#include <Servo.h>          //Servo motor library. This is standard library

#include <NewPing.h>        //Ultrasonic sensor function library. You must install this library

//our L298N control pins

const int LeftMotorForward = 7;

const int LeftMotorBackward = 6;

const int RightMotorForward = 4;

const int RightMotorBackward = 5;

//sensor pins

#define trig\_pin A1 //analog input 1

#define echo\_pin A2 //analog input 2

#define maximum\_distance 200

boolean goesForward = false;

int distance = 100;

NewPing sonar(trig\_pin, echo\_pin, maximum\_distance); //sensor function

Servo servo\_motor; //our servo name

void setup(){

  pinMode(RightMotorForward, OUTPUT);

  pinMode(LeftMotorForward, OUTPUT);

  pinMode(LeftMotorBackward, OUTPUT);

  pinMode(RightMotorBackward, OUTPUT);

  servo\_motor.attach(10); //our servo pin

  servo\_motor.write(115);

  delay(2000);

  distance = readPing();

  delay(100);

  distance = readPing();

  delay(100);

  distance = readPing();

  delay(100);

  distance = readPing();

  delay(100);

}

void loop()

{

  int distanceRight = 0;

  int distanceLeft = 0;

  delay(50);

  if (distance <= 20){

    moveStop();

    delay(300);

    moveBackward();

    delay(400);

    moveStop();

    delay(300);

    distanceRight = lookRight();

    delay(300);

    distanceLeft = lookLeft();

    delay(300);

    if (distance >= distanceLeft){

      turnRight();

      moveStop();

    }

    else{

      turnLeft();

      moveStop();

    }

  }

  else{

    moveForward();

  }

    distance = readPing();

}

int lookRight(){

  servo\_motor.write(50);

  delay(500);

  int distance = readPing();

  delay(100);

  servo\_motor.write(115);

  return distance;

}

int lookLeft(){

  servo\_motor.write(170);

  delay(500);

  int distance = readPing();

  delay(100);

  servo\_motor.write(115);

  return distance;

  delay(100);

}

int readPing(){

  delay(70);

  int cm = sonar.ping\_cm();

  if (cm==0){

    cm=250;

  }

  return cm;

}

void moveStop(){

  digitalWrite(RightMotorForward, LOW);

  digitalWrite(LeftMotorForward, LOW);

  digitalWrite(RightMotorBackward, LOW);

  digitalWrite(LeftMotorBackward, LOW);

}

void moveForward(){

  if(!goesForward){

    goesForward=true;

    digitalWrite(LeftMotorForward, HIGH);

    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);

    digitalWrite(RightMotorBackward, LOW);

  }

}

void moveBackward(){

  goesForward=false;

  digitalWrite(LeftMotorBackward, HIGH);

  digitalWrite(RightMotorBackward, HIGH);

  digitalWrite(LeftMotorForward, LOW);

  digitalWrite(RightMotorForward, LOW);

}

void turnRight(){

  digitalWrite(LeftMotorForward, HIGH);

  digitalWrite(RightMotorBackward, HIGH);

  digitalWrite(LeftMotorBackward, LOW);

  digitalWrite(RightMotorForward, LOW);

  delay(500);

  digitalWrite(LeftMotorForward, HIGH);

  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(LeftMotorBackward, LOW);

  digitalWrite(RightMotorBackward, LOW);

}

void turnLeft(){

  digitalWrite(LeftMotorBackward, HIGH);

  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(LeftMotorForward, LOW);

  digitalWrite(RightMotorBackward, LOW);

  delay(500);

  digitalWrite(LeftMotorForward, HIGH);

  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(LeftMotorBackward, LOW);

  digitalWrite(RightMotorBackward, LOW);

}