 4

#define IRSensorL A2
#define IRSensorM A1
#define IRSensorR A0

#define buzzer 13

int cm;
bool turning;
int counter;


NewPing sonar(Trig_PIN, Echo_PIN, max_dist);
Servo myservo;

void setup() {

  pinMode(RightMotorDirPin1, OUTPUT);
  pinMode(RightMotorDirPin2, OUTPUT);
  pinMode(speedPinL, OUTPUT);

  pinMode(LeftMotorDirPin1, OUTPUT);
  pinMode(LeftMotorDirPin2, OUTPUT);
  pinMode(speedPinR, OUTPUT);
```

```
  pinMode(Trig_PIN, OUTPUT);
  pinMode(Echo_PIN, INPUT);

  pinMode(IRSensorL, INPUT);
  pinMode(IRSensorM, INPUT);
  pinMode(IRSensorR, INPUT);

  pinMode(buzzer, OUTPUT);


  myservo.attach(SERVO_PIN);
  myservo.write(85);
  delay(200);

  Serial.begin(9600);

  stop_Stop();
//  counter = 0;
}

void loop() {


//  int counter2 = 0;
//  cm = readDistance();
//  Serial.println(cm);

  int x = digitalRead(IRSensorL);
  int y = digitalRead(IRSensorR);


  if(detectObj())
  { turnUSsensorMiddle();
    obstacleDetector();
  }
  else
  { followLine();
  }
}


void followLine() {

  int valueLeft = digitalRead(IRSensorL);
  int valueRight = digitalRead(IRSensorR);
  int valueMiddle = digitalRead(IRSensorM);

  if ((valueLeft == HIGH && valueRight == LOW )) {
    go_Left();
```

```
  } else if ((valueLeft == LOW &&  valueRight == HIGH)) {
    go_Right();
  } else if (valueLeft == HIGH &&  valueRight == HIGH && valueMiddle == HIGH) {
    go_Right();
  } else if (valueLeft == LOW &&  valueRight == LOW && valueMiddle == LOW) {
    turn_back();
  }
  else {
    go_Advance();
  }
}

//Set Motor Speed
void set_Motorspeed(char g) {
  if (g == 'L' || g == 'R' || g == 'S') {
    analogWrite(speedPinL, SPEED2);
    analogWrite(speedPinR, SPEED2);
  }else if(g == 'T'){
    analogWrite(speedPinL, SPEED3);
    analogWrite(speedPinR, SPEED3);
  }
  else {
    analogWrite(speedPinL, SPEED1);
    analogWrite(speedPinR, SPEED1);
  }

}

//Read Ultrasonic Sensor
int readDistance() {
  int cm = sonar.ping_cm();
  return cm;
}

void go_Advance() {
  digitalWrite(RightMotorDirPin1, LOW);
  digitalWrite(RightMotorDirPin2, HIGH);
  digitalWrite(LeftMotorDirPin1, LOW);
  digitalWrite(LeftMotorDirPin2, HIGH);
  set_Motorspeed('A');
}
void go_Right() {
  digitalWrite(RightMotorDirPin1, LOW);
  digitalWrite(RightMotorDirPin2, HIGH);
  digitalWrite(LeftMotorDirPin1, HIGH);
  digitalWrite(LeftMotorDirPin2, LOW);
  set_Motorspeed('R');
}
```

```
void go_Left() {
  digitalWrite(RightMotorDirPin1, HIGH);
  digitalWrite(RightMotorDirPin2, LOW);
  digitalWrite(LeftMotorDirPin1, LOW);
  digitalWrite(LeftMotorDirPin2, HIGH);
  set_Motorspeed('L');
}

void stop_Stop() {
  digitalWrite(RightMotorDirPin1, LOW);
  digitalWrite(RightMotorDirPin2, LOW);
  digitalWrite(LeftMotorDirPin1, LOW);
  digitalWrite(LeftMotorDirPin2, LOW);
  set_Motorspeed('S');
}

void turn_back() {
  digitalWrite(RightMotorDirPin1, LOW);
  digitalWrite(RightMotorDirPin2, HIGH);
  digitalWrite(LeftMotorDirPin1, HIGH);
  digitalWrite(LeftMotorDirPin2, LOW);
  set_Motorspeed('T');
}

void obstacleDetector()
{
  stop_Stop();
  turnUSsensorMiddle();
  delay(3000);

  if(detectObj())
  { turnUSsensorLeft();
    delay(1000);
    if(detectObj())
    { turnUSsensorMiddle();
      stop_Stop();
      delay(100);
    }
    else
    { offRoadL();
    }
  }
  else
  { go_Advance();
    delay(100);
  }
}

void turnUSsensorLeft()
```

```
{ Serial.println("sensor turned left");
  myservo.write(135);
}

void turnUSsensorRight()
{  Serial.println("sensor turned right");
   myservo.write(2);
}

void turnUSsensorMiddle()
{ Serial.println("sensor turned middle");
  myservo.write(90);
}


void offRoadL()
{ turnUSsensorMiddle();
  delay(500);
  while(detectObj())
  { go_Right();
    delay(50);
    stop_Stop();
    delay(50);
  }
  while(detectObjDistance()>20)
  { turnUSsensorRight();
    delay(100);
    go_Advance();
    delay(100);
    stop_Stop();
    delay(10);
  }
  delay(1000);
  turnUSsensorRight();
  delay(500);
  while(!detectLineIRm())
  { if(detectObjDistance()<20)
    { go_Right();delay(100);
      go_Advance();delay(20);
      stop_Stop();delay(50);
      Serial.println(detectObjDistance());
    }
    else if(detectObjDistance()>30)
    { go_Left();delay(100);
      go_Advance();delay(20);
      stop_Stop();delay(50);
      Serial.println(detectObjDistance());
    }
    else if(detectObjDistance()>20 && detectObjDistance()<30)
```

```
   { go_Advance();delay(100);
     stop_Stop();delay(50);
     Serial.println(detectObjDistance());
   }
  }turnUSsensorMiddle();
}

bool detectObj()
{ digitalWrite(Trig_PIN,HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig_PIN,LOW);
  unsigned long t = pulseIn(Echo_PIN,HIGH); //microseconds
  float d = 340.0*(t/1000000.0)/2 *100; //cm
  Serial.println(d);
  if(d<25)
  { return true;
  }
  else
  { return false;
  }
}

int detectObjDistance()
{ digitalWrite(Trig_PIN,HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig_PIN,LOW);
  unsigned long t = pulseIn(Echo_PIN,HIGH); //microseconds
  float d = 340.0*(t/1000000.0)/2 *100; //cm
  return d;
}

bool detectLineIRm()
{ int z = digitalRead(IRSensorM);
  if(z==1)
  { return true;
  }
  else
  { return false;
  }
}
```