//HC SR-04-SERVO 3 Obstacle Avoidance Robot Buggy

//Project 1: 2WD Obstacle Avoiding Robot - Arduino Project Hub

//Modified by Pat McMahon 29/5/2021, distance<=35cm

//A057

#include <Servo.h> //Servo motor library

#include <NewPing.h> //Ultrasonic sensor function library

const int LeftForward = 11;// IN4 -L298N control pins

const int LeftBackward = 10;//IN3

const int RightForward = 13;//IN2

const int RightBackward = 12;//IN1

#define trig\_pin A1 //sensor pins - 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; //servo name

void setup() {

 pinMode(RightForward, OUTPUT);

 pinMode(LeftForward, OUTPUT);

 pinMode(LeftBackward, OUTPUT);

 pinMode(RightBackward, OUTPUT);

 servo\_motor.attach(7); //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 <= 35) {

 moveStop(); // obstacle probably on the route forward, so stop

 delay(300);

 moveBackward();

 delay(300);

 moveStop();

 delay(300);

 distanceRight = lookRight();

 delay(400);

 distanceLeft = lookLeft();

 delay(400);

 if (distanceRight >= distanceLeft) {

 turnRight(); // calculate in which direction the obstacle is more far

 delay(300);

 moveStop();

 }

 else {

 turnLeft();

 delay(300);

 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(RightForward, LOW);

 digitalWrite(LeftForward, LOW);

 digitalWrite(RightBackward, LOW);

 digitalWrite(LeftBackward, LOW);

}

void moveForward() {

 if (!goesForward) {

 goesForward = true;

 digitalWrite(LeftForward, HIGH);

 digitalWrite(RightForward, HIGH);

 digitalWrite(LeftBackward, LOW);

 digitalWrite(RightBackward, LOW);

 }

}

void moveBackward() {

 goesForward = false;

 digitalWrite(LeftBackward, HIGH);

 digitalWrite(RightBackward, HIGH);

 digitalWrite(LeftForward, LOW);

 digitalWrite(RightForward, LOW);

}

void turnRight() {

 digitalWrite(LeftForward, HIGH);

 digitalWrite(RightBackward, HIGH);

 digitalWrite(LeftBackward, LOW);

 digitalWrite(RightForward, LOW);

 delay(200);

 digitalWrite(LeftForward, HIGH);

 digitalWrite(RightForward, HIGH);

 digitalWrite(LeftBackward, LOW);

 digitalWrite(RightBackward, LOW);

}

void turnLeft() {

 digitalWrite(LeftBackward, HIGH);

 digitalWrite(RightForward, HIGH);

 digitalWrite(LeftForward, LOW);

 digitalWrite(RightBackward, LOW);

 delay(200);

 digitalWrite(LeftForward, HIGH);

 digitalWrite(RightForward, HIGH);

 digitalWrite(LeftBackward, LOW);

 digitalWrite(RightBackward, LOW);

}