**LINE FOLLOWER ROBOT**

****

**BAJAJ INSTITUTE OF TECHNOLOGY**

**WARDHA-442 001 (INDIA)**

**2022 - 23**

**LINE FOLLOWER ROBOT**

***Project report submitted to***

***Dr. Babasaheb Ambedkar Technological University, Maharashtra***

***in partial fulfilment of the requirements for the award of***

***the degree***

**Bachelor of Technology**

**In**

**Mechanical Engineering**

*By*

**Tejas Waghmare (2046491612030)**

**Vaishnavi Zade (2046491612033)**

**Rushab Dhole (2046491612026)**

**Manish Lakhe (2046491612012)**

under the guidance of

**Dr. M. D. Pasarkar**

**Associate Professor**

****

**Department of Mechanical Engineering**

**Bajaj Institute of Technology**

**Wardha-442 001 (India)**

**2022-23**

**BAJAJ INSTITUTE OF TECHNOLOGY, WARDHA**

**DEPARTMENT OF MECHANICAL ENGINEERING**

**CERTIFICATE**

****

***This is to certify that the Project report titled***

***LINE FOLLOWER ROBOT* *has been successfully completed***

***by***

**Tejas Waghmare (2046491612030)**

**Vaishnavi Zade (2046491612033)**

**Rushab Dhole (2046491612026)**

**Manish Lakhe (2046491612012)**

***in partial fulfilment of the requirements for the award of***

***the degree in Bachelor of Technology, Mechanical Engineering***

|  |  |
| --- | --- |
| **Dr. M. D. Pasarkar**  **(Associate professor)**  **Guide**  Department of Mechanical Engineering  Bajaj Institute of Technology, Wardha | |
| **Dr. D. V. Bhope**  **Head of the Department**  Department of Mechanical Engineering  Bajaj Institute of Technology, Wardha | **Dr. N. M. Kanhe**  **Principal**  Bajaj Institute of Technology,  Wardha |
|  | |

**Date: / /**

**Place: Wardha**

**DECLARATION**

I/We, hereby declare that the project report titled “LINE FOLLOWER ROBOT” submitted by me to the Bajaj Institute of Technology, Wardha ,in partial fulfilment of the requirement for the award of Degree of B. Tech. in Mechanical Engineering discipline is a record of bonafide project work carried out by me under the guidance of **Dr. M. D. Pasarkar**

I/We, further declare that this submission by the undersigned represents my/our original work and I/We have quoted the references where others ideas/words have been included. I understand any violation of the above will levy a disciplinary action on me/us.

I/We, further declare that the work reported in this project report has not been submitted either in-part or in-full for the award of any other degree in any other Institute or University.

Date :- / /

Place :- Wardha

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Name of Student** | **PRN Number** | **Signature** |
|  | Tejas Waghmare | 2046491612030 |  |
|  | Vaishnavi Zade | 2046491612033 |  |
|  | Rushab Dhole | 2046491612026 |  |
|  | Manish Lakhe | 2046491612012 |  |

**ACKNOWLEDGEMENT**

We would like to thank Dr. M. D. Pasarkar Sir our project guide and Dr. D. V. Bhope Sir Head of the Department of Mechanical Engineering as well as the Principal, Dr. N. M. Kanhe Sir for their support and guidance in completing our project on the topic **“LINE FOLLOWER ROBOT”** It was a great learning experience.

Your valuable guidance and suggestions helped us in various phases of the completion of this project. We will always be thankful to you in this regard. We are ensuring that this project was finished by us and not copied.

Date: / /

Place: Wardha

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Name of Student** | **PRN Number** | **Signature** |
|  | Tejas Waghmare | 2046491612030 |  |
|  | Vaishnavi Zade | 2046491612033 |  |
|  | Rushab Dhole | 2046491612026 |  |
|  | Manish Lakhe | 2046491612012 |  |

**ABSTRACT**

The line follower robot is an autonomous robotic system designed to navigate and follow a predefined path marked by contrasting lines on the surface. This project presents the development of a line follower robot equipped with five infrared (IR) sensors for efficient and precise line tracking. The IR sensors are strategically positioned on the robot to detect the line's position relative to the robot's center.The primary objective of this project is to create a robot capable of accurately tracking complex paths with varying curvature and sharp turns. The five IR sensors allow the robot to gather more data points along the path, enhancing its ability to follow intricate lines accurately. The sensor data is processed in real-time, enabling the robot to make rapid decisions and adjust its motion accordingly.The robot's control algorithm utilizes the input from the IR sensors to calculate the error between the desired path and the robot's actual position.This abstract outlines the design, construction, and testing phases of the line follower robot. The mechanical design incorporates a lightweight and compact chassis for improved maneuverability. The sensors calibration process are detailed to optimize the robot's tracking performance.Extensive testing is conducted on various line patterns and complex paths to evaluate the robot's performance. The experimental results demonstrate that the line follower robot equipped with five IR sensors exhibits exceptional accuracy and stability in following intricate lines. It effectively adapts to changes in line width and maintains reliable tracking even under challenging conditions.In conclusion, the line follower robot with five IR sensors offers a cost-effective and robust solution for precise line tracking in diverse scenarios. The implementation of multiple IR sensors enhances the robot's responsiveness and tracking capabilities, making it a versatile platform for applications in robotics competitions, industrial automation, and educational purposes.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter**  **No.** | **Title** | **Page No.** |
|  |  |  |
|  | Certificates | 1-3 |
|  | Declaration | 4 |
|  | Acknowledgement | 5 |
|  | Abstract | 6 |
|  | Table of Contents | 7 |
|  | Abbreviations | 8 |
|  | List of symbols | 9 |
|  |  |  |
| **1** | **Introduction** | 10-11 |
| **2** | **Literature Survey** | 12 |
|  | 2.1 Problem Statement |  |
|  | 2.2 Objectives |  |
| **3** | **Methodology & System Requirements and Specifications** | 13-23 |
| **4** | **Implementation of Project** | 24-36 |
| **5** | **Results and Discussions** | 37 |
| **6** | **Conclusions and Future Scope** | 38-39 |
|  | 6.1 Conclusions |  |
|  | 6.2 Future Scope |  |
| **7** | **References** | 40 |
|  |  |  |

**ABBREVIATIONS**

**Examples**

|  |  |
| --- | --- |
| *LFR* | Line Follower Robot |

**LIST OF SYMBOLS**

**Samples**

|  |  |
| --- | --- |
| S1 | Extreme left sensor |
| *S2* | Left sensor |
| *S3* | Middle sensor |
| *S4* | Right sensor |
| *S5* | Extreme right sensor |

**CHAPTER 1**

**INTRODUCTION**

The line follower robot is a fascinating and widely explored application in the field of robotics, attracting enthusiasts, hobbyists, and researchers alike. It involves designing an autonomous robot capable of tracking and following a predefined path marked by contrasting lines on the surface. With advancements in sensor technology and microcontrollers, the line follower robots have become more sophisticated, exhibiting enhanced accuracy and adaptability to various line patterns.

This project presents the development of a line follower robot equipped with five infrared (IR) sensors, an Arduino Uno R3 microcontroller, an L298 motor driver module, two DC motors, and a well-designed chassis. The integration of these components allows the robot to execute the required motions, including forward, right and left turns, and u-turn, while accurately tracing the designated path.

The primary objective of this project is to create a versatile line follower robot capable of navigating complex routes with varying curvature and sharp turns. The use of five IR sensors provides the robot with a wider field of perception, enabling it to gather more data points along the path. This enhanced sensor array facilitates smoother and more precise tracking, even when encountering intricate line patterns.

The Arduino Uno R3 microcontroller serves as the brain of the robot, responsible for processing sensor data, implementing control algorithms, and coordinating the motor movements. Its flexibility and ease of programming make it an ideal choice for this application. The L298 motor driver module acts as the interface between the Arduino and the two DC motors, granting the robot the ability to execute multi-directional motions with varying speeds.The chassis of the robot is carefully designed to ensure stability, maneuverability, and easy integration of the electronic components. The combination of well-aligned wheels and motors allows the robot to perform the required motions, including forward movement, precise turns, and u-turns, contributing to its efficient line following capability.

Throughout this project, we will delve into the mechanical design of the robot, the calibration and positioning of the IR sensors, the development of the control algorithm, and the integration of the hardware components. Extensive testing and performance evaluation will be conducted to validate the robot's ability to accurately track diverse line patterns and navigate complex paths with agility and stability.

This project aims to present a comprehensive approach to building an advanced line follower robot with five IR sensors, an Arduino Uno microcontroller, an L298 motor driver module, two DC motors, and a well-designed chassis. The robot's ability to execute forward, right and left turns, and u-turn while accurately following lines opens up possibilities for various practical applications, including robotics competitions, industrial automation, and educational purposes. With the fusion of electronics, mechanics, and control algorithms, this project exemplifies an intriguing integration of disciplines in the field of robotics.

**CHAPTER 2**

**LITERATURE SURVEY**

* We observed that the Existing line follower robots in the market can only follow one line either a Black or White line only in a repetitive manner and lack the ability to take varying degrees of turns.
* Some line follower robots can track both black and white lines & they can take varying degrees of turns, but they fail to take U-turns and reverse on the same path.
* So we trying to fulfill this gap, therefore we aim to design and develop a line follower robot that possesses all the existing and new features like Shifting the following path & taking U-turns, and reverse.
* Our line follower robot will have the capability to shift its path and follow both black and white lines while taking different possible degrees of turns.
* Additionally, it will also be able to follow the reverse path by taking U-turns, making it a versatile and efficient solution for line-tracking applications.

**2.1 PROBLEM STATEMENT:**

A line follower robot which identifies both black and white surface to execute given motion like forward motion , right and left turn.

**2.2 OBJECTIVES:**

* The robot must be capable of following a black surface or white surface.
* It should be capable of taking various turns like left,right and u-turn.
* It should be able to change path from black line to white line or vice-versa.
* The robot must be insensitive to environmental factors such as lighting and noise.
* It must allow calibration of the line's darkness threshold.
* Scalability must be a primary concern in the design.

**CHAPTER 3**

**METHODOLOGY & SYSTEM REQUIREMENTS AND SPECIFICATIONS**

Line follower robot is a robot which follows both black and white surface /line to execute the needed motion.

To make a line follower we need components which meet our requirements. So for recognizing the surfaces we need IR sensors to detect the surface, for interpreting the sensors data and process the received information which can coded as per our need , we need Arduino Uno R3. When the sensors data match the condition of code which is given to Arduino Uno R3,it need to be executed to get needed motion, so here we used L298 module which also connected to motors for controlling speed and direction of rotation of motor.So for fixing all these components on one board, we need a chassis to fixed all these components.

**Methodology of Line follower Robot Contains:**

1. First we have arranged all these components which are needed for our line follower robot.
2. By considering all the components dimension, we have design the LFR chassis which has optimal size and strong enough to bear the load of each components.
3. All the components are fixed on chassis and connected to the controller with the help of jumper wire.
4. Black & white Path making.
5. Implementation of logical programming code .
6. IR Sensor Calibration.
7. Test and Debug.
8. Handling Motions and Turns.
9. **First we have arranged all these components which are needed for our line follower robot:**

The components are:

1. Arduino
2. 5 channel IR sensor Array
3. L298 Motor Driver
4. N20 6v-600rpm Motor
5. Castor wheel & Wheel having Grip.
6. Battery (12 volt)
7. Jumper Wires.
8. **IR SENSOR ARRAY:**

An IR array is combined set of six IR sensors & having six input and output pins for controlling. Here five sensors pins are to be used as an out pin towards Arduino to detect the path, one input pin for power supply and one output pin for ground. The IR array is mounted on the front part of chassis as the surfaces must be detected first to perform particular motion. The spacing between IR sensors is 9.525 mm as four sensors are aligned and one sensor is in front of array ,as this combination is useful in turning line follower. It detects both black surface as well as white surface to follow the given path.We selected this IR array as IR sensors are internally connected and it also reduced the need for separately implementation of sensors. The specifications of IR array is given below.

Input voltage : 3.0-5.5v

Optimal sensing distance: 3mm

Max. recommended sensing distance: 6mm

Distance between two IR sensors on array is 9.525mm

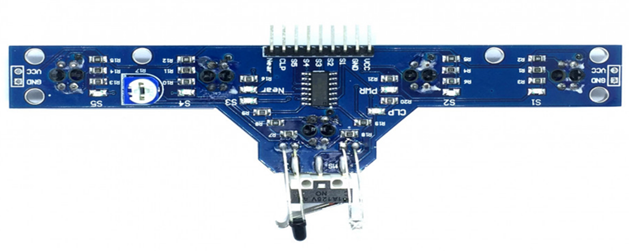
Length: 13mm

Width: 5mm

Height: 2mm

Weight: 12grams

The IR sensor consists of a transmitter and a receiver which works on principle of transmitting IR rays towards the surface which is then reflected back and detected by receiver making difference in output signal of array which is then send towards Arduino. If the surface is black, then the output difference created will be less. The main function of IR array is to detect the surface and send signal to Arduino Uno to command as per given input condition.

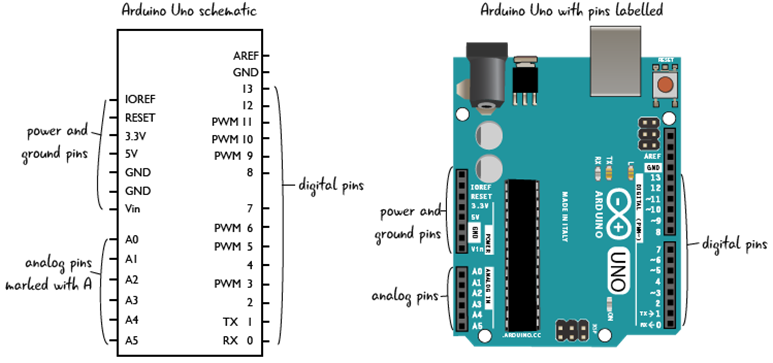


**Fig.3.1**

1. **ARDUINO:**

In this project we are using Arduino Uno R3. Arduino UNO is a micro controller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.It is a programmable board which is programmed in IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms. The [IDE](https://www.javatpoint.com/arduino-ide) is common to all available boards of Arduino. We are using Arduino Uno R3 based on our requirements. As Arduino Uno R3 have 14 digital I/O pins and 6 analog I/O pins which is sufficient for coordinating with given components. Arduino is a programmable board which is programmed through IDE stands for Integrated Development Environment. We used IDE software in our laptop to write c code for our given problem statement to be run in Arduino Uno R3.The code is uploaded in Arduino Uno through USB port from IDE software.

The main function of Arduino Uno in this project is to take input signal from IR array which is connected to the analog pins of Arduino Uno from A0-A4.The data from ir array are generally in the form of voltage difference as 0 or 1. 1 for white surface and 0 for black surface.The micro controller processed the input data from array to send command to Motor Driver Module which is connected to digital pin of Arduino Uno from 2-4 and 9-10 for controlling the speed of motor. As the input data match the given coded condition, the signal is then send to motor driver module for execution of given motion to motor. For maintain proper function of all the components, all ground connection of components are made common.



**Fig.3.2**

1. **Motor Driver Module L298:**

L298N module is a high voltage, high current dual full-bridge motor driver module for controlling DC motor and stepper motor. We have selected L298 module as it can control both the speed and rotation direction of two DC motors. The logical voltage is 5v. This module consists of an L298 dual-channel H-Bridge motor driver IC and uses two techniques for the control of speed and rotation direction of the DC motors. The digital pins of Arduino Uno is connected to IN1, IN2, IN3, IN4 of L298 module which work on H-Bridge for controlling rotation direction and PWM pins of Arduino is connected to ENB1 and ENB2 pin of L298 module for controlling speed of tires. L298n motor driver module uses the H-Bridge technique to control the direction of rotation of a DC motor. In this technique, H-Bridge controlled DC motor rotating direction by changing the polarity of its input voltage. An H-Bridge circuit contains four switching elements, like transistors (BJT or MOSFET), with the motor at the centre forming an H-like configuration. Input**IN1, IN2, IN3, and IN4** pins control the **switches**of the H-Bridge circuit inside L298N IC. These modules can control two DC motor at the same time.

A picture containing electronics, circuit

Description automatically generatedThe 12 V power supply is provided to its Vin for operating it. The output power supply of module is also given towards the Arduino Uno for functioning. The out pin OUT1, OUT2, OUT3 and OUT4 of L298 module is connected to two motors for controlling its speed and direction of motor.

**Fig.3.3**

1. **MOTOR:**

In this project we are using n20-6v-600rpm micro gear motor. As per our requirements motor should have high torque for quick start of line follower robot. We are using two motors which will be mounted on back corner ends of chassis. The N20 Micro Gear 6V 600 RPM DC Motor (High Torque) is lightweight, high torque, and low RPM motor. It is equipped with gearbox assembly to increase the torque of the motor. It has a cross-section of 10 × 12 mm, and the D-shaped gearbox output shaft is 9 mm long and 3 mm in diameter. It has a very small size so as fit in complex spaces of small-scale application. One can connect this Micro Gear Motor to wheels to drive them from one place to other while carrying high loads. The output pin 1,2,3,4 of L298 module are connected to this motors to execute given motions.

Diagram, engineering drawing

Description automatically generated

**Fig.3.4**

1. **WHEEL:**

In this project we are using two types of wheels - ball caster wheel and plastic wheel. The wheel as shown in Fig 3.9 is a plastic material wheel having rubber grip surrounded to it. The outer diameter is 43mm and thickness is 20mm and can bear weight upto 5 kg. The number of wheels used is 2 which is mounted on motor which is mounted on corner side of chassis.Due to weight bearing capacity and grip of rubber material , we are using it.

The ball caster wheel is also used in this line follower which is mounted on front part of chassis to support front weight alongside plastic grip wheel/tire. The caster wheel material is steel which is driven by back wheels of robot. It can bear weight upto 5kg/wheel having 3 mounting holes with wheel height 18mm and base material of 27.5mm.

A picture containing gear, metalware, wheel

Description automatically generated

**Fig.3.5 Fig3.6**

1. **BATTERY:**

In this project we are using 3.7 volt rechargeable lithium-ion cell.The number of cells used is 3 which are connected in series for increasing the voltage upto 11.1 V.As it is rechargeable and last upto 3 hours which meet our requirement.

Nominal voltage:12 V

Capacity:2200mAh

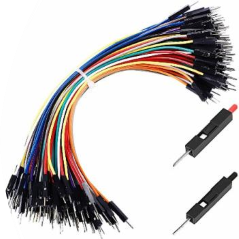
Material: ABS case

**Fig.3.7**

1. **JUMPER WIRE:**

Jumper wire is used for connecting all the components with each other.

The types of jumper wire used is male-to-male, male-to-female and female-to-female. As it can carry current from 4-20 mA and voltage upto 12V which match our basic requirement for connections. The cable length is 20cm-8 inch with rated pressure of 25KPA.

****

**Fig3.8**

1. **By considering all the components dimension, we have design the LFR chassis which has optimal size and strong enough to bear the load of each components:**

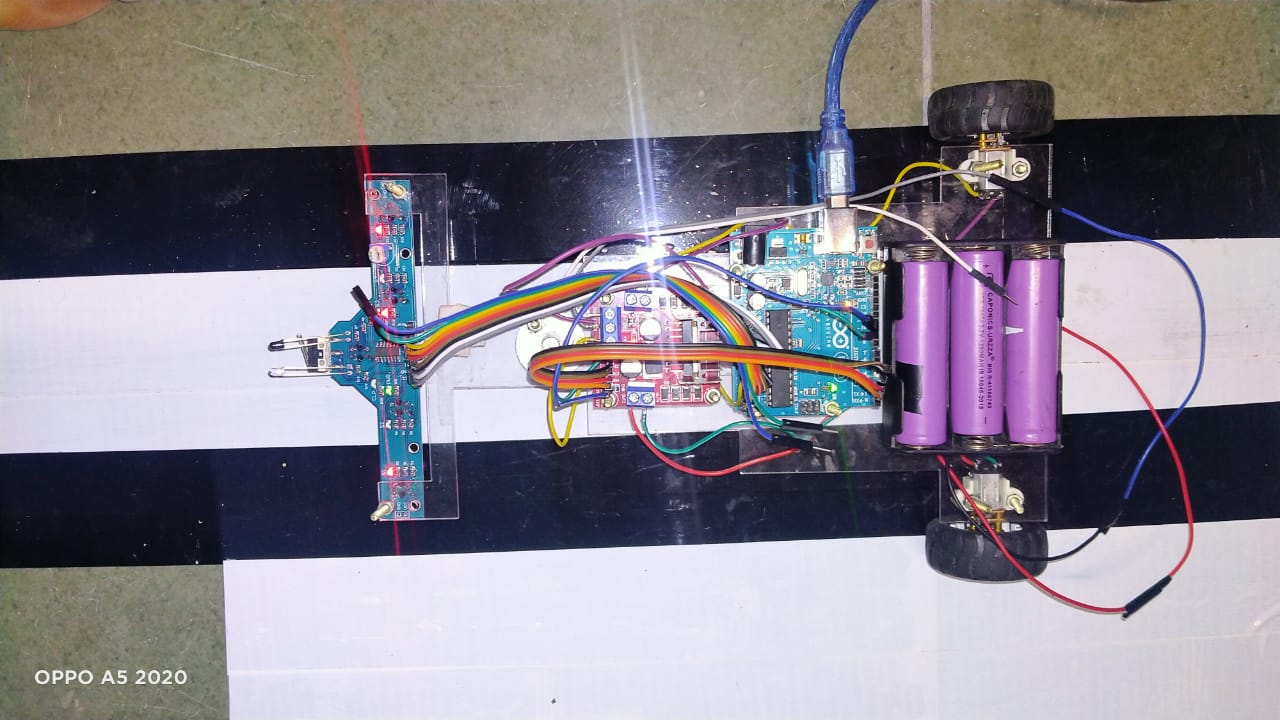
A diagram of a diagram

Description automatically generated**CHASSIS :** Chassis is the main base for line follower robot where all the components are to be mounted on.The design of chassis is made while considering dimensions of all the components which are to be used.The dimensions of chassis in terms of length,breath and width are 24×14×0.5 cm.The material used for the chassis is acrylic sheet which is a transparent sheet and can bear the weight of mounted components on it.We made the design in coral software as shown in Fig 3.1 with respective considered dimensions.The design then cut from acrylic sheet through laser cutting as shown in Fig 3.1.Then the assembly is made on designed acrylic sheet as shown in Fig 3.2. All the components are fixed on chassis and connected to the controller with the help of jumper wire.

A machine with a flame

Description automatically generated**Fig.3.9**

**Fig.3.10**

1. **All the components are fixed on chassis and connected to the controller with the help of jumper wire.**
2. **Black & white Path making:**

Path making of LFR is done in a such a way that its middle sensors i.e s2,s3 & s4 of 5 channel array lies always on path weather the path is Black or White.

Therefor the width of this path is equal to the spacing between three middle sensors which is 6.5cm.

1. **Implementation of logical programming code:**

Write the code for your robot's microcontroller, implementing the line following algorithm using the calibrated sensor values. The code should read the sensor values, calculate the error (difference between desired position and actual position on the line), and adjust the motor speeds to keep the robot on the line.

1. **Sensor Calibration:**

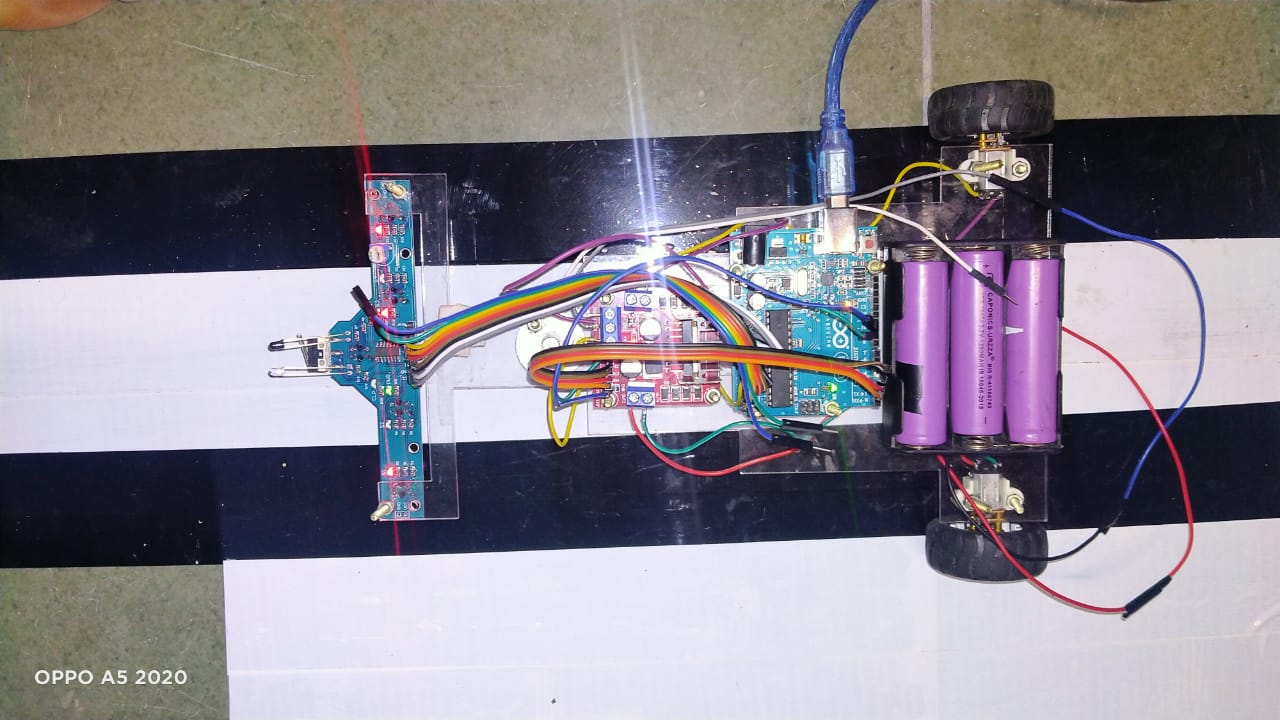
Calibrate the IR sensors to determine the threshold values for detecting the black and white surfaces. During calibration, read the sensor outputs while placing the robot on black and white lines. Adjust the threshold values accordingly, so the robot can reliably distinguish between the two.

1. **Test and Debug:**

Upload the code to the microcontroller and perform initial tests. Make sure the robot responds correctly to changes in line color and direction. Debug and fine-tune the algorithm as needed to improve its performance.

1. **Handling Motions and Turns:**

To handle intersections or turns, modify your algorithm to detect such scenarios using the sensor inputs. Implement a decision-making logic to guide the robot through the intersection or turn and back onto the line.

1. **Final Assembly**

**CHAPTER 4**

**IMPLEMENTATION OF LINE FOLLOWER ROBOT**

**OVERALL WORK:** We have done assembly & Path making of Line follower robot. The Implementation of LFR programming logic is done according to its motion on path. These motions are forward motion, Left-Turn, Right-Turn, U-Turn (applicable to both the path).

1. **FORWARD MOTION:** In this motion the line follower should move forward direction on given path. When S2, S3, S4 these middle sensors are on black path. S1 & S2 all have a value of 1 which means these sensors are on the white surface then the LFR moves forward.

**S5**

**S1**

**S4**

**S2**

**S3**

**Forward Motion**

 //forward moment on black line

  if((s1 == 1) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 1))

  {

    analogWrite(e1, 255);

    analogWrite(e2, 255);

    digitalWrite(m1, HIGH);

    digitalWrite(m2, LOW);

    digitalWrite(m3, HIGH);

    digitalWrite(m4, LOW);

  }

1. **RIGHT-TURN:** When S1 is on a White surface i.e. these sensors have a value of 1, and S2,S3,S4,&S5 are on a Black surface means these sensors have a value of 0. The left motor rotates at high speed while the right motor rotates at low speed or stops so that LFR takes the Right turn.

**Right Turn**

**S5**

**S1**

**S4**

**S2**

**S3**

//Take Right turn on Black path

  if((s1 == 1) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 0))

  {

    analogWrite(e1, 100);

    analogWrite(e2, 255);

    digitalWrite(m1, LOW);

    digitalWrite(m2, HIGH);

    digitalWrite(m3, HIGH);

    digitalWrite(m4, LOW);

  }

1. **LEFT-TURN:** When S1, S2, S3 & S4 are on a black surface i.e. have a value of 0. and S5 is on a white surface that is these sensors have a 1 value. The right motor rotates at high speed and the left motor rotates at low speed or stops. So that LFR takes a Left turn.

**Left Turn**

**S5**

**S1**

**S4**

**S2**

**S3**

//Take Left turn on black path

  if((s1 == 0) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 1))

  {

    analogWrite(e1, 255);

    analogWrite(e2, 100);

    digitalWrite(m1, HIGH);

    digitalWrite(m2, LOW);

    digitalWrite(m3, LOW);

    digitalWrite(m4, HIGH);

  }

1. **U-TURN:** When all sensors are on a White or Black surface i.e. all have a value of 1 or 0 then the LFR moves 180 degrees i.e. take a U-turn.

**S5**

**S1**

**S4**

**S2**

**S3**

**U-TURN.**

  //if all sensors are on White line take u-turn

 if((s1 == 1) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 1))

  {

    while((s1 == 0) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 0))

    {

    digitalWrite(m1, HIGH);

    digitalWrite(m2, LOW);

    digitalWrite(m3, LOW);

    digitalWrite(m4, HIGH);

   }

  }

1. **PATH SHIFT FROM BLACK TO WHITE:** When S1, S3, S5 these sensors are on the white surface then it means that these sensors read 1 digital value & if sensors S2 & S4 are on a Black surface read 0 value. It indicates that LFR is in between the Black and White line & the robot is shifting the path from the Black line to the White line & moves in the forward direction on the White line.

**S5**

**S1**

**S4**

**S2**

**S3**

**Path change from Black to white.**

 //path shift from Black to white

  if((s1 == 1) && (s2 == 0) && (s3 == 1) && (s4 == 0) && (s5 == 1))

  {

    analogWrite(e1, 255);

    analogWrite(e2, 255);

    digitalWrite(m1, HIGH);

    digitalWrite(m2, LOW);

    digitalWrite(m3, HIGH);

    digitalWrite(m4, LOW);

  }

1. **PROGRAMMING CODE FOR LFR:** We have written the logical programming code for Line follower robot into the Arduino IDE. The code is capable to execute all the motion conditions like forward Motion, Left & Right Turn, U-Turn on both (Black & White) Path.

#define m1 4 //Right Motor MA1

#define m2 5 //Right Motor MA2

#define m3 2 //Left Motor MB1

#define m4 3 //Left Motor MB2

#define e1 9 //Right Motor Enable Pin EA

#define e2 10 //Left Motor Enable Pin EB

//\*\*\*\*\*\*\*\*\*\*5 Channel IR Sensor Connection\*\*\*\*\*\*\*\*\*\*//

#define ir1 A0

#define ir2 A1

#define ir3 A2

#define ir4 A3

#define ir5 A4

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void setup() {

pinMode(m1, OUTPUT);

pinMode(m2, OUTPUT);

pinMode(m3, OUTPUT);

pinMode(m4, OUTPUT);

pinMode(e1, OUTPUT);

pinMode(e2, OUTPUT);

pinMode(ir1, INPUT);

pinMode(ir2, INPUT);

pinMode(ir3, INPUT);

pinMode(ir4, INPUT);

pinMode(ir5, INPUT);

}

void loop() {

//Reading Sensor Values

int s1 = digitalRead(ir1); //Left Most Sensor

int s2 = digitalRead(ir2); //Left Sensor

int s3 = digitalRead(ir3); //Middle Sensor

int s4 = digitalRead(ir4); //Right Sensor

int s5 = digitalRead(ir5); //Right Most Sensor

//forward motion of LFR on white line

if((s1 == 0) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 0))

{

analogWrite(e1, 255);

analogWrite(e2, 255);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//Move little-bit right for adjusting the motion of LFR white path

if((s1 == 0) && (s2 == 0) && (s3 == 1) && (s4 == 1) && (s5 == 0))

{

analogWrite(e1, 255);

analogWrite(e2, 255);

digitalWrite(m1, LOW);

digitalWrite(m2, LOW);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//Move little-bit left for adjusting the motion of LFR white path

if((s1 == 0) && (s2 == 1) && (s3 == 1) && (s4 == 0) && (s5 == 0))

{

analogWrite(e1, 255);

analogWrite(e2, 255);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, LOW);

digitalWrite(m4, LOW);

}

//Take right turn on White path

if((s1 == 0) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 1))

{

analogWrite(e1, 100);

analogWrite(e2, 255);

digitalWrite(m1, LOW);

digitalWrite(m2, HIGH);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//Take right turn on White path

if((s1 == 1) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 0))

{

analogWrite(e1, 255);

analogWrite(e2, 100);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, LOW);

digitalWrite(m4, HIGH);

}

//forward moment on black line

if((s1 == 1) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 1))

{

analogWrite(e1, 255);

analogWrite(e2, 255);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//Move little-bit right for adjusting the motion of LFR white Black path

if((s1 == 1) && (s2 == 1) && (s3 == 0) && (s4 == 0) && (s5 == 1))

{

analogWrite(e1, 255);

analogWrite(e2, 255);

digitalWrite(m1, LOW);

digitalWrite(m2, LOW);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//Move little-bit Left for adjusting the motion of LFR on Black path

if((s1 == 1) && (s2 == 0) && (s3 == 0) && (s4 == 1) && (s5 == 1))

{

analogWrite(e1, 255);

analogWrite(e2, 255);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, LOW);

digitalWrite(m4, LOW);

}

//Take Right turn on Black path

if((s1 == 1) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 0))

{

analogWrite(e1, 100);

analogWrite(e2, 255);

digitalWrite(m1, LOW);

digitalWrite(m2, HIGH);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//Take Left turn on black path

if((s1 == 0) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 1))

{

analogWrite(e1, 255);

analogWrite(e2, 100);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, LOW);

digitalWrite(m4, HIGH);

}

//path shift from White to Black

if((s1 == 0) && (s2 == 1) && (s3 == 0) && (s4 == 1) && (s5 == 0))

{

analogWrite(e1, 255);

analogWrite(e2, 255);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//path shift from Black to white

if((s1 == 1) && (s2 == 0) && (s3 == 1) && (s4 == 0) && (s5 == 1))

{

analogWrite(e1, 255);

analogWrite(e2, 255);

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

//if all sensors are on a black line take u-turn

if((s1 == 0) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 0))

{

while((s1 == 1) && (s2 == 0) && (s3 == 0) && (s4 == 0) && (s5 == 1))

{

digitalWrite(m1, LOW);

digitalWrite(m2, HIGH);

digitalWrite(m3, HIGH);

digitalWrite(m4, LOW);

}

}

//if all sensors are on White line take u-turn

if((s1 == 1) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 1))

{

while((s1 == 0) && (s2 == 1) && (s3 == 1) && (s4 == 1) && (s5 == 0))

{

digitalWrite(m1, HIGH);

digitalWrite(m2, LOW);

digitalWrite(m3, LOW);

digitalWrite(m4, HIGH);

}

}

}

**CHAPTER 5**

**RESULTS AND DISCUSSIONS**

The result of a line follower robot with 5 IR sensors that can follow both black and white lines and take a U-turn would be a robot capable of autonomously navigating along a marked path with turns.

Here's a general outline of how the robot would behave:

Line Following: The 5 IR sensors will be positioned on the front of the robot in a line, facing the ground. As the robot moves forward, the sensors will continuously detect the color of the surface beneath them. When the robot encounters a black line on a white surface or a white line on a black surface, the sensors will detect the contrast, and the robot will be able to follow the line by adjusting its course accordingly. The number of sensors allows for better accuracy and robustness in tracking the line.

Turning: To take a U-turn or any other turn, the robot would rely on its sensors to detect specific patterns indicating the need to turn. For example, if the robot encounters a "T" junction or a sharp curve in the line, it can activate the appropriate motors to initiate the turn. This turning behavior can be programmed into the robot's control algorithm.

Decision-making: The robot will have an embedded control algorithm that interprets the sensor readings and makes decisions based on predefined rules. For instance, when the sensors detect a significant deviation from the line, the robot will trigger a turn in the opposite direction to realign with the line.

The result of such a robot would be a line follower that can effectively navigate a track with twists, turns, and even U-turns while following both black and white lines. Keep in mind that building a successful line follower robot requires careful tuning of the control algorithm and sensors to ensure reliable performance under various conditions. Additionally, the construction and design of the robot should be such that it can smoothly execute turns without tipping over or losing balance.

**CHAPTER 6**

**CONCLUSIONS AND FUTURE SCOPE**

**6.1 CONCLUTION:**

Line Following Capability: The robot demonstrates the ability to follow lines accurately using the 5 infrared sensors. These sensors allow it to detect the contrast between the black line and the white surface, enabling it to maintain a precise path along the desired route.

Adaptive Behavior: The robot's infrared sensors provide it with the flexibility to adapt to changes in line width and line patterns. It can effectively track lines of different thicknesses and handle curves with ease.

Dual Line Recognition: The infrared sensors allow the robot to detect both black lines on a white surface and white lines on a black surface. This enables the robot to operate on various types of tracks or surfaces.

U-Turn Capability: The robot is equipped with the necessary algorithms and control mechanisms to perform U-turns accurately when encountering junctions or dead-ends. It can reverse its direction and resume line following after making the U-turn.

Real-time Responsiveness: The robot exhibits real-time responsiveness to changes in line direction or U-turn scenarios. Its ability to process sensor data quickly allows it to make prompt decisions and adjust its movements accordingly.

Robustness and Reliability: The robot demonstrates consistent performance, even in challenging environments. Its design and control algorithms ensure that it can handle different lighting conditions, track imperfections, and minor obstacles.

Potential Applications: The line follower robot's capabilities make it suitable for a wide range of applications, including automated warehouse transportation, assembly line operations, and educational purposes to teach robotics and automation principles.

Overall, the line follower robot with 5 infrared sensors proves to be an efficient and intelligent system capable of accurately tracking lines, following various line colors, and performing U-turns with precision and reliability. Its versatility and adaptability make it a valuable tool in the field of robotics and automation.

**6.2 FUTURE SCOPE:**

* Optical detection with the help of ultrasonic sensors and servo motors.
* Crop Health Monitoring using advanced navigation technology.
* Climbing robot using nano materials
* Patrolling by integrating image camera sensors and night vision cameras for nighttime use.
* Use of IoT for controlling the navigating designed path.

**REFERENCES**

https://www.instructables.com/Robot-Line-Follower

https://forum.arduino.cc/t/how-do-i-make-a-line-following-robot-turn-around/266502

https://github.com/himanshus2847/Line-Follower-Robot-using-5-Channel-IR/blob/master/Line\_follower\_5\_channel/Line\_follower\_5\_channel.ino