***Ardiuno code***:

const int trigPin1 = 11;

const int echoPin1 = 10;

const int trigPin2 = A3; //13;

const int echoPin2 = A4;

const int trigPin3 = A2;

const int echoPin3 = A5;

const int in1 = 9;

const int in2 = 8;

const int in3 = 4;

const int in4 = 3;

const int enA = 5;

const int enB = 6;

#define PWM 100

#define DIS 5

void setup()

{

pinMode(trigPin1, OUTPUT);

pinMode(echoPin1, INPUT);

pinMode(trigPin2, OUTPUT);

pinMode(echoPin2, INPUT);

pinMode(trigPin3, OUTPUT);

pinMode(echoPin3, INPUT);

pinMode (in1, OUTPUT);

pinMode (in2, OUTPUT);

pinMode (in3, OUTPUT);

pinMode (in4, OUTPUT);

pinMode (enA, OUTPUT);

pinMode (enB, OUTPUT);

}

void loop()

{

if ( FrontSensor() < DIS && RightSensor () <DIS && LeftSensor

()<DIS) // obstacle infront of all 3 sides

{

turn\_right ();

delay(600);

//then reverse

}

else if (FrontSensor() <DIS && RightSensor () <DIS && LeftSensor

()>DIS) // obstacle on right and front sides

{

turn\_left ();

// turn left side

}

else if (FrontSensor() <DIS && RightSensor () >DIS && LeftSensor

()<DIS) // obstacle on left and front sides

{

turn\_right ();

* turn right side

}

else if (FrontSensor() <DIS && RightSensor () >DIS && LeftSensor

()>DIS) // obstacle on front sides

{

turn\_right ();

* then turn right

}

else if (FrontSensor() >DIS && RightSensor () >DIS && LeftSensor

()<DIS) // obstacle on left sides

{

turn\_right ();

* then turn right and then forward

delay(600);

forward();

}

else if (FrontSensor() >DIS && RightSensor () <DIS && LeftSensor

()>DIS) // obstacle on right sides

{

turn\_left ();

* then turn left and then forward

delay(600);

forward();

}

else

{

forward();

}

}

void forward ()

{

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

analogWrite(enA, PWM);

analogWrite(enB, PWM);

}

void turn\_left ()

{

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

analogWrite(enA, PWM);

analogWrite(enB, PWM);

}

void turn\_right ()

{

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

analogWrite(enA, PWM);

analogWrite(enB, PWM);

}

void reverse ()

{

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

analogWrite(enA, PWM);

analogWrite(enB, PWM);

}

void stop()

{

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

analogWrite(enA, LOW);

analogWrite(enB, LOW);

}

long FrontSensor ()

{

long dur;

digitalWrite(trigPin1, LOW);

delayMicroseconds(5); // delays are required for a succesful sensor operation.

digitalWrite(trigPin1, HIGH);

delayMicroseconds(10); //this delay is required as well!

digitalWrite(trigPin1, LOW);

dur = pulseIn(echoPin1, HIGH);

return (dur/60);// convert the distance to centimeters.

}

long RightSensor ()

{

long dur;

digitalWrite(trigPin2, LOW);

delayMicroseconds(5); // delays are required for a succesful sensor operation.

digitalWrite(trigPin2, HIGH);

delayMicroseconds(10); //this delay is required as well!

digitalWrite(trigPin2, LOW);

dur = pulseIn(echoPin2, HIGH);

return (dur/62);// convert the distance to centimeters.

}

long LeftSensor ()

{

long dur;

digitalWrite(trigPin3, LOW);

delayMicroseconds(5); // delays are required for a succesful sensor operation.

digitalWrite(trigPin3, HIGH);

delayMicroseconds(10); //this delay is required as well!

digitalWrite(trigPin3, LOW);

dur = pulseIn(echoPin3, HIGH);

return (dur/50);// convert the distance to centimeters.

}

***GITHUB LINK:***

https://github.com/Kawshik1702121/Final-Project-report-maze-solving-robot-and-code-for-this-