PROJECT REPORT LINE FOLLOWING ROBOT

Technical Report · August 2016

DOI: 10.13140/RG.2.2.21215.79522

3 authors:

Sujeet Kumar Jha
Tribhuvan University
8 PUBLICATIONS **17** CITATIONS
SEE PROFILE

Manish Karn
Indian Institute of Technology Ropar
2 PUBLICATIONS **1** CITATION
SEE PROFILE

Ahmed Raja Khan
Kathmandu University
2 PUBLICATIONS **13** CITATIONS
SEE PROFILE

KATHMANDU UNIVERSITY

SCHOOL OF ENGINEERING
Department of Electrical & Electronics Engineering

PROJECT REPORT

LINE FOLLOWING ROBOT

A **third year project report** submitted in partial fulfilment
of the requirements for the degree of
Bachelor of Engineering

by:
Sujeet Kumar Jha (32009)
Saurab Dulal (32007)
Manish Karn (32035)
Ahmed Raja Khan (32432)

August 2016

CERTIFICATION

THIRD YEAR PROJECT REPORT
on

LINE FOLLOWING ROBOT

by:

Sujeet Kumar Jha (32009)

Saurab Dulal (32007)

Manish Karn (32035)

Ahmed Raja Khan (32432)

Approved by:

1. Project Supervisor

_____	_____	_____
(Signature)	(Name)	(Date)

2. Head/In-Charge of the Department

_____	_____	_____
(Signature)	(Name)	(Date)

Table of Contents

ABSTRACT

Line Following is one of the most important aspects of robotics. A Line Following Robot is an autonomous robot which is able to follow either a black line that is drawn on the surface consisting of a contrasting color. It is designed to move automatically and follow the line. The robot uses arrays of optical sensors to identify the line, thus assisting the robot to stay on the track. The array of four sensor makes its movement precise and flexible. The robot is driven by DC gear motors to control the movement of the wheels. The Arduino Uno interface is used to perform and implement algorithms to control the speed of the motors, steering the robot to travel along the line smoothly.This project aims to implement the algorithm and control the movement of the robot by proper tuning of the control parameters and thus achieve better performance.In addition the LCD interface is added in order to display the distance travelled by the robot. It can be used industrial automated equipment carriers, small household applications, tour guides in museums and other similar applications, etc.

ACKNOWLEDGEMENT

We wish to express our profound and deep sense of gratitude to Mr. Anand Gacchadar, Project Co-ordinator,Department of Electrical and Electronics Engineering for sparing his valuable time to extend help in every step of our project work.
We whole heartedly express our thanks to, Mr. Bikalpa Upadhayaya, Project Supervisor, Department of Electrical and Electronics Engineering, for sparing time to go through every tiny detail and give his valuable suggestions to make this project and report a success.
We are mainly indebted to the authors of many references and articles which were used as the reference.
Last but not the least we would like to thank our friends and family for their help in every way for the success of this project report.

ABBREVIATIONS AND SYMBOLS

A. Abbreviations

Abbreviation	Full Form	First Used in Page
CPU	Central Processing Unit	7
EEPROM	Electrically Erasable Programmable Read Only Memory	7
IC	Integrated Circuit	11
LCD	Liquid Crystal Display	1
LDR	Light Dependent Resistor	4
LED	Light Emitting Diode	4
MHz	Mega Hertz	7
PWM	Pulse Width Modulation	8
Rx	Receiver	5
SRAM	Static Read Only Memory	7
Tx	Transmitter	5

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

1. Introduction

Line follower is a machine that can follow a path. The path can be visible like a black line on a white surface.Sensing a line and maneuvering the robot to stay on course, while constantly correcting wrong moves using feedback from the sensor forms a simple yet effective system. It can be used in automobile, industrial automations, guidance, etc [1].

1.1. Background

As technology becomes increasingly important in today's world, it is invaluable to not only learn how to use technology, but also to understand how to create it. Since being the engineer one should have sound knowledge of the other discipline.Most of the projects have limted scope to only specific discipline.This would limit ones innovation and creativity. This project inspires to make connections across several disciplines rather than learning topics in isolation as it combines mechanical, electronic, electrical and programming skills.

- It gives visual grasp of math and science.
- It builds logical thinking.
- It brings out innovation and creativity.
- It enhances problem solving skills.

The robot designed in a such way that it not only track the path and follow it but also visualize the distance travelled through the LCD displays.

1.2. Motivation

How ants always travel in a line, following ani nvisible route in search of food, or back home. How on roads the lanes is followed to avoid accidents and traffic jams. Ever thought about a robot which follows line? A perfect or near perfect mimic of nature? After all the purpose of robotics is to recreate in terms of machines what one see around to solve a problem or fulfill a requirement.

The area will be benefitted from the project

- Industrial automated equipment carriers
- Entertainment and small household applications.
- Tour guides in museums and other similar applications.
- Second wave reconnaissance operations.

1.3. Problem Description

In the industry carriers are required to carry products from one manufacturing plant to another which are usually in different buildings or separate blocks. Conventionally, carts or trucks were used with human drivers. Unreliability and inefficiency in this part of the assembly line formed the weakest link. The project is to automate this sector, using carts to follow a line instead of laying railway tracks which are both costly and an inconvenience.

1.4. Objectives

The objectives of the project are:

- The robot must be capable of following a line.
- It should be capable of taking various degrees of turns.
- 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.
- It also shows the distance travelled by it through LCD display.

1.5. Methodology

- After the detail literature survey through the books, periodical, journal, magazine, websites. The idea of the project is well defined.
- The logic is derived for the intelligence of the robot. It is programmed and burn it to the Arduino by using the software Arduino® 1.65.
- The accuracy and viability of the program and electronic components is tested in the simulation software Proteus®.
- After the sucessful simulation result it is implemented in the hardware.
- After the finishing the programming, electrical and electronics part, the stable,reliable and flexible mechanical design and fabrication is completed.
- Finally system is tested and encountered error is omitted.

1.6. Limitations

The system has restricted to the following limitation.

- Choice of line is made in the hardware abstraction and cannot be changed by software.
- Calibration is difficult, and it is not easy to set a perfect value.

- Few curves are not made efficiently, and must be avoided.
- The turning radius should be of minimum 100m to take smooth U-turning of robot.
- The width of the path must be of 45mm so that it can cover minimum 3 sensors.
- The path should be plane and obstacle free.
- The steering mechanism is not easily implemented in huge vehicles and impossible for non-electric vehicles.

1.7. Organisation of Report

This report is a documentary delivering the ideas generated, concepts applied, activities done It contains four chapters. The following is a description of information in this thesis.
Chapter 1 provides a general overview of the project and the use and importance of autonomous robots in the world. The objectives, scope of project, problem statement are also described in this chapter.
Chapter 2 describes the hardware development unit in line following robot. This chapter describes about sensor arrays, Arduino, motor driving system, it also describes the project methodology and explains hardware development for the design of the robot.
Chapter 3 contains the process explanation with working algorithm, flowchart and sketch of the Arduino.
Chapter 5 contains all the results obtained from the software experiments that include the algorithm implemented in a program.
Finally, chapter 5 will summarize the final year project. The conclusion, suggestions or recommendations for improvements that can be implemented in future are discussed within this chapter.

1.8. Summary

Line Follower is one of the most important aspects of robotics. A Line Following Robot is an autonomous robot which is able to follow either a black or white line that is drawn on the surface consisting of a contrasting color. It is designed to move automatically and follow the plotted line. It enhances interdisciplinary approach to mechanical,electronic, electrical and programming skills. The application of the project is range from the individual domestic appliance to automation and control aspect of large industry. Human are intelligent natural machine but it has serious limation of efficiency and realiblity. Robots are made to replace dependency of human force partially. The project is somehow designed to perform the similar task.

CHAPTER 2

2. Technology and Literature Survey

The line follower is a self-operating robot that detects and follows a line that is drawn on the floor .The line follower robot using Arduino is a self-operating system that detects and follows track drawn on the floor. The track consists of a black path drawn on white surface.

2.1. Basic Operation

The basic operations of line follower are as follows:

- Capture line position with optical sensors mounted at front end of the robot. For this a combination of IR-LED and Photodiode called an optical sensor has been used. This make sensing process of high resolution and high robustness.
- Steer robot requires steering mechanism for tracking. Two motors governing wheel motion are used for achieving this task.
- This system has LCD display panel to show the distance that it covers.
- On the detecting no black surface robot move in a circular motion until line is found.

2.1.1. Block Diagram

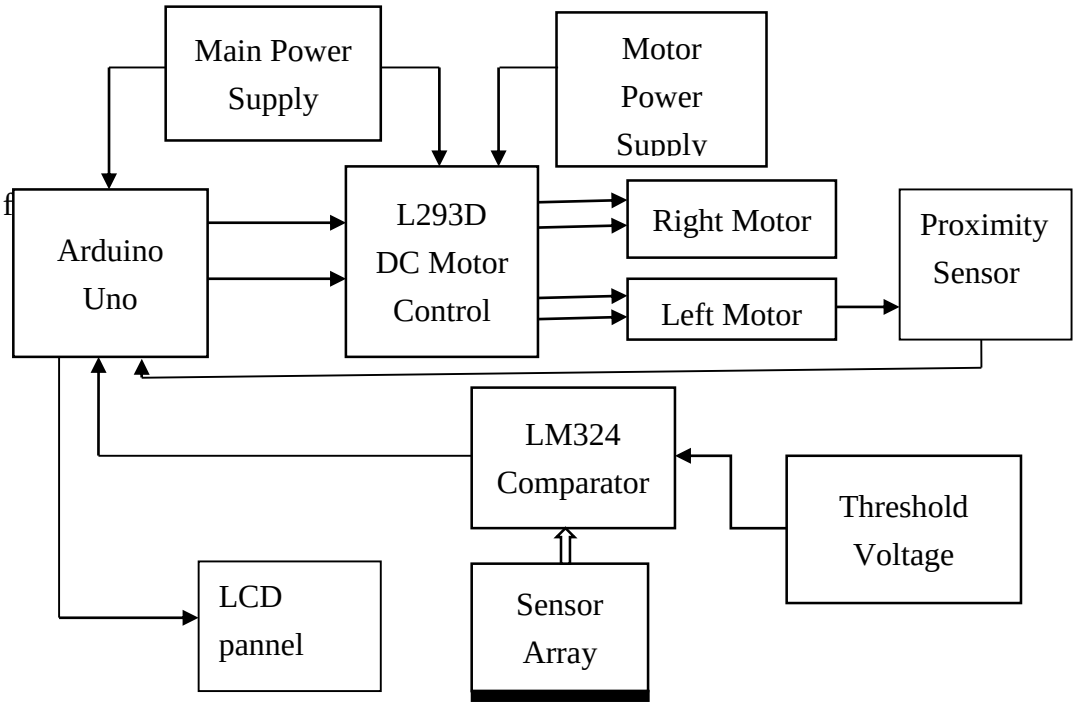


Figure 2-1: Block Digram of the Line following Robot.

2.2. Hardware Required

The hardware required is divided in the following category:

2.2.1 Input System

The detail of each components is discussed below.

2.2.1.1. Optical Sensors

The robot uses photodiode sensors to sense the line; an array of four IR- LEDs (TX) and Photodiode sensors (Rx), facing the ground used in this setup. An analog signal is obtained in output, depends on the amount of light reflected back, which is provided to the comparator to produce 0s and 1s which are then fed to the Arduino.

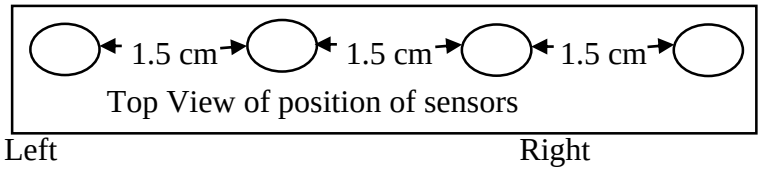


Figure 2-2: Arrangements of the Sensor

The sensors on the left named as L1, L2 and R1, R2 on the right side. Assumption should be considered that when a sensor is on the line it reads 0 and when it is off the line it reads 1. The Arduino correspondence to the algorithm given below decides the next movement, trying to position the robot such that all sensors read 0.

With sensors, robots can react and respond to changes in their environment in ways that appear intelligent or life-like.

2.2.1.1.1. Arrangement of Sensors

An array of sensors arranged in a straight row pattern is bolted under the front of the robot. It locates the position of line below the robot. Distance between two adjacent sensors is about 1.5 cm.

We can use any number of sensors. If we have low number, then our robot movement is not smooth and it may face problems during sharp turns. If higher number of sensors were, used robot movement will become smooth and reliable for sharp turns; it requires complex programming for micro-controller and requires more hardware, which is its disadvantage. Thus, optimum number of sensors required.

2.2.2. Comparator

Comparator is a device, which compares two input voltages and gives output high or low. In circuit diagram it is normally represented by a triangle having-Inverting (negative) Input, Non-Inverting (positive) Input (+), Vcc, Ground, Output.

Properties of comparator:

If V+ > V- then Vo=Vcc (Digital High Output is 1)

If V+ < V- then Vo=0 (Digital Low Output is 0)

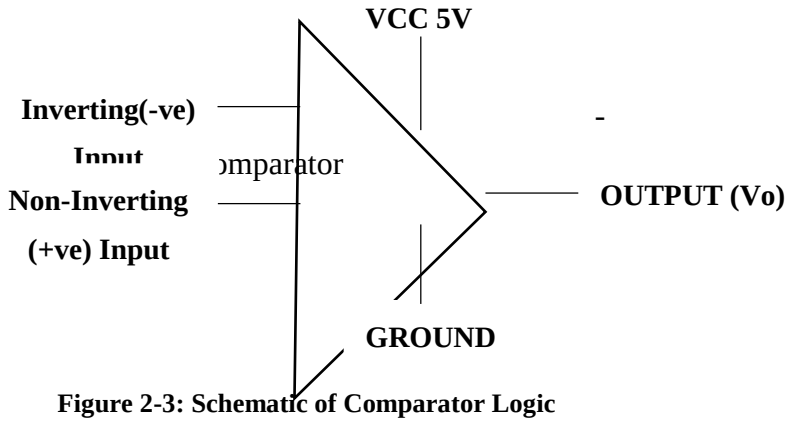


Figure 2-3: Schematic of Comparator Logic

2.2.2.2.1. Optical sensor

As shown above that two inputs are required for comparator. One input is from photo-receiver (like Photodiode),other is generated by resistor.The second voltage is reference voltage for that sensor.

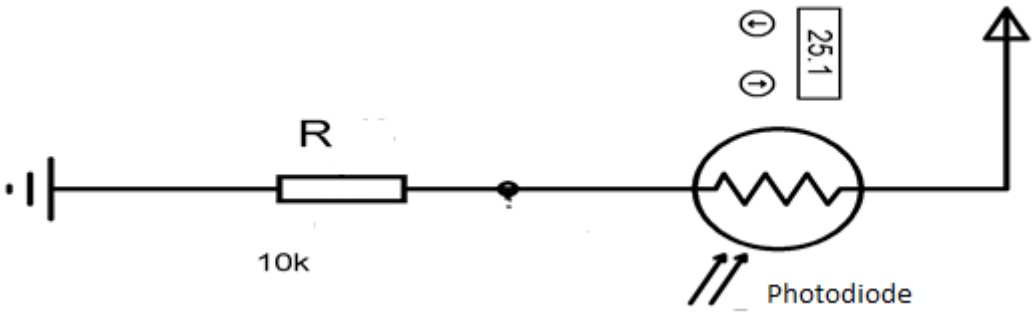


Figure 2-4: Optical Sensor schematic

2.2.2 Processing unit

Processing system acts as the brain of robot, generating desired output for corresponding inputs, in which microcontrollers are used. There are several companies nowadays that manufacture microcontrollers, for example ATMEL corporation, Microchip, Intel, Motorola etc. We will be using ATmega32L microcontroller in our robot. It is known as AVR[1].

2.2.2.1. Arduino

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control objects in the physical world. The heart of Arduino is the microcontroller. For Arduino Uno ATmega328 is used.It has specification of 8 bit CPU, 16 MHZ clock speed, 2 KB SRAM 32 KB flash Memoary, 1 KB EEPROM [2].

Features :-

- 14 digital input output pins (3,5,6,9,10 and 11 pins are able to generate PWM).
- 6 analog input pins
- Voltage input from the 7 – 12 V

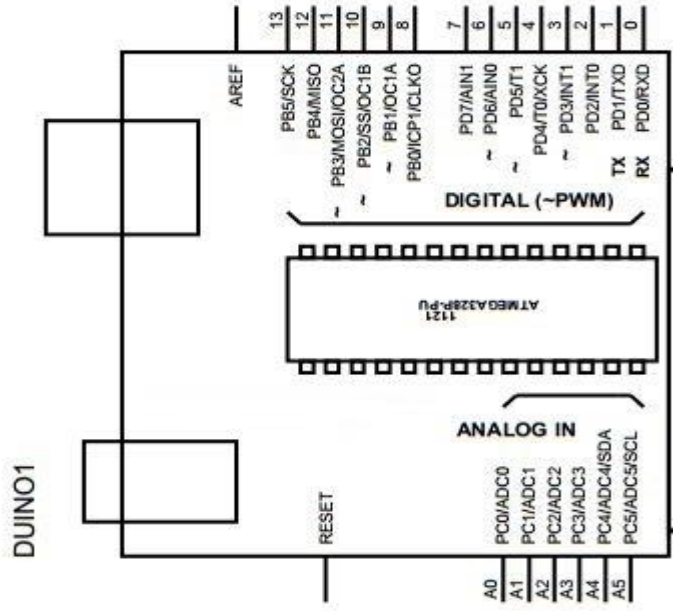


Figure 2-5: Arduino Uno Schematic

2.2.3. Output System

[1] From Appendix datasheet

The output system is designed with the function of the following components.

2.2.3.1 Motor Driver

Motor driver is a current enhancing device; it can also be act as Switching Device. Thus, after inserting motor driver among the motor and microcontroller. Motor driver taking the input signals from microcontroller and generate corresponding output for motor.

IC L293D

This is a motor driver IC that can drive two motor simultaneously. Supply voltage (Vss) is the voltage at which motor drive. Generally, 6V for dc motor and 6 to 12V for gear motor are used, depending upon the rating of the motor. Logical Supply Voltage deciding what value of input voltage should be considered as high or low .So if the logical supply voltage equals to +5V, then -0.3V to 1.5V will be considered as Input low voltage and 2.3V to 5V is taken into consider as Input High Voltage. The Enable 1 and Enable 2 are the input pin for the PWM led speed control for the motor L293D has 2 Channels .One channel is used for one motor.[2]

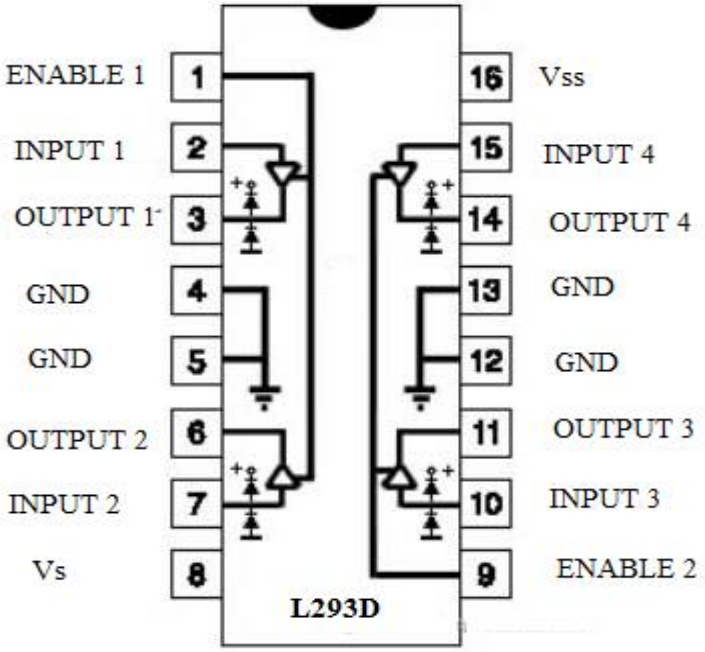


Figure 2-6: Pin Configuration IC L293D

[2] Appendix L293D datasheet

2.2.3.2. DC Motor

Motor is a device that converts any form of energy into mechanical energy or imparts motion. In constructing a robot, motor usually plays an important role by giving movement to the robot. In general, motor operating with the effect of conductor with current and the permanent magnetic field. The conductor with current usually producing magnetic field that will react with the magnetic field produces by the permanent magnet to make the motor rotate.There are generally three basic types of motor, DC motor, even servomotor and stepper motor, which are always being used in building a robot.

DC motors are most easy for controlling. One DC motor has two signals for its operation. Reversing the polarity of the power supply across it can change the direction required. Speed can be varied by varying the voltage across motor.

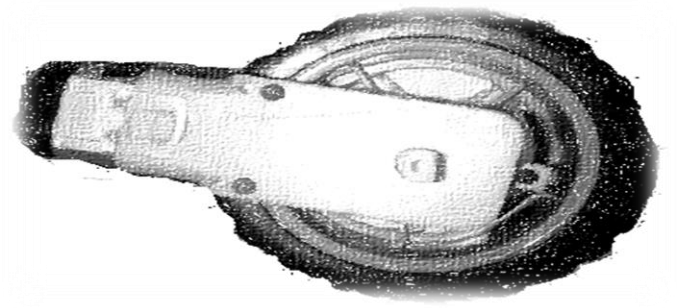


Figure 2-7: Low Volt DC Gear Motor attach with Wheel.

2.2.3.2. LCD Display

In our project it is used to display the distance travelled by the robot through the output from the tachometer system.

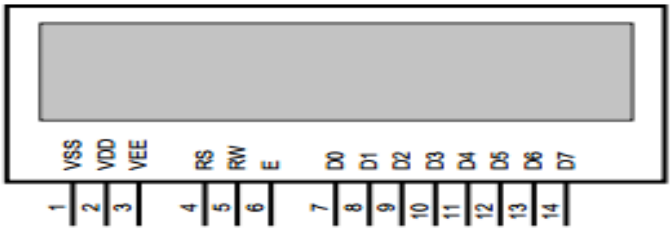


Figure 2-8: Schematic of LCD Panel

2.2.3.3. Proximity Sensor

The combination of IR- LED and Photodiode is used as the reflective optical sensor. It generate interrupt when the IR-beam is break to the photodiode. To create the IR break-beam, IR LED is used with a low value resistor so that it shines very bright. The receiver is Photodiode which biases 'on' whenever the IR LED's light is detected. A sensor will be placed adjacent the IR link and turned on so as to generate a pulse to the Arduino. The Arduino LCD interface is used to show the covered distance digitally.

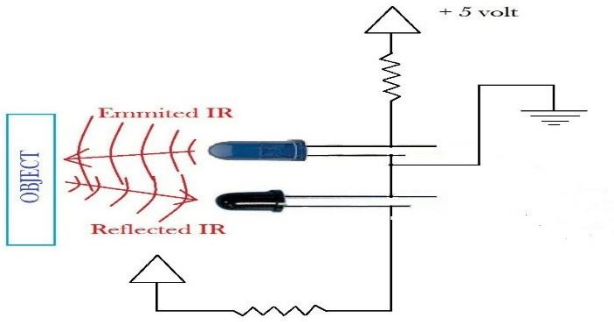


Figure 2-9: Proximity Sensor

2.3. Software Required

For the simulation of the circuit, Proteus® software is used. For coding and uploading the sketch, the Arduino 1.65 ® is used .

2.4. Summary

The system is completed in to the three divison i.e is the Input system, Processing System and Output system.

Input system comprises of the optical sensor, is an array of 4 IR-LED and Photodiode pairs arranged in the form straight lines. The output from each sensor is fed into an analog comparator with the threshold voltage (used to calibrate the intensity level difference of the line with respect to the surface). These 4 signals(from each photo-reflective sensor) is given to a priority analog input of the arduino [3].

Processing system is Arduino Uno interface consisting microcontroller Atmega328 which works on the basic of the logic of the program burn to it.

Output system consist combination L293D and gear-motor The control has 6 modes of operation, turn left/right, move left/right, and drift left/right. The actual action is caused by controlling the direction/speed of the two motors (the two back wheels), thus causing a turn.

CHAPTER 3

3. Design and Implementation

3.1. Schematic

The schematic of the “Line Following Robot” is shown in the figure. The main component is the Arduino Uno. Schematic is drawn by using Proteus.

The main features incorporated into the hardware are given below:

- Ardunio Uno.
- The IR-LED with IR illuminance, modified to be reflective sensor.
- The LM324 quad comparator IC.
- A potentiometer to calibrate the reference voltage.
- The H-bridge motor control IC (L293D)
- Motors, with coupled reduction gears.
- Connectors to join the different boards to form one functional device.
- A pair of IR-LED and Photodiode is used as proximity sensor for the designing techometer

Each of the hardware is dissected and was designed/implemented separately for their functional and later incorporated as one whole application. This helped in the debugging processes.

The schematic of the circuit is given in the next page.

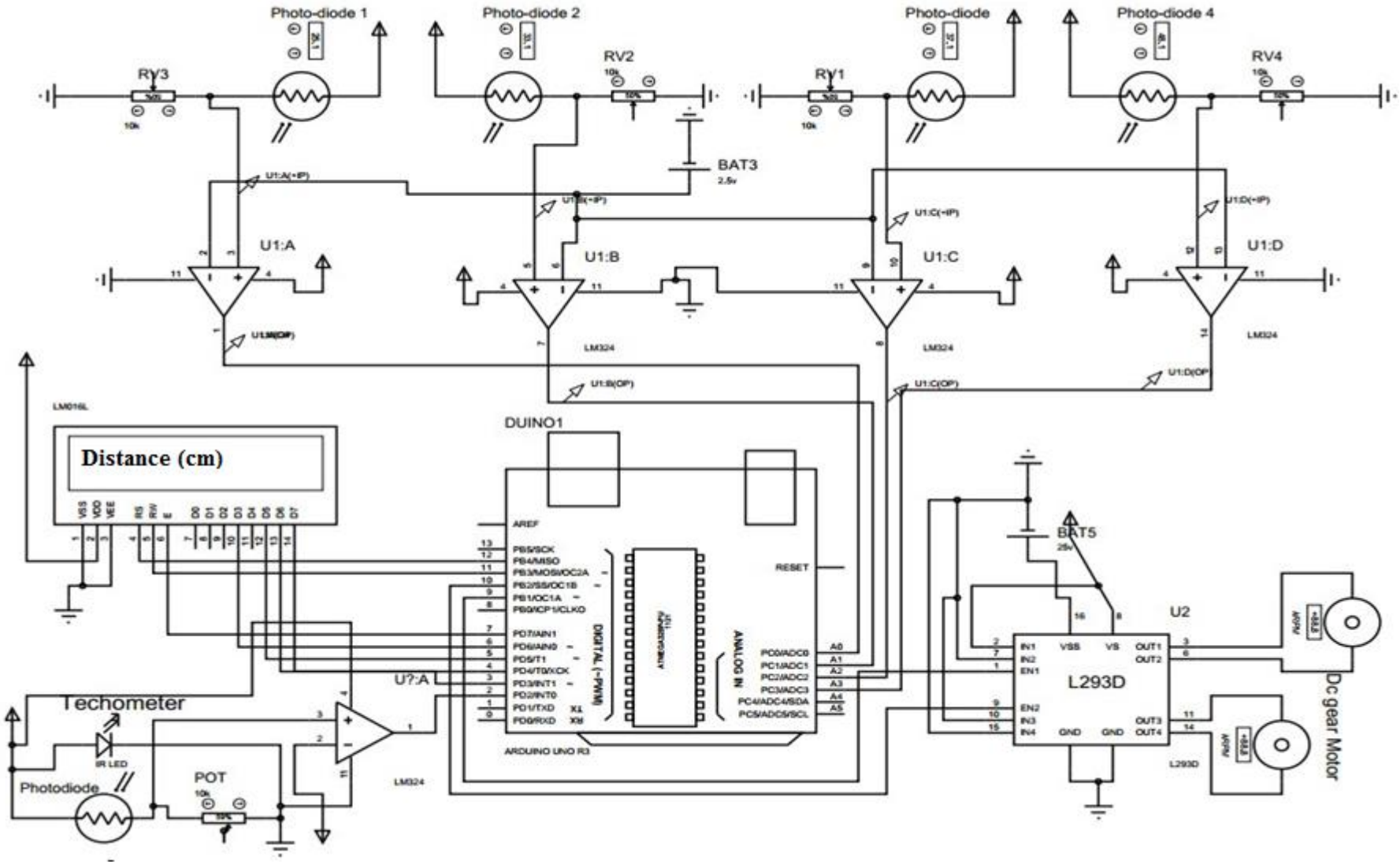


Figure 3-1 :Circuit Diagram Of Line following Robot

3.2. Arduino Working Logic

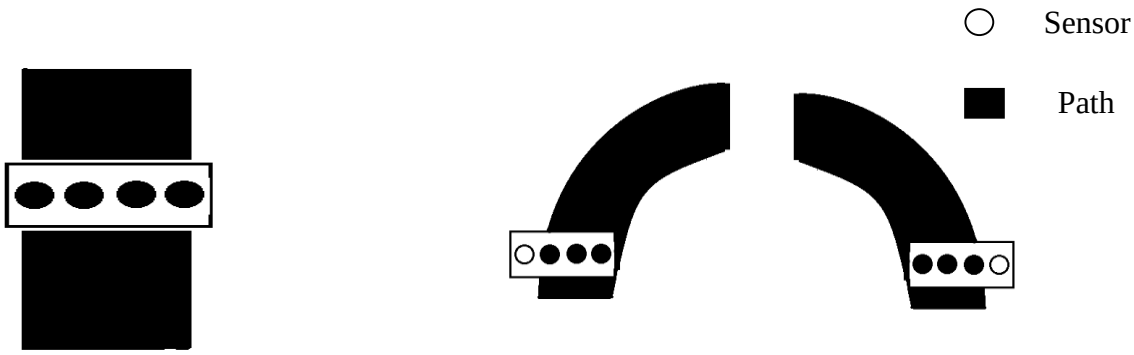
Thus totally the microcontroller gets 4 inputs from the sensor circuitry, to the (A3 – A0) of Arduino to decide what to do when on the line. Below is the complete description about what each input mean and what needs to be done [4].

Table 1: Arduino Working Logic

Input				Output (PWM)		State In	Action
A0	A1	A2	A3	9	10		
0	0	0	0	255	255	All sensor in position	Go straight
1	0	0	0	255	191	Leftmost sensor is out of track.	Move right
1	1	0	0	255	127	Two sensor from the left is out of track	Turn right
1	1	1	0	255	64	Three sensor from the left is out of Track	Sharp turn right
0	0	0	1	191	255	Rightmost sensor is out of track	Move left
0	0	1	1	127	255	Two sensor from the right is out of track	Turn left
0	1	1	1	64	255	Three sensor from the left is out of Track	Sharp turn left
1	1	1	1	0	255	All sensor is out of track	Move circularly Until track is detected

3.3. Process Explanation

As shown in the data below, is a typical situation involved. At every sampled time the commands executed by the Arduino is also shown. From the below figure, it should be clear about the software requirements [5].



.Go Straight

Move right for the right most condition and move left for left most condition

Turn right for the rightmost condition and turn left for left most condition.

Sharp turn right for the rightmost condition and sharp turn left for left most condition.

Move circular until the line is traced down

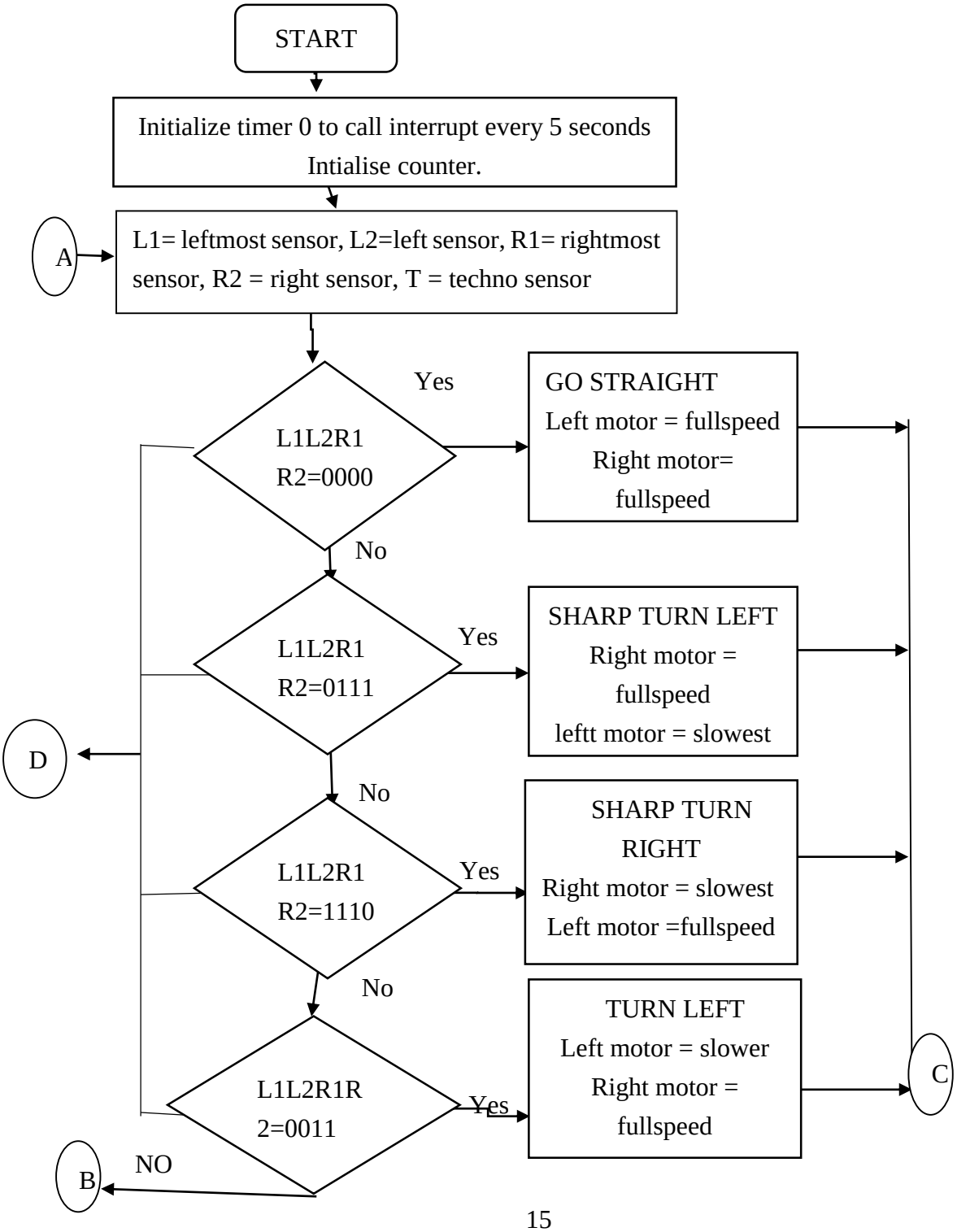
Figure 3-2: Line Following Process

3.4. Programming and Simulation

The program code acts as the decision-maker embedded in the micro-controller deciding about the outputs for particular set of inputs.The program is coded using Arduino® 1.65.and is then compiled to form a ".hex" file which can then be burnt into the Arduino.The output is also checked in simulation using Proteus® [6] .

3.4.1. Flowchart.

The following chart shows implemented logic.



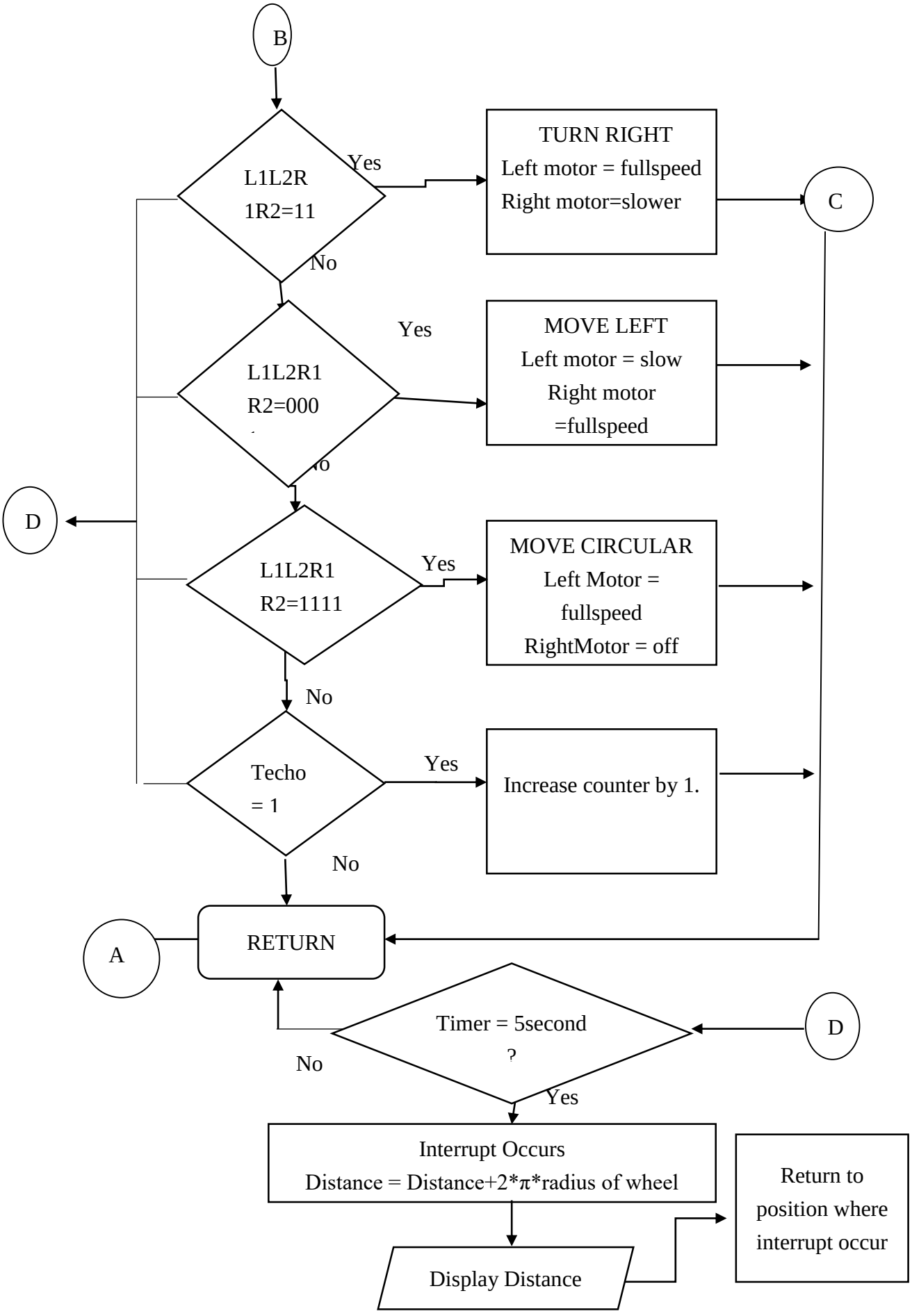


Figure 3-3: Flowchat of the line following Process and Distance Calculation

3.5. Summary

The logic behind the working of Arduino is to analyse the input from the sensor according to program fed to it and provide corresponding output to the the motor driver which finally drive the motor in such way that, it produce required motion.

The differential steering system is implemented to turn the robot. In this system, each back wheel has a dedicated motor while the front wheels are free to rotate. To move in a straight line,both the motors are given the same voltage . To manage a turn of different sharpness, the motor on the side of the turn required is given lesser voltage as level of steering required.

CHAPTER 4

4. Result and Analysis

The practical analysis of the components and the mathematical calculation is performed at different stage.

Dc motor are most easy for controlling. One dc motor has two signals for its operation. Reversing the polarity of the power supply across it can be change the direction required. Speed can be varied by varying the voltage across the motor. The Dc motor don't have enough torque to drive the robot directly. So in order to solve the problem the dc gear motor is which increase the torque of the dc motor in the expense of the speed.

Mathematical Interpretation:

Rotational power (Pr) is given by:

Pr= Torque (T) X Rotational Speed (ω)

$$\Gamma = \frac{pr}{\omega}$$

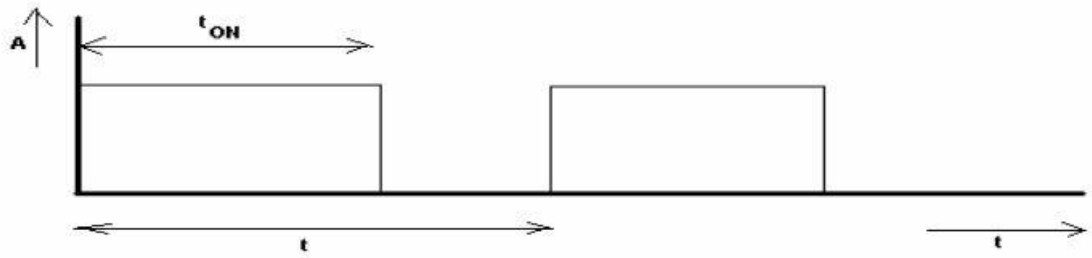
Pr is constant for DC motor for a constant input electrical power. Thus torque (T) is inversely proportional speed (ω)

$$\Gamma \propto \frac{1}{\omega}$$

4.1. Speed Control

The speed of the dc motor is control by feeding PWM from Arduino to the enable pin of the L293D which change the voltage across the motor. Due to which speed is also decreased. On the command over the speed and the direction is also controlled.

4.2.1. PWM Speed Control



Calculation:

$$\mathrm{A} = \frac{1}{t}\int_0^{ton} Vdt$$

$$\mathrm{A} = \frac{1}{t}\int_0^{ton} 12dt$$

$$A = 12\frac{ton}{t}$$

$$A = 12 \times \rho$$

Where ρ is the duty cycle of the PWM control signal; this shows that by varying the duty cycle of the PWM control,we effectively vary the DC voltage supplied to the motors, thus controlling their speed. This is generated by the arduino built-in hardware [9].

4.2.2. Practical Result Analysis

Here the motor is tested under all condion in the lab. The reading of the motor RPM and the voltage across it is illustrated below. Inertia is neglected.

Table 2: Analysis of effect of PWM on RPM of Motor

Duty cycle (ρ)	Voltage Across Motor	RPM
0%	0	0
25%	3	16
50%	6	29
75%	9	45
100%	12	58

4.2. Digital Revolution Counter

A Digital RPM counter is a measuring instrument which can measure the number of a rotation machine digitally. The similar concept is used in instrument called “Tachometer”. It is an important measuring device in the field of electrical engineering & widely used in industries and laboratorial work. IR- Pair is the heart of the system which found the number of revolution. It works by detecting the black spot in the wheel and transfer digital high to the arduino.

4.3.1. Calculation

The system is designed in such way that it updates the distance value in every 5 seconds. i.e. 12 times per minute.

4.3.2. Practical Result Analysis

Here the build revulution counter is used as Tachometer tested under all condion in the lab with reference of standard Tachometer. The reading of the motor RPM is illustrated below.

Table 3: Comparison between standard Tachometer and Build Tachometer

Voltage Across Motor	RPM (Standard Tachometer)	RPM (Build Tachometer)	Percentage Error
3	70	75	6.25%
6	120	132	9.3%
9	184	173	6.67%
12	258	235	10.4%
15	330	284	12.4%

4.4. Summary

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, change or modulate that pulse width.

Since, the build techometer can be used, as from the analysis the error percentage within limit, for the maximum rpm is to be measured i.e 250 rpm.

CHAPTER 5

5. Kinematics of the Robot

The backbone of our design is the differential steering system which is familiar from ordinary life because it is the arrangement used in a wheelchair. Two wheels mounted on a single axis are independently powered and controlled, thus providing both drive and steering. Additional passive wheels (usually casters) are provided for support. Most of us have an intuitive grasp of the basic behavior of a differential steering system. If both drive wheels turn in tandem, the robot moves in a straight line. If one wheel turns faster than the other, the robot follows a curved path. If the wheels turn at equal speed, but in opposite directions, the robot pivots [7].

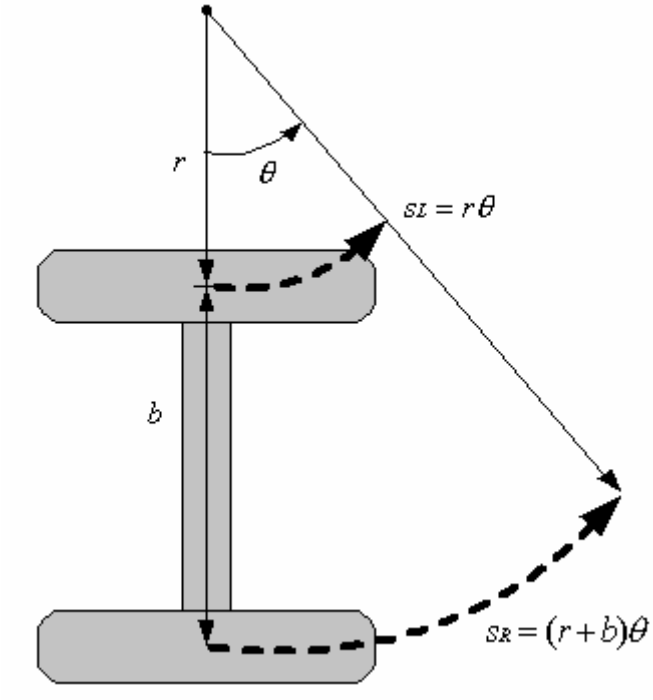


Figure 5.5-1: The Differential steering model

$$SL = r\theta$$

$$SR = (r + b)\theta$$

$$SM = \left(r + \frac{b}{2}\right)\theta$$

Where SL and SR give the displacement for the left and right wheels respectively, r is the turning radius. b is the distance between wheel, and θ is the angle of turn. SM is the speed at the center point as the main axle.

5.1. Mechanical Design

The chassis of the robot is made up of the acrylic glass since it can carry more load and have lighter in weight.

The detail orthographic projection of the robot is drawn at the appendix.

Design of Path

The path consists of three U-turns of different radii. Among three, the system swiftly turn two U-turn and it goes out of track in one initial U-turn and retrace it's path.

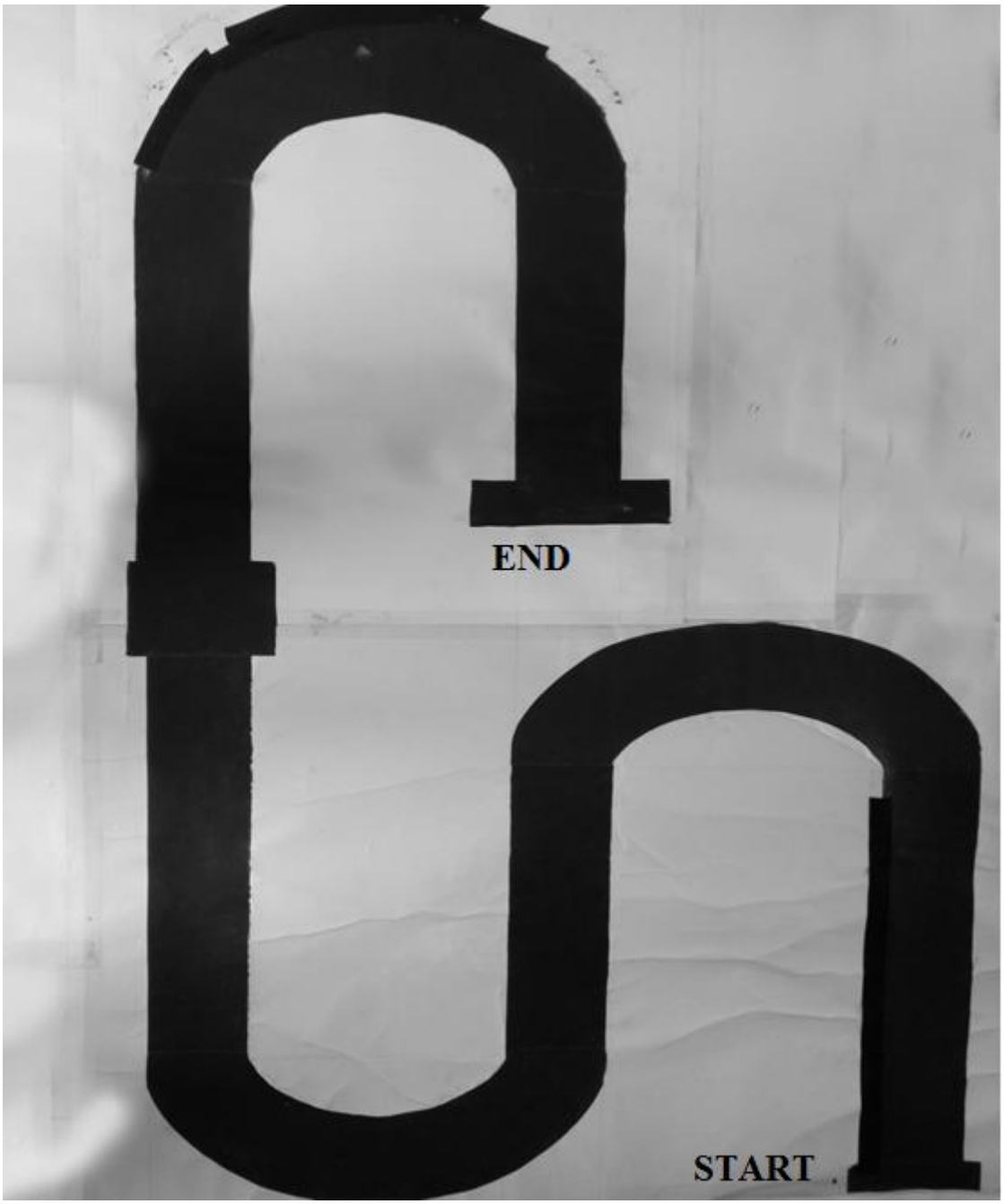


Figure 5-2: Path of the Line Following Robot

Picture of Line Following Robot

The picture of the line following robot is captured by the still camera at some angle and shown below.

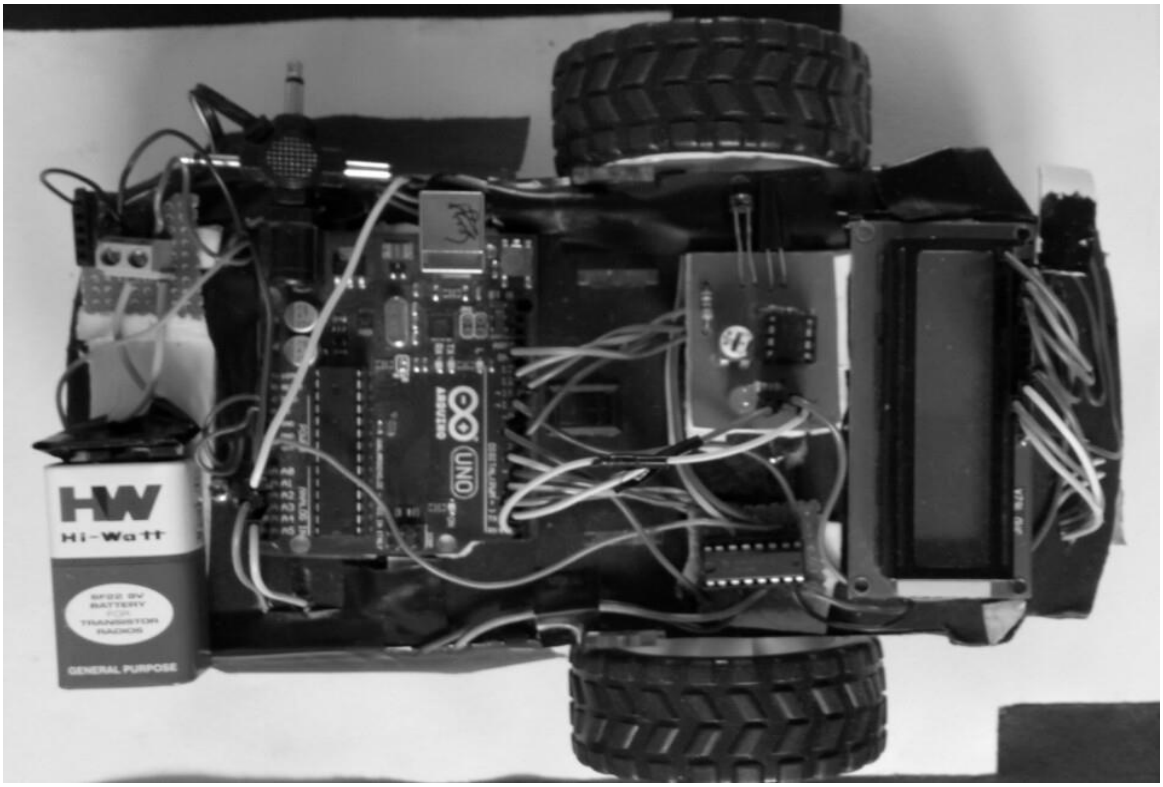


Figure 5-3: Top View of the Line following Robot

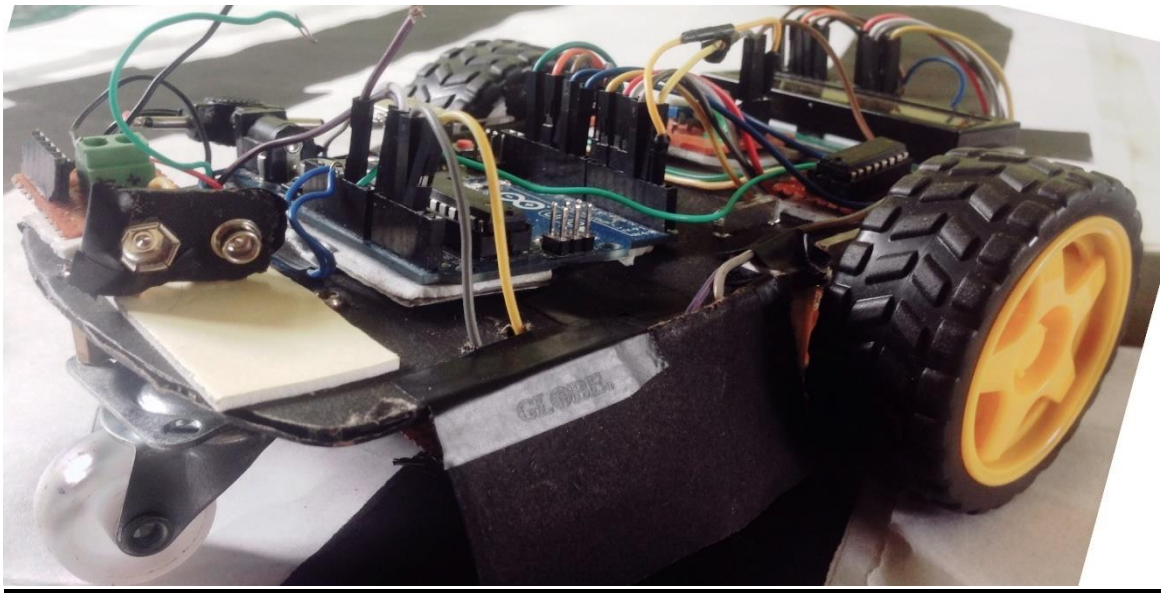


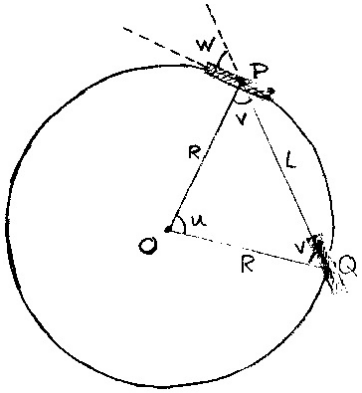
Figure 5-4: Side View of the Line Followinng Robot.

5.2.1 Turning Radius

The turning radius of a vehicle is the radius of the smallest circular turn (i.e. U-turn) that the vehicle is capable of making. The term 'turning radius' is a technical term that is commonly used to mean the full diameter of the smallest circle, but in technical usage the turning radius still is used to denote the radius.

Calculation :

The total length of the robot car is 172mm (approx 17.2mm.) the minimum turning radius that it can turn sharply.



The diagram in which robot car is making a turn. The situation is show in the figure above. The wheels are shown shaded in the figure. The front wheel's center is *P* and the rear wheel's center is *Q*. Let *O* be the center of the circle, along whose circumference the car moves. Let *L=PQ* be the length of the car and let *R* be the radius of the circle. Sharpest turn corresponds to smallest possible value of *R*. Note that *PQ* is a chord of the circle. Let us say that the front wheel can rotate at most by an angle *w* relative to the 'straight' position. For triangle OPQ [11].

$$u + 2v = 180^0 \qquad (1)$$

also , $w + v = 90^0$ (2)

From, cosine rule for OPQ:

$$L^2 = R^2 + R^2 - 2R^2 \cos(u) \qquad (3)$$

From eq^n (1), (2) and (3)

$$R = \frac{L}{2sinw}$$

At the maximum value of sinw i.e 1 . The turning radius is found to be L/2 = 87.2mm.

Now, the velocity of the automobile is the prime factor to determine the ability to take turn smoothly or not. So we have to derive the relation between the the turning radius and The velocity of the robot car.

Turning Radius(r) = 100mm.

Acceleration due to gravity(g) = 9.81m/s^2 = 9.81×100 mm/s^2

Velocity(v) = ?

Coefficient of friction between rubber and cardboard (μ) = 0.07 [11]

$v = \sqrt{\mu r g}$

$= \sqrt{0.07 \times 8.75 \times 9810}$

= 245.20mm/s = 24.52cm/s

Which is attained at the speed of 1.4rps (or, 84 rpm).

5.2. Cost Estimation

The details of price of components used in project is given below.

Table 4: Details of Price of components

S.N.	Particulars	Quantity	Rate per item(NRs.)	Total(NRs.)
1	Arduino	1	2000	2000
2	Chassis	1	150	150
3	Motor	2	500	1000
4	Rear Wheel	2	100	200
5	Front Wheel	1	150	150
6	LCD	1	450	450
7	Motor Driver	1	350	350
8	Jumper Wire	20	10	200
9	Photodiode	5	30	150
10	IR LED	5	15	75
11	LM 358	1	80	80
12	LM 7805	1	45	45
13	AAA Battery	4	15	60
14	9V Battery	1	80	80
15	Battery Holder	1	50	50
16	Switches	1	5	5
17	Miscellaneous			1000
			Total	**6045**

Hence, the product is finalised in NRs. 6045 excluding the labor cost.

5.3. Summary

The main criteria of the mechnical design is mainly depends upon the differential steering mechanism, turning radius. The caculation of turning radius is necessary because it finds out the mechanical limitation for the types of the curve which is avoded. The turning radius for the design is found to be 100mm .

So, in case of designing path the maximum turning radius mustn’t be less than 100mm.

CHAPTER 6

6.1. Conclusion

The line following robot is automobile system that has ability to recognize it’s path , move and change the robot's position toward the line in the best way to remain in track.
This project report presents a photodiode sensor based line follower robot design of 200gm weigh which always directs along the black line on white surface.The electromechenical robot dimension is 192×100×70 mm^3 with max rpm 180 at no load and frictionless condition. The minimum turning radius for the system is 100mm at velocity of 24.2 cm/s. The robot is able to detect it’s path in case it is out of path.

The line following robot project challenged the group to cooperate, communicate, and expand understanding of electronics, mechanical systems, and their integration with programming. The successful completion of every task demonstrated the potential of mechatronic systems and a positive group dynamic.

6.2. Future Work

In the process of development of the line follower, most of the useful feature is identified and many of them was implemented . But due to the time limitations and other factor some of these cannot be added.
So the development features in brief:

- Use of color sensor.
- Use of ccd camera for better reconigsation and precise tracking the path..

GANTT CHART

Table 5: Gantt Chart

Task	Sept. 2015	Nov. 2015	Dec. 2015	Jan-Feb. 2016	March 2016	April-May 2016	June-July 2016
Study of possible project							
Project Title Selection							
Literature Review							
Proposal Writing And Defense							
Circuit design and implementation of the circuit							
Mid-Term Report							
Hardware and PCB development							
Testing and evaluation of the system							

	Work Accomplished
	Work Remaining

BIBILOGRAPHY

[1] B. Klaus and P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.

[2] J. Warren, J. Adams and H. Molle, "Arduino for Robotics," in *Arduino Robotics*, New York, Apress publiction, 2014, pp. 51-83.

[3] Jim, "PWM/PID/Servo Motor Control," 2005. [Online]. Available: http://www.uoxray uoregon.edu. [Accessed 15 December 2015].

[4] Komonya, S. Tachi, K. Tanie, "A Method for Autonomous Locomotion of .Mobile Robots,"in *Journal of Robotics Society of Japan*, vol. 2, pp.222-231, 1984.

[5] S. Monk, Programming Arduino Getting Started with Sketches, New Delhi, India: Tata Macgrawhill, 2012.

[6] Open Source community, " Open Source Sketch," January 2015. [Online]. Available: https://www.arduino.cc/en/Guide/Introduction. [Accessed 25 November 2015].

[7] A. Parsad, "Line Following Robot,"Dept. Elex. & Comm. Eng., Visvesvaraya Technological University, Banglore, India, 2005

[8] S. Bhatia, "Engineering garage," 23 May 2011. [Online]. Available: http://www.engineersgarage.com//tachometer-microcontroller-circuit-project. [Accessed 03 March 2016].

[9] S. Debopath and Md. Jinnah, "Digital RPM meter using Arduino,"Dept. Elect.Eng., Ahsanullah Univ. of Sci. & Tech., Dhaka, Bangladesh, 2012.

[10] Zin, "Alldatasheet," March 2003. [Online]. Available: http://www.alldatasheet.com [Accessed 06 December 2015].

[11] Wordpress"Turning radius of the Car," July 2013 [Online]. Available: https://goodmaths.wordpress.com/2013/07/19/turning-radius-of-a-car/. [Accessed 28 June, 2016]

APPENDICES

PROGRAM CODE

Here code is developed in the same view represented by the flowchart. Code accompany with algorithm side of it.

```
;============================================================;
;==============LINE FOLLOWING ROBOT ========================;
;====================Aurduino 1.65=============================;
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 7, 5, 4, 3, 6);
float distance;
volatile int count = 0;     // volatile is a keyword known as a variable qualifier.
int leftInput1=A2; // Input from the Sensor send to various pin of Arduino from A0 – A3.
int leftInput2=A3;
int rightInput1=A0;
int rightInput2=A1;
int leftMotor= 9;        // Arduino output PWM to motor for speed control.
int rightMotor= 10;
int leftValue1 = 0; // The digital value assigned to the digital read from analog pin from the Arduino.
int rightValue1 = 0;
int leftValue2 = 0;
int rightValue2 = 0;
unsigned long lastmillis = 0;    // Asigned lastmillis variable to store the current time value.
void setup()
   // put your setup code here, to run once:
   {
Serial.begin(9600);
  lcd.begin(2, 20);    // setup instruction for LCD 2×20.
lcd.setCursor(0,0);    //Intialise the position of the cursor for print.
lcd.print("distance (cm)"); // The title print in the LCD displays

   pinMode (leftMotor, OUTPUT);    // output is the vale assign to left motor
   pinMode (rightMotor, OUTPUT);    // output is the value assign to Right motor
   attachInterrupt(0, techno, RISING);//interrupt zero (0) is on pin two(2) activate on rising i.e low to high.
```

```
}
void loop()
  // put your main code here, to run repeatedly:
{
  leftValue1 = digitalRead (leftInput1); // digital data from the input sensor is taken
  rightValue1= digitalRead (rightInput1); // there are all together four sensors two at the
left most and two at the rightmost
  leftValue2 = digitalRead (leftInput2);
  rightValue2= digitalRead (rightInput2);
   if(millis() - lastmillis >= 6000) //Uptade every Six seconds
   {
    detachInterrupt(0)
    distance = distance + 2*3.1416*3*count;
    lcd.setCursor(0,1);
     lcd.print(distance);
     count = 0; // Restart the RPM counter
 lastmillis = millis(); // Update lastmillis to maintain count difference 6 seconds.
 attachInterrupt(0, techno, RISING);
} if
   ( leftValue1 == HIGH && leftValue2 == HIGH && rightValue1== HIGH &&
rightValue2 == HIGH)
   {
    digitalWrite(leftMotor,LOW);   //When all the sensors are out of track it stop for two
sends and begin                                                    search for
path for 1sec and again repeat
    digitalWrite(rightMotor,LOW);
    delay(2000);
    digitalWrite(leftMotor,LOW);
    digitalWrite(rightMotor,HIGH);
    delay(1000);

  }
   else
 if
     ( leftValue1 == HIGH && leftValue2 == LOW && rightValue1== LOW &&
rightValue2== LOW)
    {
```

```
    analogWrite (leftMotor, 191);   //The left motor moves with slightly slow speed while
right motor moves   with slower speed
    analogWrite (rightMotor,185);   //The   robot moves turns toward the right
    //delay(1000);
      }
 else
    if ( leftValue1 == HIGH && leftValue2 == HIGH && rightValue1== LOW &&
rightValue2== LOW)
    {
    analogWrite (leftMotor,191 );   //The left motor moves with slightly slow speed while
right motor moves   with slower speed
    analogWrite (rightMotor, 64);   //The   robot moves turns toward the right
   // delay(1000);
    }

   else
    if ( leftValue1 == HIGH && leftValue2 == HIGH && rightValue1== HIGH &&
rightValue2== LOW)
    {
    analogWrite (leftMotor,255 );   //The left motor moves with slightly slow speed while
right motor moves   with slower speed
    analogWrite (rightMotor,64);   //The   robot moves turns toward the right
    //delay(1000);
    }
    Else
 if ( leftValue1 == LOW && leftValue2 == LOW && rightValue1== LOW &&
rightValue2== HIGH)
       {
       analogWrite (leftMotor, 127);   //The left motor moves with slightly slow speed
while right motor moves   with slower speed
    analogWrite (rightMotor,191);
   // delay(1000);
       }
       else
       if( leftValue1 == LOW && leftValue2 == LOW && rightValue1== HIGH &&
rightValue2== HIGH)
       {
```

```
            analogWrite (leftMotor, 64);   //The left motor moves with slightly slow speed while right motor moves   with slower speed
    analogWrite (rightMotor,191);
  // delay(1000);
      }
      else
      if( leftValue1 == LOW && leftValue2 == HIGH && rightValue1== HIGH && rightValue2== HIGH)
      {
            analogWrite (leftMotor,64);   //The left motor moves with slightly slow speed while right motor moves   with slower speed
    analogWrite (rightMotor,255);
  // delay(1000);
      }
      else if ( leftValue1 == LOW && leftValue2 == LOW && rightValue1== LOW && rightValue2== LOW)
      {
       analogWrite (rightMotor,255); //   The robot moves with full speed when all of the sensor is low.
        analogWrite (leftMotor,255 );
       }   }
      void techno()
  {
  count ++;   }
```