**MICROCONTROLLER PROJECT REPORT**

***In partial fulfilment of the requirements for the award of the degree of***

**Bachelor of Technology**

**in**

**Electronics and Communication Engineering**

V semester

*Submitted by*

**Student Name: Ankush Verma**

**Register Number: 200907058**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

MANIPAL INSTITUTE OF TECHNOLOGY

(A Constituent institute of MAHE, Manipal)

MANIPAL – 576104, KARNATAKA, INDIA

**PROJECT: LINE-FOLLOWER-BOT-UTILIZING-ARDUINO-TECHNOLOGY**

**1. INTRODUCTION:**

The Line Follower Robot is a straightforward and popular robot that I designed to track and follow a line, whether it is a black line or a white line. These robots are considered ideal for beginners venturing into the world of robotics due to their simplicity and ease of construction. Essentially, there are two types of line follower robots: the black line follower, which I programmed to trace and follow black lines, and the white line follower, which I designed to track and follow white lines. The fundamental concept behind a line follower robot involves its ability to detect the line and pursue its path accordingly.

While the basic idea may seem simple, with further advancements and refinement, these robots have found practical applications in various industries. For instance, they are employed as efficient management robots in factory floors or warehouses. In my endeavour, I have successfully developed an Arduino UNO Line Follower Robot.

*Concepts of Line Follower:*

The working principle of a line follower robot revolves around the behavior of light when it encounters black and white surfaces. Specifically, I leverage the properties of light reflection and absorption to create the line following mechanism.

*Concept of White Line Follower Robot:*

In the case of a white line follower robot, we utilize IR Transmitters and IR receivers, also known as photodiodes. These components play a crucial role in sending and receiving light signals. The IR transmitter emits infrared light, which, upon falling on a white surface, is mostly reflected and captured by the photodiodes. As a result, voltage changes occur in the photodiodes.

*Concept of Black Line Follower Robot:*

Contrarily, for a black line follower robot, the behaviour of light is different. When IR light falls on a black surface, the surface absorbs the light, leading to no reflection of rays. Consequently, the photodiodes do not receive any light or rays in this scenario. Thus, the photodiodes produce no significant voltage changes.

In this Arduino-based line follower robot, these principles are effectively applied to detect white and black surfaces. When the sensors sense a white surface, the Arduino receives a logic high (1) as input. Conversely, when the sensors encounter a black line, the Arduino receives a logic low (0) as input.

A diagram of a light source

Description automatically generated

A diagram of a light and a light

Description automatically generated

**2.** **CIRCUIT EXPLANATION:**

The entire Arduino line follower robot can be categorized into three main sections: the sensor section, the control section, and the driver section.

Sensor Section:

In this section, IR diodes, a potentiometer, a Comparator (Op-Amp), and LEDs are integrated. The potentiometer serves the purpose of setting the reference voltage at one terminal of the comparator, while the IR sensors are responsible for detecting the line and producing a voltage change at the second terminal of the comparator. Subsequently, the comparator compares these voltages and generates a digital signal at its output. In the line follower circuit, two comparators are used, each for a different sensor. The LM 358 is employed as a comparator, equipped with two low-noise Op-amps.

Control Section:

The Arduino Pro Mini is utilized to oversee the entire process of the line follower robot. The outputs of the comparators are connected to digital pin numbers 2 and 3 of the Arduino. The Arduino reads these signals and sends corresponding commands to the driver circuit, facilitating the operation of the line follower.

Driver Section:

Within the driver section, a motor driver and two DC motors are incorporated. The motor driver plays a crucial role in driving the motors since the Arduino alone cannot provide sufficient voltage and current for the motor's operation. Thus, the motor driver circuit is implemented to obtain the necessary voltage and current for the motors. Arduino sends specific commands to this motor driver, enabling it to drive the motors efficiently.

**3.** **WORKING OF LINE FOLLOWER ROBOT USING ARDUINO**

Building a Line follower robot using Arduino is an interesting venture. The line follower robot senses a black line by using a sensor, and then it sends the signal to Arduino. Then Arduino drives the motor according to the sensors' output.

A diagram of a motor driver

Description automatically generated

Here in this project, I am using two IR sensor modules, namely the left sensor and the right sensor. When both the left and right sensors sense white, then the robot moves forward.

Diagram of a robot with text and words

Description automatically generated

Figure 3.1 Both the left and right sensors sense white

If the left sensor comes upon a black line, then the robot turns left.

A diagram of a robot

Description automatically generated

Figure 3.2 Left sensor comes upon a black line.

If the right sensor senses a black line, then the robot turns right until both sensors come across the white surface. When the white surface is detected, the robot starts moving forward again.

A diagram of a robot

Description automatically generated

Figure 3.3 Right sensor comes upon a black line.

If both sensors come upon the black line, the robot stops.

A diagram of a robot

Description automatically generated

Figure 3.4 Both sensors comes upon a black line.

**4.** **Circuit Diagram**

The circuit diagram for the Arduino line follower robot is depicted in the below image. As observed, the outputs of the comparators are directly linked to Arduino digital pin number 2 and 3. Additionally, the input pins 2, 7, 10, and 15 of the motor drivers are connected to Arduino's digital pin number 4, 5, 6, and 7, respectively. Furthermore, one motor is connected to the output pins of the motor drivers 3 and 6, while another motor is connected to pins 11 and 14.

A circuit board with many wires

Description automatically generated

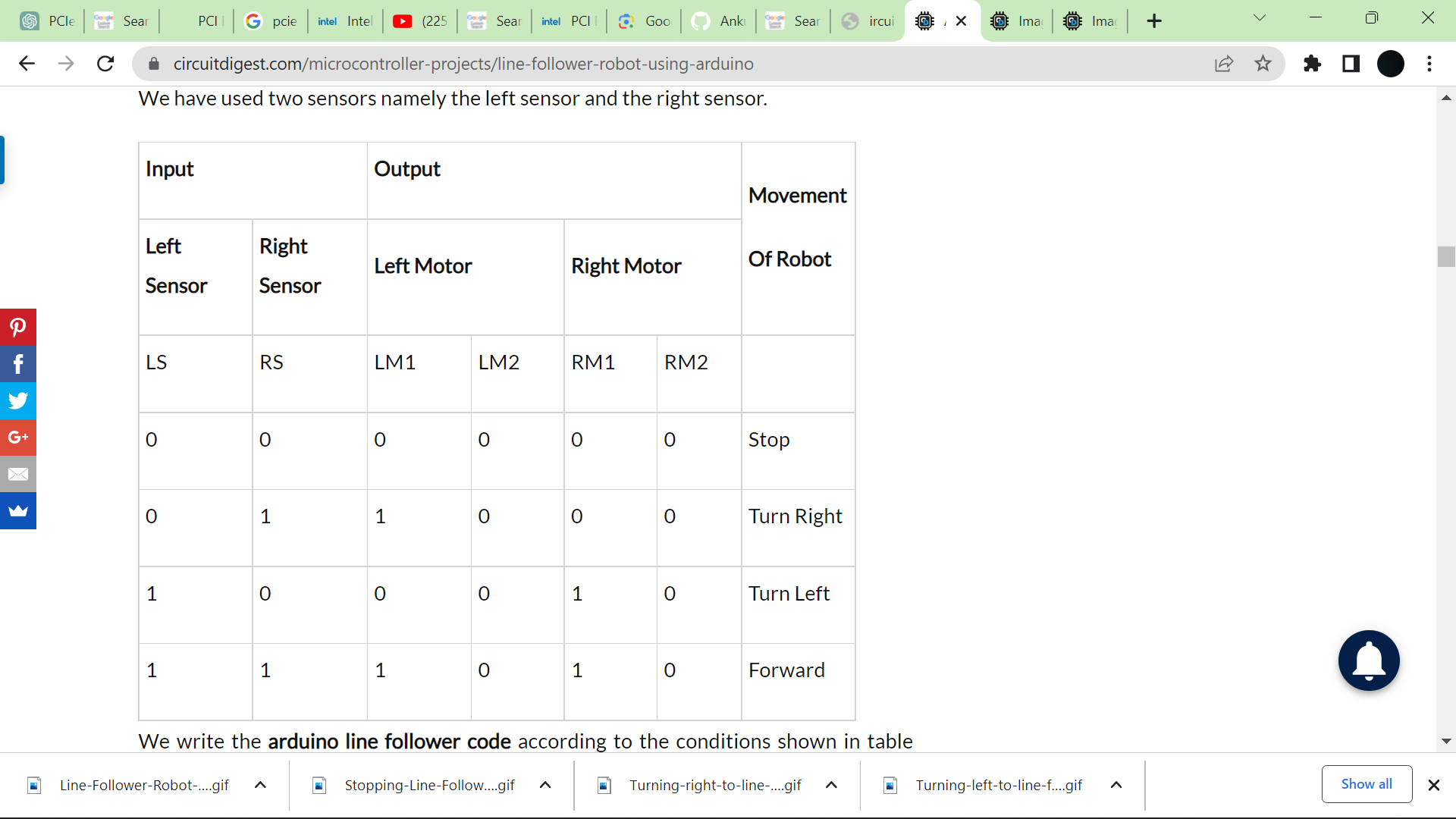


**4.** PROGRAM EXPLANATION:

In the program, I first define the input and output pins. Then, within the loop, I continuously check the inputs and send corresponding outputs to the output pins for driving the motors. This is achieved through the use of "if" statements to determine the appropriate movement of the robot based on the sensor inputs.

*Code Logic for Line Follower Robot:*

Below is the Arduino code logic for the line follower robot. There are four conditions in this line following robot that we read by using Arduino. We have used two sensors namely the left sensor and the right sensor.



**5. ARDUINO LINE FOLLOWER CODE**

/\*-------defining Inputs------\*/

#define LS 2 // left sensor

#define RS 3 // right sensor

/\*-------defining Outputs------\*/

#define LM1 4 // left motor

#define LM2 5 // left motor

#define RM1 6 // right motor

#define RM2 7 // right motor

void setup()

{

pinMode(LS, INPUT);

pinMode(RS, INPUT);

pinMode(LM1, OUTPUT);

pinMode(LM2, OUTPUT);

pinMode(RM1, OUTPUT);

pinMode(RM2, OUTPUT);

}

void loop(

{

if(digitalRead(LS) && digitalRead(RS)) // Move Forward

{

digitalWrite(LM1, HIGH);

digitalWrite(LM2, LOW);

digitalWrite(RM1, HIGH);

digitalWrite(RM2, LOW);

}

if(!(digitalRead(LS)) && digitalRead(RS)) // Turn right

{

digitalWrite(LM1, LOW);

digitalWrite(LM2, LOW);

digitalWrite(RM1, HIGH);

digitalWrite(RM2, LOW);

}

if(digitalRead(LS) && !(digitalRead(RS))) // turn left

{

digitalWrite(LM1, HIGH);

digitalWrite(LM2, LOW);

digitalWrite(RM1, LOW);

digitalWrite(RM2, LOW);

}

if(!(digitalRead(LS)) && !(digitalRead(RS))) // stop

{

digitalWrite(LM1, LOW);

digitalWrite(LM2, LOW);

digitalWrite(RM1, LOW);

digitalWrite(RM2, LOW);

}

}