**LINE FOLLOWER ROBOT**

**A MINI PROJECT REPORT**

Submitted towards the professional course

**15Z601 EMBEDDED SYSTEMS LABORATORY**

Harhi Varrman V (17Z318)

Johann Andre Sebastain A (17Z322)

Sashwath K G (17Z345)

Bachelor OF ENGINEERING

Branch: Computer Science and Engineering



**FEBRUARY 2020**

***Department of Computer Science and Engineering***

***PSG College of Technology,***

***Coimbatore - 641 004.***

# INTRODUCTION:

A line follower robot is essentially created to trail a line or path. A simple line/path or as compound path marking structures can be used. Numerous sensing schemes can be used in order to sense these markers or lines. The sensing accuracy and flexibility is determined by the kind of schemes to be used. Sensor positioning also plays a dynamic role in improving the robot’s performance. Primarily, a line-following robot is a self-moving robot that detects and follows a line or a path. The path is designed by using a black line on a white surface.

# PROBLEM STATEMENT:

Detecting a line and steering the robot to stay on path, while continuously adjusting mistaken moves using feedback mechanism forms an unpretentious yet effective closed loop system. This is basically creating the human emotion of “Responding to Stimulus” in a robot.

Practical applications of a line follower: Automated Cars – Tesla

Manufacturing Unit - Tesla

# Components Required:

* Arduino UNO
* L293D Motor Driver IC
* Motor Shield
* DC Motors x 2
* Robot Chassis
* IR Sensor Module x 2
* Black Tape (Electrical Insulation Tape)
* Connecting Wires
* Power supply
* Battery Connector
* Battery Holder

# Schematic Diagram:

# 

# Code:

int EA = 9; //ENA connected to digital pin 9

int EB = 3; //ENB connected to digital pin 3

int MA1 = 7; // MOTOR\_A1 connected to digital pin 7

int MA2 = 6; // MOTOR\_A2 connected to digital pin 6

int MB1 = 5; // MOTOR\_B1 connected to digital pin 5

int MB2 = 4; / / MOTOR\_B2 connected to digital pin 4

int RI = A0; // RIGHT sensor connected to analog pin A0

int LF = A1; // LEFT sensor connected to analog pin A1

void setup() {

pinMode(EA, OUTPUT); // initialize ENA pin as an output

pinMode(EB, OUTPUT); // initialize ENB pin as an output

pinMode(MA1, OUTPUT); // initialize MOTOR\_A1 pin as an output

pinMode(MA2, OUTPUT); // initialize MOTOR\_A2 pin as an output

pinMode(MB1, OUTPUT); // initialize MOTOR\_B1 pin as an output

pinMode(MB2, OUTPUT); // initialize MOTOR\_B2 pin as an output

pinMode(RI, INPUT); // initialize RIGHT pin as an input

pinMode(LF, INPUT); // initialize ENA pin as an input

}

void loop() {

if (analogRead(RI)<=35 && analogRead(LF)<=35) //compare the both sensor to decide the direction

{

analogWrite(EA, 100); // set right motors speed

analogWrite(EB, 100); // set left motors speed

//run right motors clockwise

digitalWrite(MA1, LOW);

digitalWrite(MA2, HIGH);

//run left motors clockwise

digitalWrite(MB1, HIGH);

digitalWrite(MB2, LOW);

}

else if (analogRead(RI)<=35 && !analogRead(LF)<=35) //compare the both sensor to decide the direction

{

analogWrite(EA, 255); //set right motors speed

analogWrite(EB, 255); //set left motors speed

//run right motors clockwise

digitalWrite(MA1, LOW);

digitalWrite(MA2, HIGH);

//run left motors anti-clockwise

digitalWrite(MB1, LOW);

digitalWrite(MB2, HIGH);

}

else if (!analogRead(RI)<=35 && analogRead(LF)<=35) //compare the both sensor to decide the direction

{

analogWrite(EA, 255); //set right motors speed

analogWrite(EB, 255); //set left motors speed

//run right motors anti-clockwise

digitalWrite(MA1, HIGH);

digitalWrite(MA2, LOW);

//run left motors clockwise

digitalWrite(MB1, HIGH);

digitalWrite(MB2, LOW);

}

else if (!analogRead(RI)<=35 && !analogRead(LF)<=35) //compare the both sensor to decide the direction

{

analogWrite(EA, 0); //set right motors speed

analogWrite(EB, 0); //set left motors speed

//stop right motors

digitalWrite(MA1, LOW);

digitalWrite(MA2, LOW);

//stop left motors

digitalWrite(MB1, LOW);

digitalWrite(MB2, LOW);

}

}

# Challenges Faced:

1. Determining the position of sensors.
2. Soldering the Components perfectly

# Contribution of Team Members:

1. Building the Robot – Harhi Varrman and Johann
2. Implementation – Sashwath

# References:

[1] Pakdaman, M. and Sanaatiyan, M.M., 2009, December. Design and implementation of line follower robot. In 2009 Second International Conference on Computer and Electrical Engineering (Vol. 2, pp. 585-590). IEEE.

[2] Bajestani, S.E.M. and Vosoughinia, A., 2010, August. Technical report of building a line follower robot. In 2010 International Conference on Electronics and Information Engineering (Vol. 1, pp. V1-1). IEEE.

[3] Pakdaman, M., Sanaatiyan, M.M. and Ghahroudi, M.R., 2010, February. A line follower robot from design to implementation: Technical issues and problems. In 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE) (Vol. 1, pp. 5-9). IEEE.

[4] Chowdhurya, N.H., Khushib, D. and Rashidc, M.M., 2017. Algorithm for line follower robots to follow critical paths with minimum number of sensors. International Journal of Computer, 24(1), pp.13-22.

[5] https://www.scribd.com/doc/59265674/Line-Follower-Report