Line following robot

under the guidence of r.e.c

R.E.C robotics and electronics club works under the Technical board of Indian institute of information technology bhagalpur

By:-

|  |  |  |
| --- | --- | --- |
|  |  |  |
| AYUSH  ABHIJEET AWASHTHI  DHRUV MISHRA |  |  |
|  |  |  |

Contents

[what is LFR ? 1](#_Toc139532087)

[Working of line following robot . 2](#_Toc139532088)

[Navigate the LFR 3](#_Toc139532089)

[Circuit diagram 4](file:///C:\Users\Ayush\Downloads\lfr.docx#_Toc139532090)

[circuit and components 4](#_Toc139532091)

[Components 5](#_Toc139532092)

[Solving maze 6](#_Toc139532093)

[Final result 12](#_Toc139532094)

[Contact Information 13](#_Toc139532095)

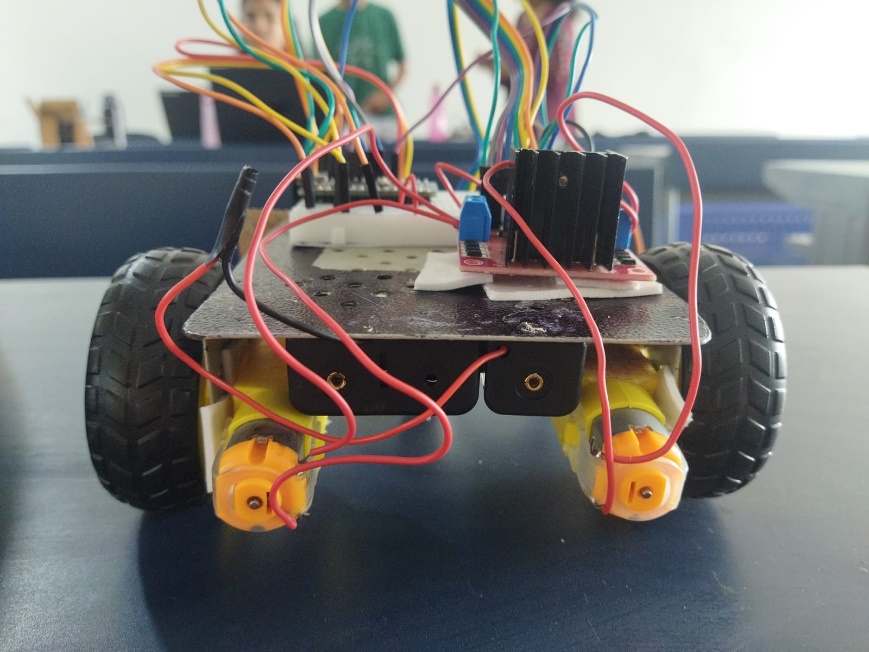
# what is LFR ?

“we will learn how to build a black line follower robot using Arduino Uno and some easily accessible components”

## LINE FOLLOWING ROBOT

### **Line Follower Robot** (**LFR**) is a simple autonomously guided robot that follows a line drawn on the ground to either detect a dark line on a white surface or a white line on a dark.

## Operating Highlights

In order to follow a line, robot must detect the line first as We all know that the reflection of light on the white surface is maximum and minimum on the black surface because the black surface absorbs maximum amount of light. So, we are going to use this property of light to detect the line.

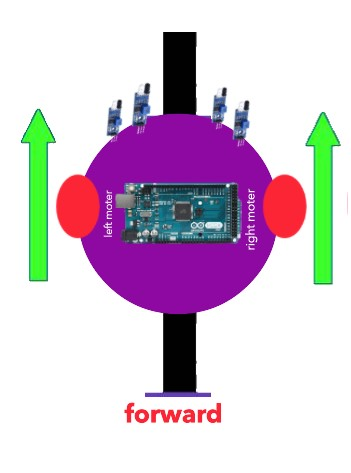
# Working of line following robot .

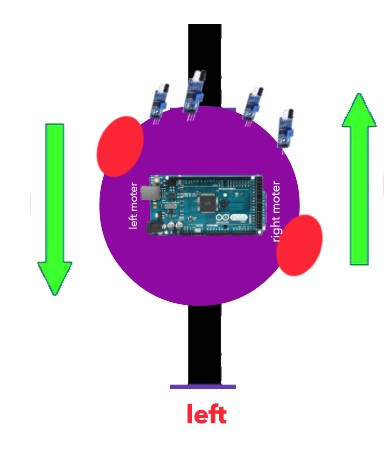
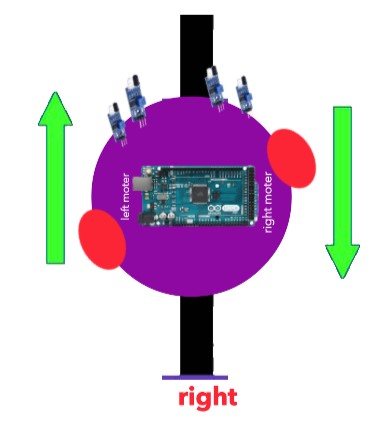
As stated earlier, line follower robot (LFR) follows a line, and in order to follow a line, robot must detect the line first.

So, we are going to use this property of light to detect the line. To detect light, either LDR (light-dependent resistor) or an IR sensor can be used. For this project, we are going with the IR sensor because of its higher accuracy. To detect the line, we place two IR sensors two on the left and other two on the right side of the robot as marked in the diagram below.

Infrared sensors consist of two elements, a transmitter and a receiver. The transmitter is basically an IR LED, which produces the signal and the IR receiver is a photodiode, which senses the signal produced by the transmitter. The IR sensors emits the infrared light on an object, the light hitting the black part gets absorbed thus giving a low output but the light hitting the white part reflects back to the transmitter which is then detected by the infrared receiver, thereby giving an analog output. Using the stated principle, we control the movement of the robot by driving the wheels attached to the motors, the motors are controlled by a microcontroller.

# Navigate the LFR

A line follower robot has two sets of motors, let's call them left motor and right motor. Both motors rotate  on the basis of the signal received from the IR sensors respectively. The robot needs to perform 4 sets of motion which includes moving forward, turning left, turning right and coming to a halt. The description about the cases are given below.

In this case, when both the sensors are on a white surface and the line is between the two sensors, the robot should move forward, i.e., both the motors should rotate such that the robot moves in forward direction.

In this case, the left two sensor is on top of the dark line, whereas the right two sensor is on the white part, hence the sensor detects the black line and gives a signal, to the microcontroller. Since, signal comes from the left sensors, the robot should turn to the left direction. Therefore, the left motor rotates backwards and the right motor rotates in forward direction. Thus, the robot turns towards left side.

This case is similar to the left case, but in this situation only the right sensors detects the line which means that the robot should turn in the right direction. To turn the robot towards the right direction, the left motor rotates forward and the right motor rotates backwards and as a result, the robot turns towards the right direction.

In this case, all the sensors are on top of the line and they can detect the black line simultaneously, the microcontroller is fed to consider this situation as a process for halt. Hence, both the motors are stopped, which causes the robot to stop moving.

# circuit and components

# Circuit diagram

# Components

1.Arduino Uno

2.2\*Dc gear moter

3.4\*Ir sensor

4.L293D moter driver

5.2\*Smart Robot car tyres

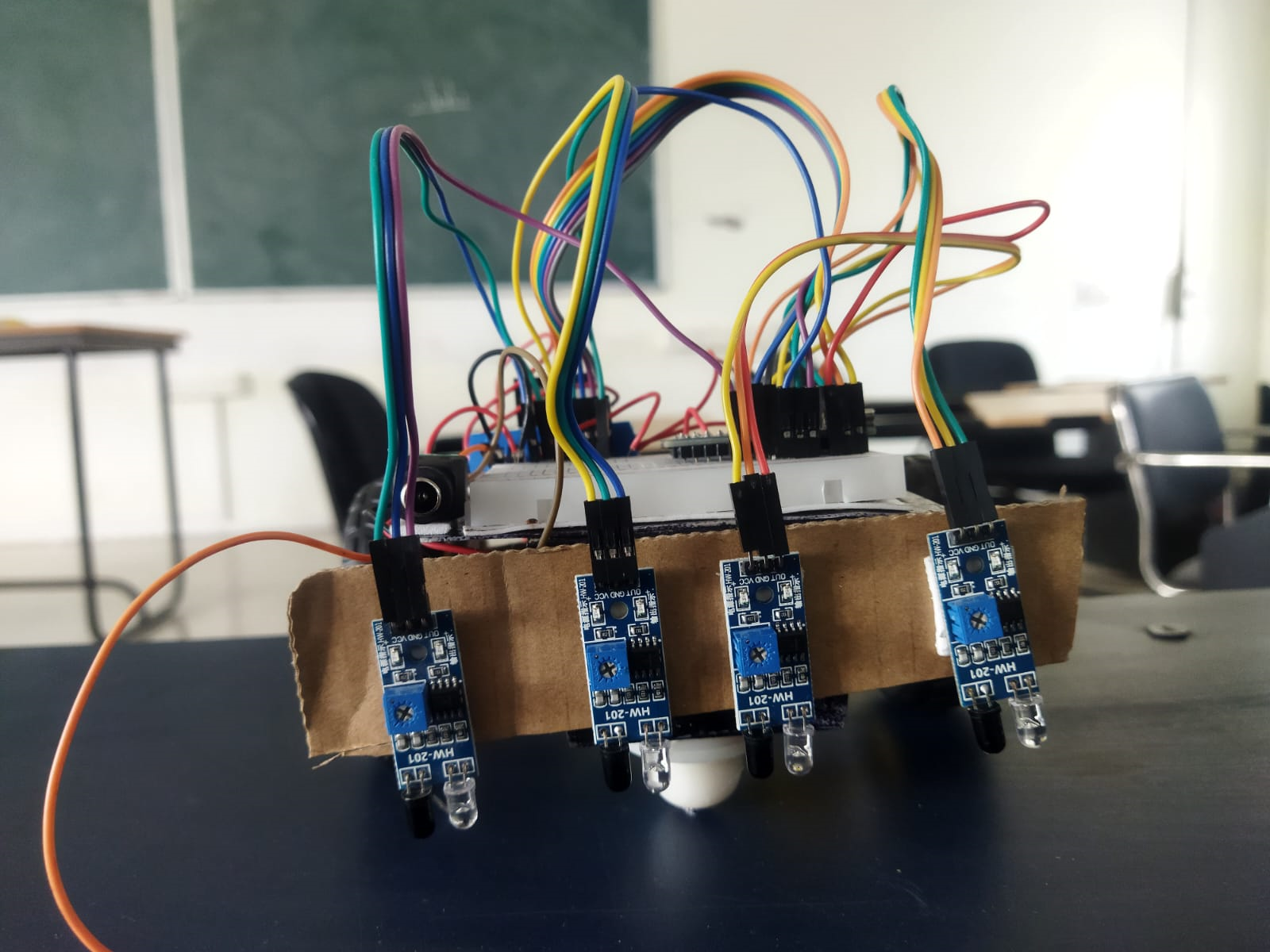
6.caster wheel

7.on/off switches.

8.4\*battery cell 3.7v and 4\*battery holder

9.Acrylic sheet(r= 10cm)circular.

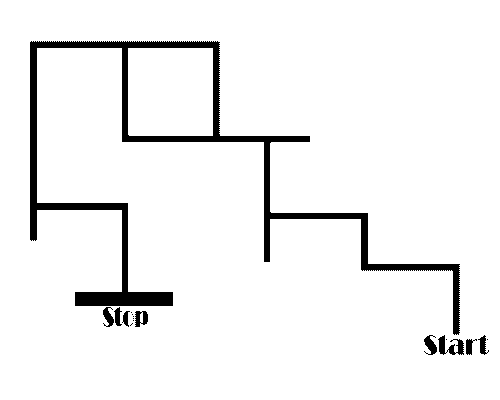
10.Jumper wires(mm,fm,ff).



# Solving maze

## ****Maze****

A **maze**is a tour puzzle in the form of a complex branching passage through which the solver must find a route. In everyday speech, both maze and labyrinth denote a complex and confusing series of pathways, but technically the maze is distinguished from the labyrinth, as the labyrinth has a single through-route with twists and turns but without branches, and is not designed to be as difficult to navigate.

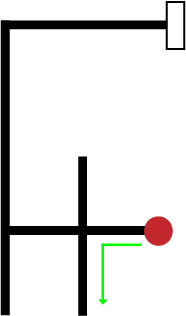


## LSRB Algorithm

This LSRB algorithm can be simplified into these simple conditions:

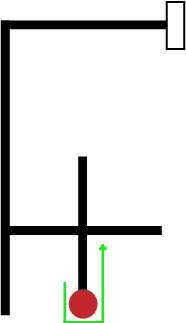
1. If you can turn left then go ahead and turn left,
2. else if you can continue driving straight then drive straight,
3. else if you can turn right then turn right.
4. If you are at a dead end then turn around.  
     
   The expunction of LSRB is shown below:
5. L- Left
6. R- Right
7. S- Straight
8. B- Turning around (Back).

The robot has to make these decisions when at an intersection. An intersection is any point on the maze where you have the opportunity to turn. If the robot comes across an opportunity to turn and does not turn then this is considered going straight. Each move was taken at an intersection or when turning around has to be stored.

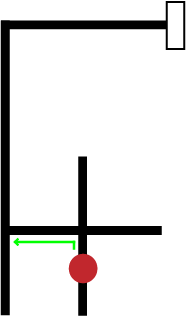
At first, we see the searching process of the LSRB Algorithms. the diagrams are the sample of the processing the LSRB.

The first diagram denotes the starting point and the first decision based on the LSRB priority so the robot is choosing the left side. Second decision Value has been stored in the register. The first decision diagram is shown,

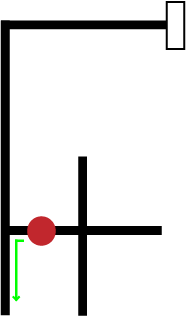
Path = L

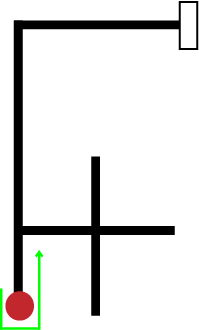
In the second diagrams is denoted the after processing of the first decision and the next decision is taken to the back side because there are no more options are available. The second decision diagram is shown,

Path = LB

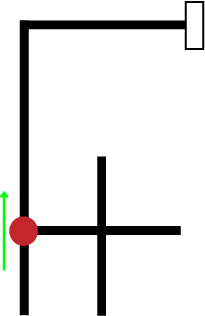
In the Third diagrams is denoted the after processing of the second decision and the next decision is taken to the left side because there are no more options are available and the second decision Value has been stored in the register. The third decision diagram is shown,

Path = LBL

In the Fourth diagrams is denoted the after processing of the third decision and the next decision is taken to the left side because there are no more options are available and the fourth decision Value has been stored in the register. The fourth decision diagram is shown,

In the fifth diagrams is denoted the after processing of the fourth decision and the next decision is taken to the back side because there are no more options are available and the fifth decision Value has been stored in the register. The fifth decision diagram is shown,

Path = LBLL

In the sixth diagrams is denoted the after processing of the fifth decision and the next decision is taken to the straight side because there are no more options are available and the sixth decision Value has been stored in the register. The sixth decision diagram is shown,

Path = LBLLB

In the seventh diagrams is denoted the after processing of the sixth decision and the next decision is taken to the right side because there are no more options are available and the seventh decision diagram is shown below, After the seventh step the searching processes will be finished and then finally the “LBLLBSR” value is stored into the register.The second process is the travelling processes. This process is performing the robot to go the destination without any searching and using the shortest path. The travelling is simplifying the register value using the following equation and shorted and follow the value of the register the equations are shown below,

Path = LBLLBS

LBR = B

LBS = R

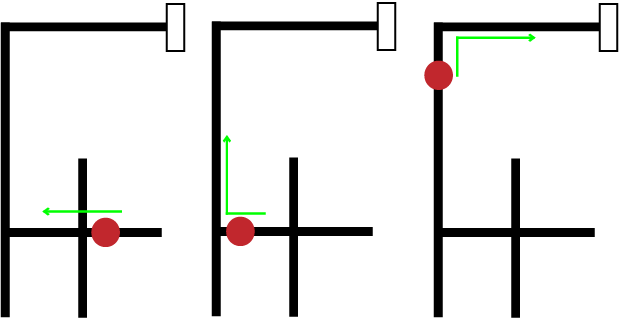
RBL = B

SBL = R

SBS = B

LBL = S

The register value is shorted from using the equations and the final register value is diagram is shown below

Path = S **→** SR **→** SRS

## Coding

https://www.tinkercad.com/things/3cUr0JqLSJ5

# Final result

# Contact Information

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name AYUSH**  **Tel 9005956767**  **EMAIL**  Ayush.2201188me@iiitbh.ac.in |  | **Name ABHIJEET AWASTHI**  **Tel** 7607321181  **EMAIL**  Abhijeet.2201164ec@iiitbh.ac.in |
|  |  |  |  |
|  | **Name DHRUV MISHRA**  **Tel** 8097832127  **EMAIL**  Dhruv.2201066me@iiitbh.ac.in |  |  |