East West University

Department of EEE

Project Title

Design and Implementation of a Line Following Robot with Temperature Sensing and Comparing Capability

By

Sabbir Ahmed: 2007-2-80-028

Md. Salahuddin Khan: 2007-2-80-022

Submitted to the
Department of Electrical and Electronic Engineering
Faculty of Science and Engineering
East West University
Dhaka, Bangladesh

In partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering
(B.Sc in EEE)

Fall 2011

Approved by

Project Advisor

Dr. Muhammed Mazharul Islam

19.01.2012

Department Chairperson

Dr. Anisul Haque

ABSTRACT

Line following robot is a kind of robot which can detect a path drawn in a surface. This can be a black line in white surface or white line in a black surface. In this project we made a line following robot which can follow a white line drawn on a black surface. The robot has another feature – the ability to sense the temperature. So, the objective to make this robot is to follow a white line in black surface and also sense the temperature in its path. The robot is programmed so that it can measure temperature in its path at an interval of 30 seconds. The application of this robot is in temperature sensitive areas like cold storage, chemical industries, medicine manufacturing farms etc. In these areas, maintenance of temperature is an important issue, so we can easily use this robot. The principle of detecting the path uses the reflectivity of light. Black surface absorbs light and white surface reflects it. We have used seven LEDs (Light Emitting Diode) and seven LDRs (Light Depending Resistor) to detect the path. An LED and an LDR together make a single sensor circuit. When the sensor is over a white surface, the light from the LED reflects from the surface to the LDR and due to the decrease of resistivity of the LDR, the voltage across the LDR decreases. The output of each sensor circuit works as an input of a comparator (IC LM324) which compares the output voltage of LDR to a reference voltage which is given manually. If the LDR output voltage is higher than the reference voltage, then the comparator outputs logic high; otherwise the comparator output is logic low. So the comparator IC converts the analog signal to digital. As we have used seven sensors to detect path, so we have a combination of seven bits of logical data. Value of this combination always changes during the path following. We use those combinations to the input pins of the microcontroller as the digital data. In our project we have used microcontroller ATmega16A from AVR. The microcontroller gives output (according to the data and the program code) to a motor drive IC

L293D.The motor driver IC can operate two motors separately. Our robot has three wheels in total. Two of them are for controlling position and are connected to the two motors while he third one issued for balance. When the robot has to go right, the left motor runs and the right motor remains stopped. When it has to move to the left, the right motor runs and the left motor remain stopped. When the robot moves forward then the both motors run at same speed. We have used a temperature sensor IC LM35. The LM35 senses temperature and converts it to voltage. For one degree Celsius change of temperature, its output voltage changes by 10mV.The output voltage of the LM35 goes to microcontroller. The microcontroller takes the voltage as input and displays the temperature on a seven segment display in degree Celsius. A reference temperature is externally set and if the robot finds that the current temperature that is detected on its path is more than the reference temperature, it will stop at that location and will alert by showing an error sign on the seven segment display while simultaneously and blinking a red light.

ACKNOWLEDGEMENTS

We would like to thank Dr. Muhammed Mazharul Islam, Assistant Professor, Department of Electrical and Electronic Engineering (EEE), East West University (EWU), Dhaka, who is our supervisor for this project- for his constant guidance, supervision, constructive suggestion and constant support during this project work.

We are grateful to Dr. Anisul Haque, Professor and Chairperson, Department of Electrical and Electronic Engineering (EEE), East West University (EWU), Dr. Khairul Alam, Associate Professor, Department of Electrical and Electronic Engineering (EEE), East West University (EWU) and Dr. Halima Begum, Assistant Professor, Department of Electrical and Electronic Engineering (EEE), East West University (EWU), for their valuable suggestions and help. We would also like to thank our advisors S.M. Shahriar Rashid and F. Mashuque Alamgir and other faculty members, office staffs and EWU in general.

At last we want to thank our parents and all our friends for their moral support and helpful discussion during this work.

APPROVAL

The project work titled "Design and Implementation of a Line Following Robot with Temperature Sensing and Comparing Capability" submitted by **Sabbir Ahmed (2007-2-80-028)** and **Md. Salahuddin Khan (2007-2-80-022)**, session Fall, 2011, has been accepted as satisfactory in partial fulfillment of requirement of the degree of Bachelor of Science in Electrical and Electronic Engineering on December,2011.

Dr. Anisul Haque

Professor and Chairperson

Department of Electrical and Electronic Engineering

East West University

Dhaka, Bangladesh

AUTHORIZATION PAGE

We hereby declare that we are the sole authors of this project. We authorize East West University to lend this project to other institutions or individuals for the purpose of scholarly research.

Sabbir

Sabbir Ahmed

Salahuddin

Md. Salahuddin Khan

We further authorize East West University to reproduce this project report by photocopy or other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Sabbir

Sabbir Ahmed

Salahuddin

Md. Salahuddin Khan

East West University

Dhaka, Bangladesh

December, 2011

Authors

Table of Contents

ABSTRACT i
ACKNOWLEDGEMENTS iii
APPROVAL iv
AUTHORIZATION PAGE v
Table of Contents vi
Chapter 1 : Introduction 1
1.1 Objective 1
1.2 Line following robot 1
1.3 Main features 2
Chapter 2 : Working Principle of a Line Following Robot 3
2.1 Introduction 3
2.2 Sample Paths 3
2.3 Line Following Principle 4
2.4 Control from any undesired situation 7
2.5 Temperature sensing 8
2.6 Programming flow chart for microcontroller: 9
2.6.1 Description of program 10
Chapter 3 : Major Components of a Line Following Robot 13
3.1 Sensors 13
3.1.1 Light Dependent Resistor (LDR) 13
3.1.1.1 LDR as a Sensor 14
3.1.1.2 Arrangement of Sensors 15
3.1.2 LM324 Comparator IC 16
3.1.3 Inverter 17
3.1.4 Temperature sensor (LM35) 17
3.2 Mechanical Components 18
3.2.1 Wheels 19
3.2.2 Motor 20
3.2.2.1 Advantage of Gear Motor 20

3.3 Motor driver IC (L293D) 22
3.4 Seven segment display 23
3.5 Microcontroller 24
3.5.1 Why AVR 24
3.5.2 Kinds of AVR 25
3.6 Power supply 26
Chapter 4 Construction 27
4.1 Introduction 27
4.2 List of Equipment 27
4.3 Procedure 28
4.3.1 Step 1: Making the base 28
4.3.2 Step 2: Attaching the motor 29
4.3.3 Step 3: Attaching wheels 29
4.3.4 Step 4: Placing the third wheel and connecting the wires to the motors 30
4.3.5 Step 5: Making the sensor array to detect path 31
4.3.6 Step 6: Construction of the main circuit 33
4.3.7 Step 7: Placing the circuit on the robot's base 34
4.3.8 Step 8: Making a cover for the robot 34
Chapter 5 : Discussion 37
5.1: Problems we faced 38
5.2: Suggestions for Future Development 39
Chapter 6 : Conclusion 40
References 41
Appendix A 42
Program code 42
Appendix B 47
Circuit Diagram 47

Table of figures

Figure 2-1: Sample paths 4
Figure 2-2: Block diagram of operation of the robot 5
Figure 2-3: Movement criteria for forward and backward 6
Figure 2-4: Movement criteria for towards left and right 6
Figure 2-5: Movement criteria for rotational movement 7
Figure 2-6: Process to find the path when the robot goes out of line 8
Figure 2-7: Flow chart for programming the microcontroller 9
Figure 3-1: Light Dependent Resistor (LDR) [1] 13
Figure 3-2: reflectivity of light from black and white surfaces. 14
Figure 3-3: Complete circuit diagram for a single sensor 15
Figure 3-4: Arrangement of LDR sensors 15
Figure 3-5: LM324 Comparator IC [2] [3] 16
Figure 3-6: Example of comparing voltage 16
Figure 3-7: Inverter IC 7404 [4] 17
Figure 3-8: LM35 temperature sensor [6] 18
Figure 3-9: Arrangement of wheels 19
Figure 3-10: 12V gear motor 21
Figure 3-11: L293D motor driver IC [7] [8] 22
Figure 3-12: Block diagram of operation of L293D [10] 23
Figure 3-13: 4 digit common cathode 7- segment display [11] 23
Figure 3-14: ATmega16A microcontroller [12] [13] 24
Figure 3-15: Voltage regulator (7805) [16] 26
Figure 4-1: Small pieces of hard board attached with main board 28
Figure 4-2: Motor attached with small pitches of hard board by screw 29
Figure 4-3: Wheels from toy car and supporting wheels 30
Figure 4-4: Attached wheel to the motor 30
Figure 4-5: Complete Lower side of the robot 31
Figure 4-6: Sensor array to detect path (front view) 32
Figure 4-7 Sensor array to detect path (back view) 32
Figure 4-8: Main circuit containing potentiometer, LM324, L239D and 7805 33
Figure 4-9 Main circuit containing microcontroller, switches and display 33
Figure 4-10: Complete circuit board 34
Figure 4-11 Complete robot (top view) 35
Figure 4-12 Complete robot (bottom view) 35
Figure 4-13: Complete robot 36
Figure B1: Circuit Diagram 47

Chapter 1 : Introduction

1.1 Objective

In our project, we made a line following robot which can follow a white line on a black surface and has the ability to continuously monitor the temperature on its path. The objective to make this robot is to sense the temperature in temperature sensitive areas such as in a cold storage or a food reserve or chemical industries etc. It is very important to measure the temperature in cold storages. This is because, if the food does not get proper temperature in cold storage, it can get rotten and thus poison other nearby foods. Usually the storage areas are very large and it is almost impossible for people to monitor the temperature manually. Another option for doing this particular job is to set some permanent temperature sensors on some places. But implementation of this technique is not cost effective and maintenance is also very difficult for poor countries like Bangladesh. Our robot is capable of detecting temperatures in almost all the areas of a cold storage and maintaining this robot is very easy and it is also cost efficient than other complex temperature monitoring systems. This is a microcontroller based project. Besides the line following, we have to set a reference temperature manually and if the robot senses a high temperature than the reference temperature in its path then it will stop and show an error alert on 7-segment display and blink a red light. So this robot can very useful for detecting the temperature in any area of a cold storage.

1.2 Line following robot

Line following robot is a kind of robot which can follow a line drawn on a surface. It can be a black line on white surface or white line on a black surface. The robot can automatically detect the path and follow this path according to its feedback system. So basically a line following

robot is a programmed device which can detect a path and follow it. There are many applications of line following robots such as to transfer materials from one place to another in a storeroom, working as a waiter in a restaurant etc. In our project we've used the line following robot to detect temperature in its path.

1.3 Main features

The main features of our robot

1. It can follow a white line (1.5cm wide) on black surface
2. The sensitivity of the sensor circuit during line following can be controlled externally
3. It has the ability to sense temperature
4. The reference temperature can be set externally
5. It displays the temperature in degree Celsius on a seven segment display
6. If the robot finds any higher temperature than a reference temperature adjusted to it, it will show error sign in a seven segment display blink a red light and stop.
7. Reset button for hardware reset to reactivate the robot when error occurred.

We made this robot by combining the concept of line following robot and temperature detecting and comparing technique, especially for temperature sensitive areas. We hope it will be very useful in those areas.

Chapter 2 : Working Principle of a Line Following Robot

2.1 Introduction

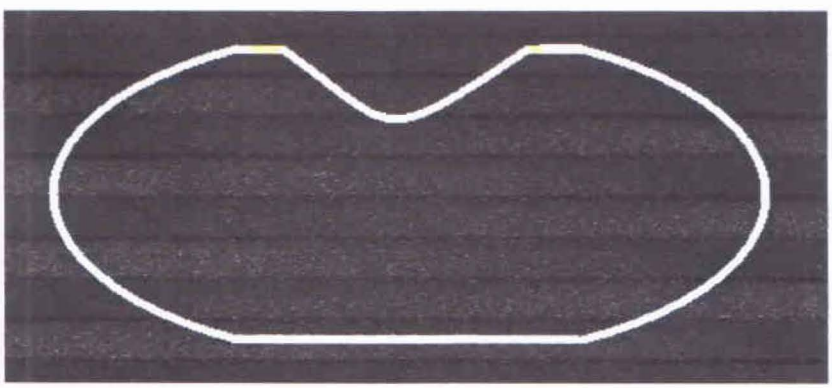
In our project we have prepared the line following robot to follow a white line on black surface. The robot is also responsible for measuring temperature on its path. So the basic job of the robot is to follow a path, sense temperature and display it. Moreover it has some extra task to control itself from undesired situations (e.g. if somehow the robot goes out of the line). To perform all these tasks, the robot needs several types of sensors.

We have used two kinds of sensors in our robot. These are

1. Light Dependent Resistor (LDR) sensor
2. Temperature sensor (LM35)

The LDR sensor is used to detect the path. The temperature sensor is used to sense the temperature. We get voltage as output from LDR sensors. Then this voltage is compared with a reference voltage in comparator. The comparator IC gives outputs 0V or 5V as output that can be used as digital signal at the input of a microcontroller. The microcontroller is programmed so that it can provide corresponding output.

2.2 Sample Paths



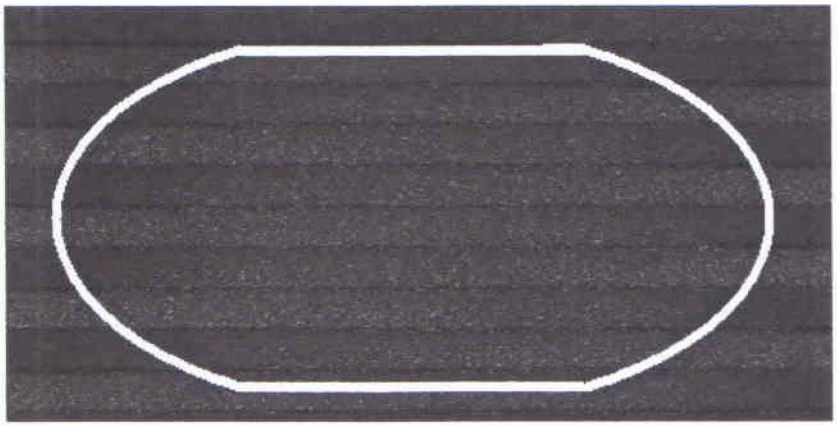


Figure 2-1: Sample paths

To reduce complexity, the robot is programmed so that it can follow the simple paths where no obstacle will interrupt the robot on its path. The robot can perfectly turn during the line following where the turning angle of the path is greater than 65°.

Actually there is no special reason why we use white line on black surface. The path could be black on white surface too. Here we consider that most of the surface color in cold storages is dark colored. So, white line on a black surface will be more preferable in this case.

2.3Line Following Principle

The principle of following a white line on black surface is the reflectivity of light. The white surface reflects light and black surface absorbs it. The LDR sensor has different resistive values depending on the amount of light incident on it and hence we get two different voltage levels corresponding to a black or white surface. This analog output works as input to a comparator IC. We set a reference voltage to the comparator IC. If the voltage level from the sensor is higher than the reference voltage, then we get logic high at the comparator output. If the voltage level of the sensor is less than the reference voltage, then we get logic low. Thus the comparator IC converts the analog data to digital data. We have used seven sensors to detect path. Because the

arrangement of seven sensors will help to detect the path more accurately when turning left or right. The details discussions on sensor arrangement are discussed at chapter-3, section 3.1.2.

Now we get digital combinations of seven bits according to the position of the robot on the line. Depending on the combinations, the robot can take a decision to move right, left, forward, backward or stop. These conditions are programmed in the microcontroller. The microcontroller output is connected to a motor driver IC (L293D).

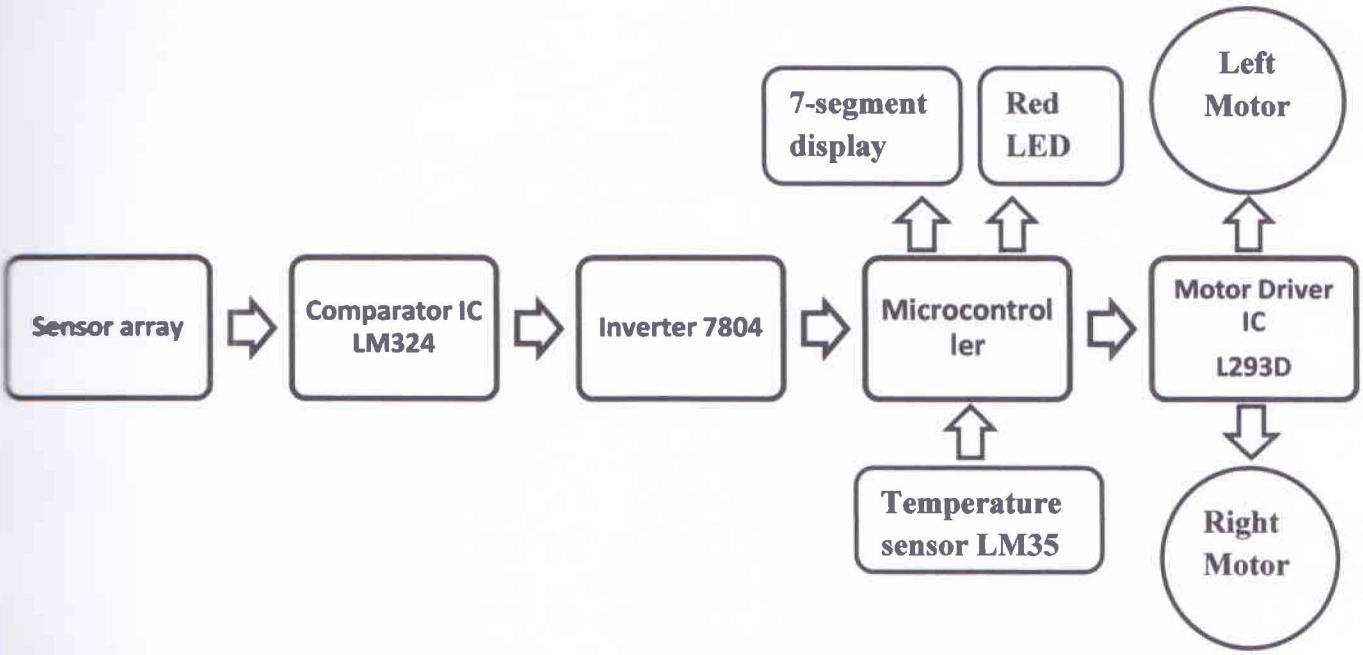


Figure 2-2: Block diagram of operation of the robot

The motor driver IC can control two motors separately. Therefore if the robot needs to turn right, the left motor will run forward and the right motor will stop. If the robot needs to turn right, the left motor will run forward and the right motor will stop. If the robot needs to go forward, both

motor will run forward. For rotational movement one wheel will run forward and another backward.

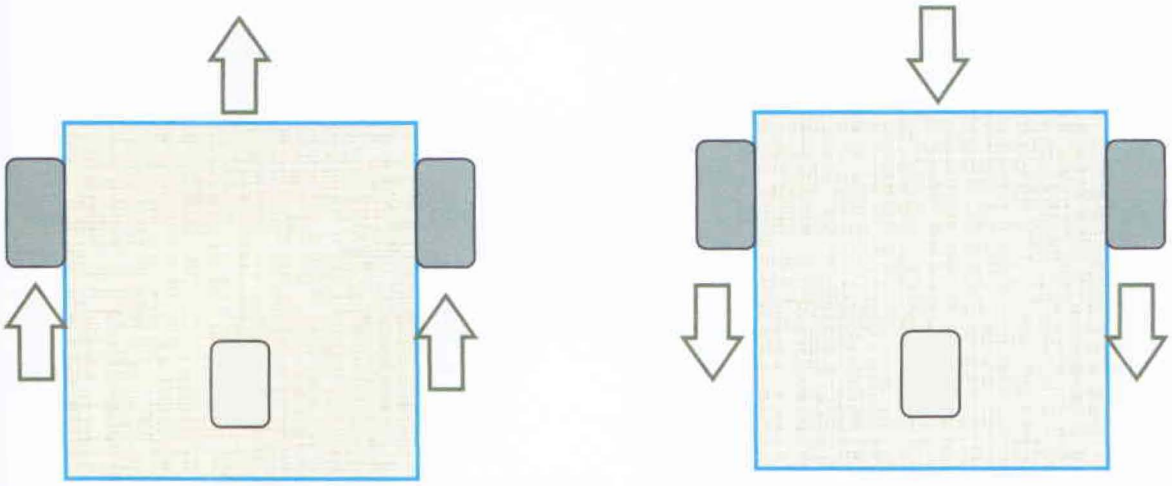


Figure 2-3: Movement criteria for forward and backward

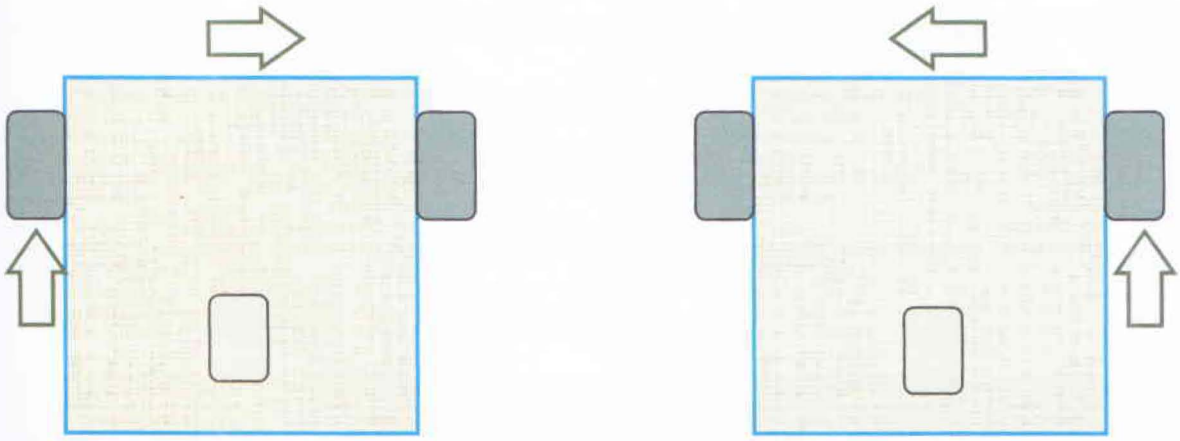


Figure 2-4: Movement criteria for towards left and right

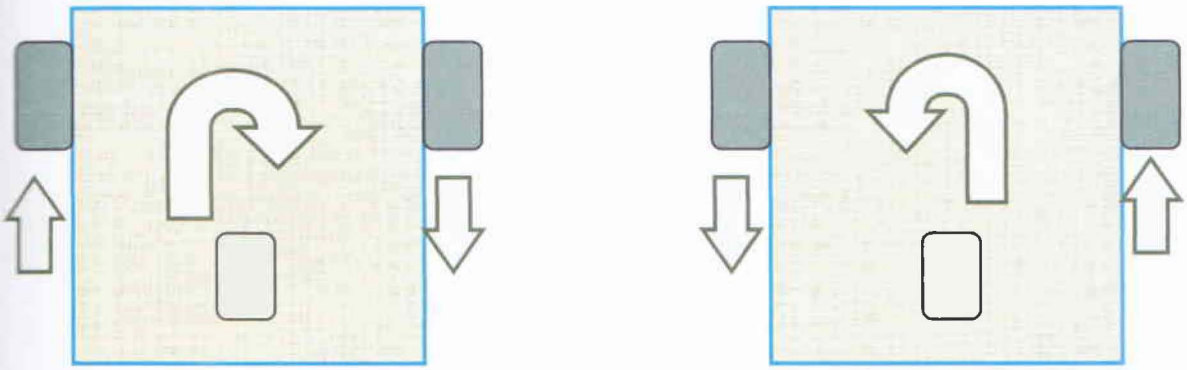
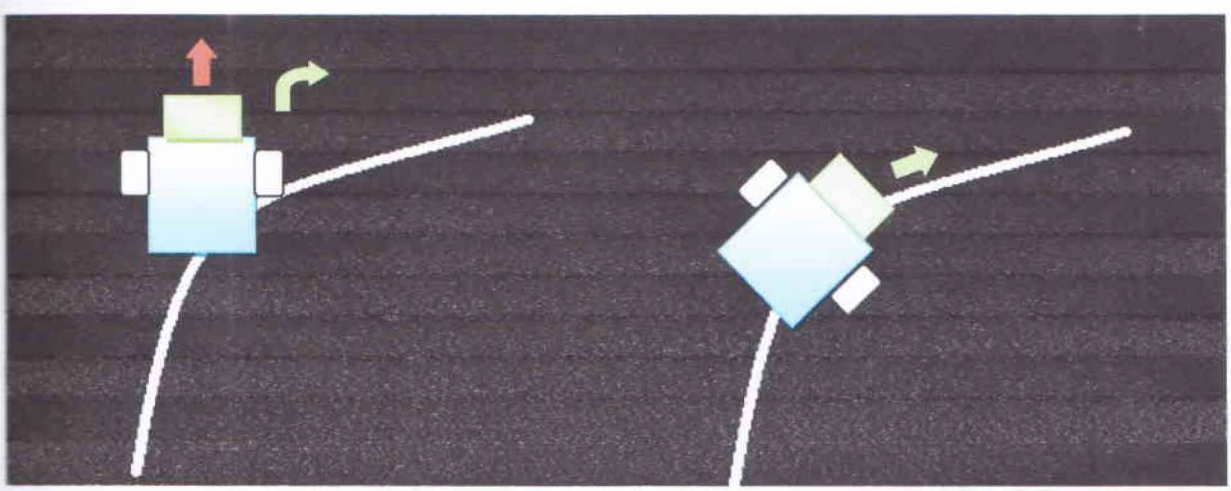


Figure 2-5: Movement criteria for rotational movement

2.4 Control from any undesired situation

The robot may face various kinds of undesired situations such as straying out of line, facing an obstacle in the path or a cross-path situation etc. To avoid those situations, the robot should be properly programmed. If the robot goes out of line, then it decides its next movement according to its last movement. Suppose the robot is turning left and while it goes out of line. Then it will decide to go left till it can't find any path again. In case of right movement and goes out of line the robot will turn right. To reduce complexity we included only the feature that if the robot goes out of line, it can come back to its original path.



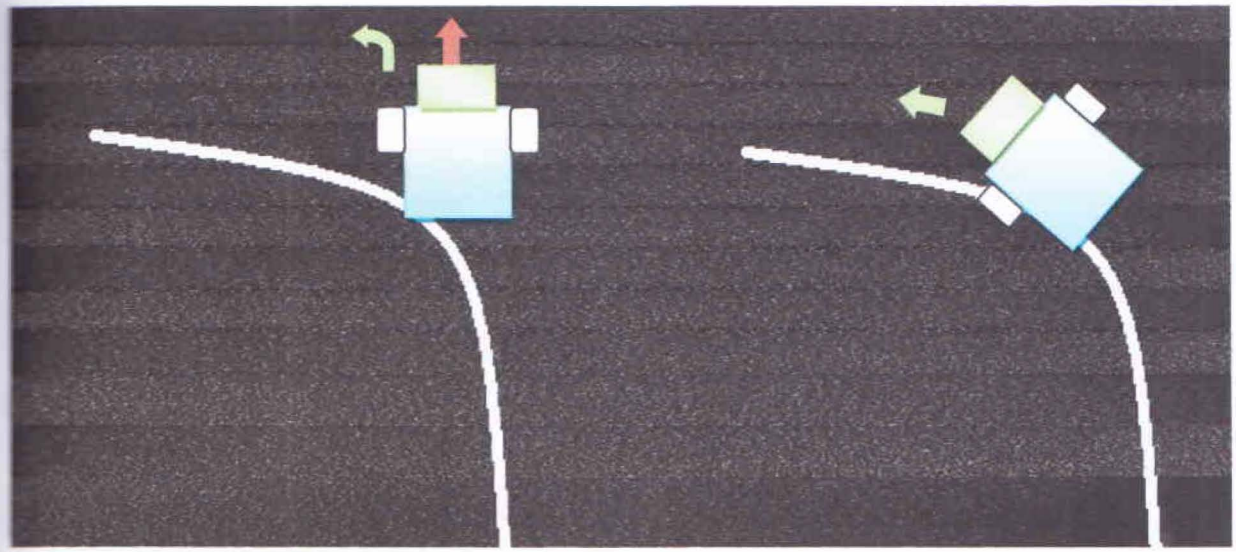


Figure 2-6: Process to find the path when the robot goes out of line

2.5 Temperature Sensing

The robot senses the temperature with the LM35 temperature sensor. It is a low cost and precise temperature sensor. First we set a reference temperature (voltage) externally using a variable resistor (pot) and give it as an input to the microcontroller. When the robot senses the current temperature from LM35 that is more than the reference temperature, the robot will stop and show an error sign. To run the robot again, we have to press reset button to hardware reset the microcontroller. The robot is programmed so that it senses the temperature every 30 seconds.

Programming flow chart for microcontroller:

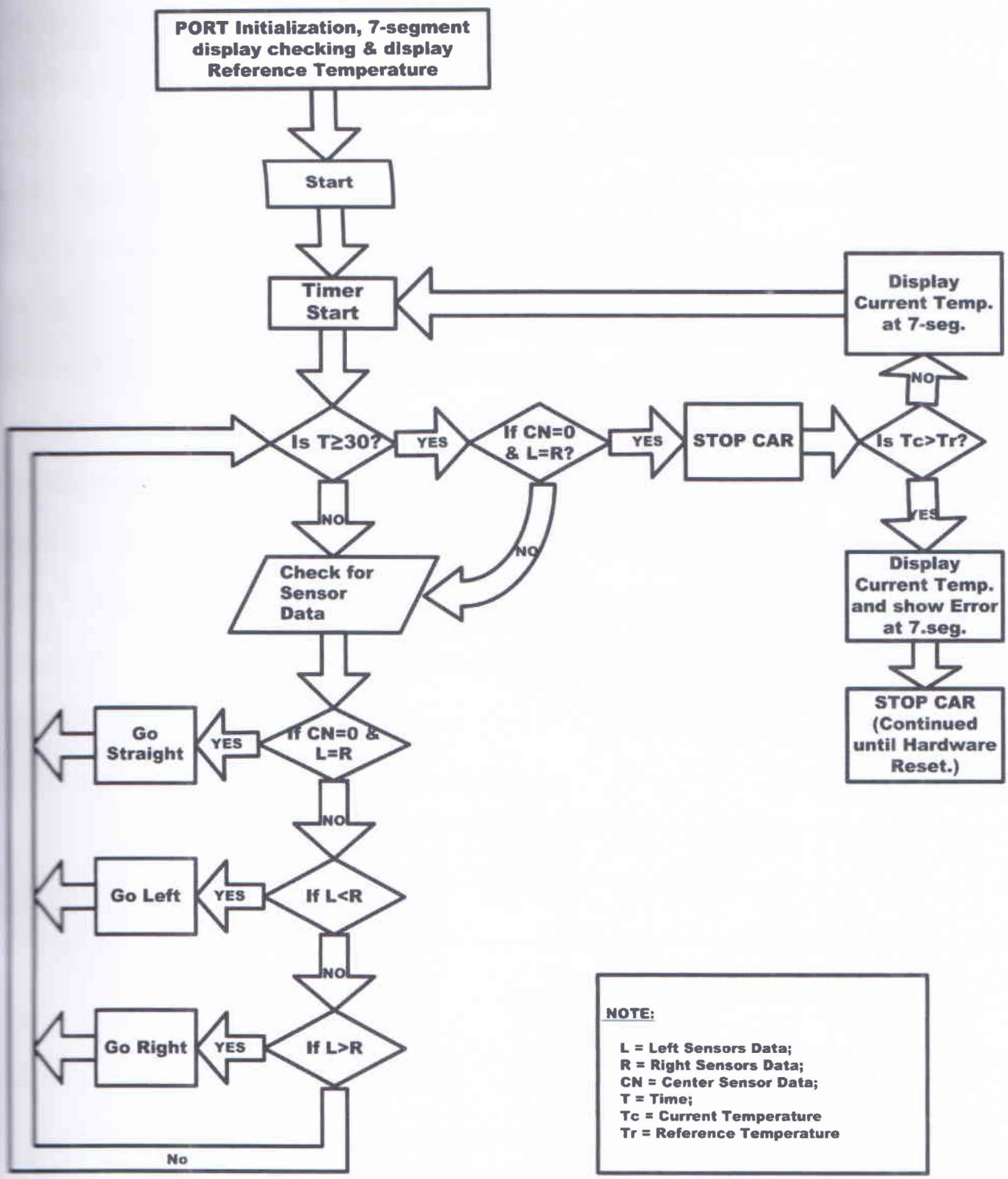


Figure 2-7: Flow chart for programming the microcontroller

2.6.1 Description of Program

The flowchart in figure 2.7 shows the whole operation of the robot. At first the microcontroller initialize the ports and then check the 7-segment display (whether all digit of the display working properly or not). Next the microcontroller checks the reference temperature pin for a value, performs an analog to digital conversion, stores the data and shows the result at the 7-segment display. At the same time, the timer of the microcontroller starts running and it is set so that if the time reaches 30s, there will be an interrupt. When the time is less then 30s, the microcontroller checks the sensor pins, process the value according to center sensor, left sensors and right sensors. Then it saves the values temporarily in the variables L, R and CN. To extract the left sensor values from the sensor array, we multiply the sensor array value with hex value 70h (01110000b) and right shift it 4 times and count the final value as L. To get the right sensors value from sensor array, we multiply the sensor array value with 07h (00000111b) and count this value as R. At last, to get the value of center sensor we multiply the sensor array value with 08h (00001000b) and count the result as CN.

For example lets the sensors read the value like below,

L3	L2	L1	CN	R1	R2	R3
0	0	0	1	0	0	0

Here, sensor array value = 00001000b

So, CN = 00001000b × 00001000b = 10000b

L = (00001000b × 01110000b) × Right shift 4x = 00000000b

Similarly, R = 00000000b.

If the sensor read like below,

L3	L2	L1	CN	R1	R2	R3
1	1	0	0	0	0	0

Here, sensor array value = 01100000b

So, CN = 01100000b × 00001000b = 00000000b

L = (01100000b × 01110000b)× Right shift 4x = 00000110b

Similarly, R = 00000000b.

If the sensor read like below,

L3	L2	L1	CN	R1	R2	R3
0	0	0	0	0	1	0

Here, sensor array value = 00000010b

So, CN = 00000010b × 00001000b = 00000000b

L = (00000010b × 01110000b)× Right shift 4x = 00000000b

And R = (00000010b × 00000111b)=00000010b.

Next the microcontroller compare the values of CN, L and R, and the robot goes straight, turns left and right according to our logic. The sensor is checked continuously and the robot moves according to it until the timer reaches 30s. When the timer reaches 30s, the microcontroller first checks whether the robot is going on a straight path at that time or not. If the robot was not going straight, it moves until it detects the straight path and the display shows a waiting sign. When it confirms that it is on straight path, the robot stops moving and the microcontroller checks the

current temperature pin, performs analog to digital conversion, saves the value in a variable and displays the current temperature in the 7-segment. At the same time, it compares the reference temperature with the current temperature. If the current temperature is greater than the reference temperature, the microcontroller show an error sign in the 7-segment display and blinks a red light to notify that there is a temperature difference in that area and the robot will stop until the reset button is pressed. If the current temperature is less then reference current temperature the robot will start the whole process again from the starting position (i.e. it will keep moving over the path and repeat the same procedure).

The complete program code is given in Appendix A.

Chapter 3 : Major Components of a Line Following Robot

3.1 Sensors

Sensors are important for automatic operation. For a robot, the sensing mechanism is very important because it has to make decision for any particular situation. By this mechanism the robot can perform different operations by itself. In our project, the sensors are used to:

1. Detect its path
2. Sense the temperature

To detect the path, we used Light Dependent Resistor (LDR) sensor and to sense temperature, we used the LM35 temperature sensor.

3.1.1 Light Dependent Resistor (LDR)

LDR is a kind of resistor. The main feature of this resistor is that the resistivity depends on the intensity of light incident on it. In dark space the resistivity of this resistor is very high. When light falls on it, the resistivity of this resistor decreases significantly.

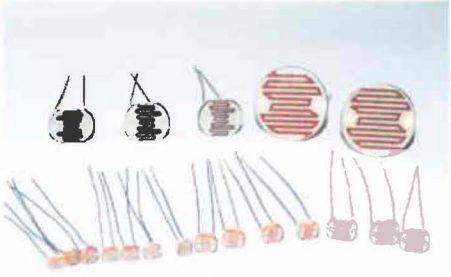


Figure 3-1: Light Dependent Resistor (LDR)[1]

3.1.1.1 LDR as a Sensor

In our robot, the LDR are used to detect the line which the robot will follow. Here we used the reflectivity of light from black and white surfaces. The white surface reflects light and the black surface absorbs it. We have used a white LED (Light Emitting Diode) and a LDR to make a single sensor. The LEDs remain turned on during the time of movement of robot. If there is white surface the light reflects to the corresponding LDRs and if there is black surface, the light is absorbed and cannot reflect to the LDRs.

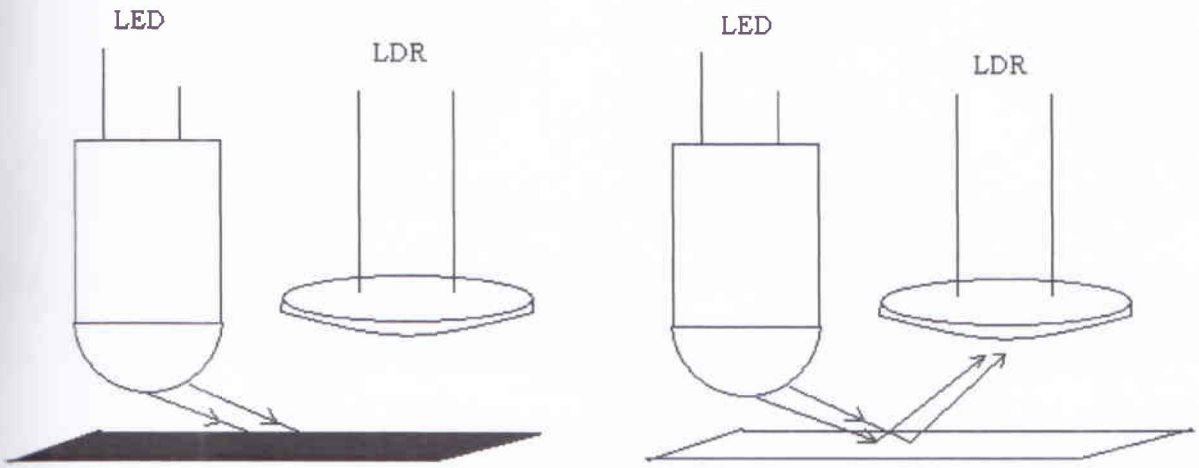


Figure 3-2: reflectivity of light from black and white surfaces.

The circuit diagram of a single sensor is given below. Here we used a fixed valued resistor in series with the LDRs. So the LDR behaves as a variable resistor. The resistivity of LDR now varies with black or white surface and hence the voltage level of output of the LDR changes. We send the output voltage of the LDR to the input of a comparator IC (LM324) and compare the voltage with a reference voltage

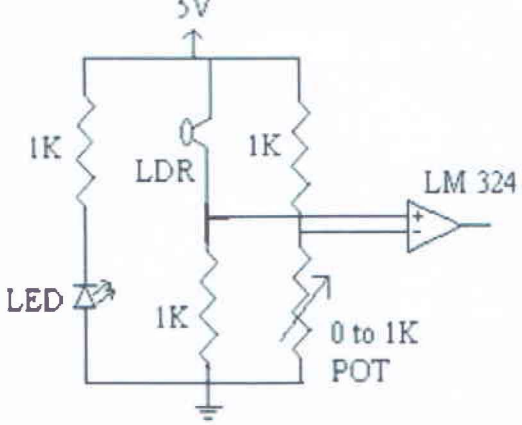


Figure 3-3: Complete circuit diagram for a single sensor

3.1.1.2 Arrangement of sensors

In our robot we have used seven LDR sensors and one LM35 temperature sensor. Arrangement of the sensors is also a very important issue. Basically, we can arrange the sensors in different ways. We can arrange them in straight line, in 'U' shape, 'V' shape etc. We have chosen the inverse 'V' shape for our robot because it has an advantage for turning in any direction.

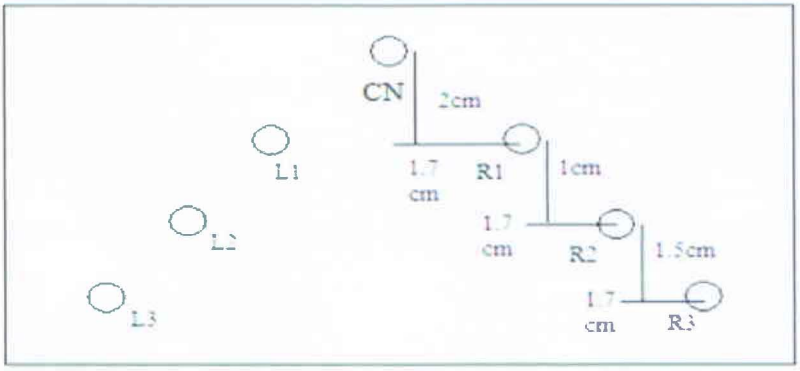


Figure 3-4: Arrangement of LDR sensors

The above figure shows the arrangement of sensors of our robot. Here sensor 'CN' always remains on the line. Sensors R1, R2 and R3 are to detect the path to the right and L1, L2 and L3 are to detect the path to the left.

3.1.2 LM324 Comparator IC

We have used LM324 as comparator. Each LM324 chip has four comparators. So for seven LDR sensors we used two LM324 chip.

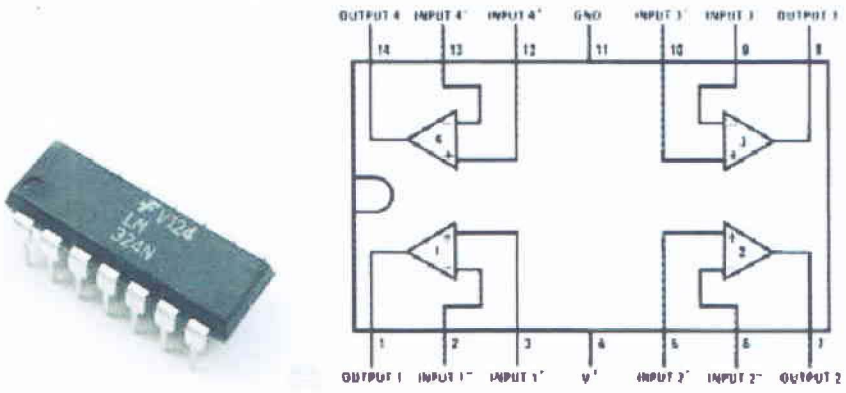


Figure 3-5: LM324 Comparator IC[2] [3]

The reference voltage is connected to the pins 2, 6, 9 and 13 and the output voltages from LDRs are collected to the pins 3, 5, 10 and 12 of LM324 comparator IC. A reference voltage that is dependent on light intensity of the room is adjusted. For normal condition this value is around 0.8V. Changing this value will increase or decrease the sensitivity of this sensing circuit. If the LDR output voltage is greater than the reference voltage the output of the comparator is logic high. Otherwise, if the output voltage of the LDR is less than the reference voltage the output of the comparator is logic low. Thus we can get a digital output from the sensors.

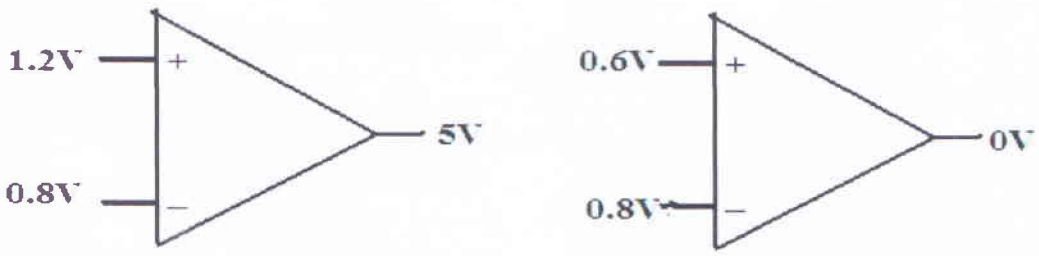


Figure 3-6: Example of comparing voltage

3.1.3 Inverter

The inverter IC 7404 has used to invert the logic from sensors of line following. We have used it because the microcontroller takes input as logic low. But we get the digital signal as logic high for the white surface. The inverter IC inverts the logic and produce proper logical conditions for microcontroller. Here, we have to say that the robot is specially designed to follow a white line in black surface. If we wanted the opposite (i.e. black line in white surface), then we would not need any inverters.

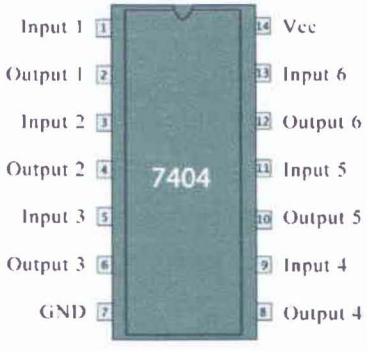


Figure 3-7: Inverter IC 7404 [4]

3.1.4 Temperature sensor (LM35)

LM35 is a low cost and precision temperature sensor. It senses the temperature and gives the output as voltage. The output voltage of this sensor is linearly proportional to the temperature. For each 1^0 C temperature change the output voltage changes by 10mV. It has following features [5]:

1. Calibrated directly in ° (Degree) Celsius
2. Linear scale factor + 10.0 mV /°C
3. 0.5°C accuracy guaranteed (at +25°C)
4. Rated for full -55° to +150°C range
5. Suitable for remote applications

6. Low cost
7. Operates from 4 to 30 volts
8. Less than 60 μA current drain
9. Low self-heating, 0.08°C in still air
10. Nonlinearity only ±¼°C typical

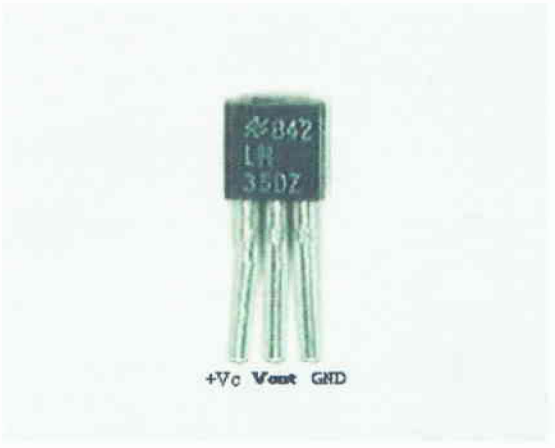


Figure 3-8: LM35 temperature sensor [6]

In our robot we've used this temperature sensor to detect the temperature on its path. The robot will detect the temperature of its surroundings and that will displayed by a seven segment display.

3.2 Mechanical Components

Robot is an electromechanical device. So at the very beginning of making a robot we have to consider two major issues, which are 'Mechanical' and 'Electrical'. The mechanical part is more related to the 'size', 'weight', 'inertia' etc. We have to consider a combination of all parts such that the robot is flexible to move in any direction. It should also have low weight and inertia.

In our robot, we chose a rectangular shaped base. The base is made by 3mm. thick board. We have arranged three wheels such that it can move easily.

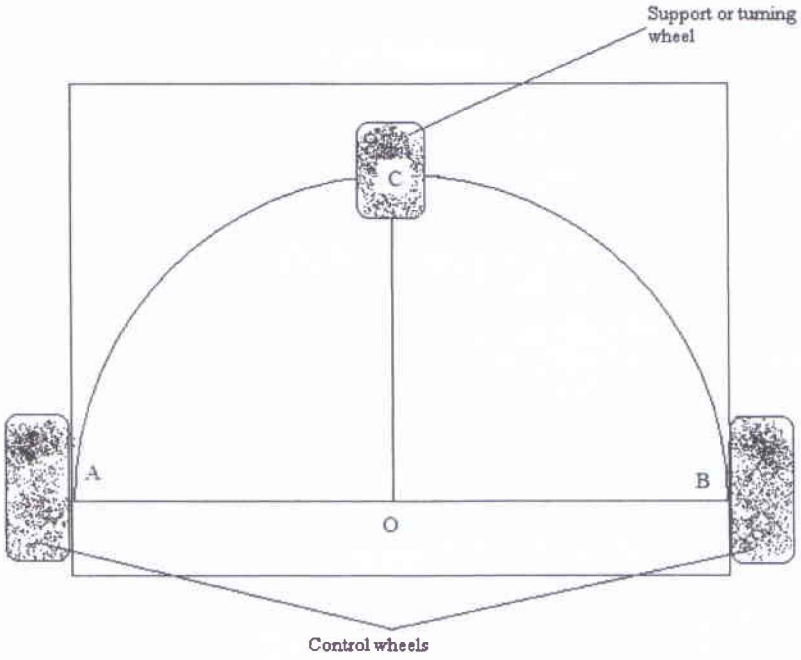


Figure 3-9: Arrangement of wheels

3.2.1 Wheels

The robot has three wheels. Two wheels are for controlling the movement and third one for balancing. The above figure describes the arrangement of the three wheels. Here, A and B are the positions of the two controlling wheels. We connect the points A and B and consider it as a diameter of a circle. Here 'O' is the middle point of the line AB and also the center of the circle. Then we drew a half circle which is centered at 'O'. After that we drew a perpendicular line on AB at point 'O'. The line crosses the circle at point 'C'. So finally, 'C' is the position of third wheel. If we position the third wheel by this way, the robot can move easily at any direction. It's because the third wheel is on the semicircle provided by the diameter AB so both controlling wheels can give full efforts to the third wheel to turn the robot in any direction.

3.2.2 Motor

To design a robot it is very important to choose the right motor. It is because; the motor will have to take all the load of the robot. We are discussing about 'Motor' in 'Mechanical' part because the choice of motor is more related to mechanical issues of the robot like torque, inertia etc. We have to choose such a motor that will be light, has a high torque and is able to be controlled by a certain range of voltage level.

In most of the cases a normal DC motor cannot produce enough torque to carry the load of the robot, so we need a gear system. We can use a gear box separately with the motor or we can use gear motors which has built in gear box. For our robot, we use a 12V gear motor. The main features of this motor are:

1. The motor is light weight
2. It can be controlled in the range of 6 to 12 volts.
3. It produces high torque as it has gear mechanism
4. It can take heavy load
5. As it is a DC motor, it is easy to control.

In our robot we controlled two wheels by two separate motors. So, the degree of freedom of the robot is two.

3.2.2.1 Advantage of gear motor

Gear motor is a special kind of DC motor which has a built in gear system. It has better torque characteristics than the normal DC motor. For DC motor, torque (τ) is proportional to current (I) and the rotational speed (ω) is proportional to the voltage (V).

$$\tau \propto I \quad \text{and} \quad \omega \propto V$$

Power (P) of the DC motor is equal to product of voltage and current.

$$P = VI$$

As the power supplied to the motor is constant, if the current increases then the voltage decreases or vice versa. In our robot we experienced that if we use normal DC motor, it needs almost 1A current to develop the torque needed to overcome its initial inertia. At the same time as the power is constant the voltage decreases a lot. So the rotational speed reduces. Thus the robot moves very slowly which is undesired for our project. In this condition as the motor draws a large amount of current, it quickly drained out the power source (i.e. battery).

In gear-motor the gear mechanism helps to develop a high torque with less current. So if we use gear-motor instead of normal DC motor, that will consume less current to overcome torque. At the same time it reserves enough voltage for rotational movement for a constant power supply.

In out project we have used 12V gear motor which is the best choice for normal movement of our robot.

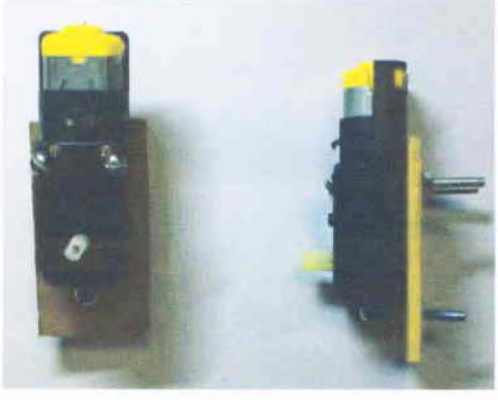


Figure 3-10: 12V gear motor

3.3 Motor driver IC (L293D)

To drive the motor properly, we need a motor driver IC. The microcontroller cannot deliver currents high enough to drive the motors. L293D is a motor driver IC which can deliver enough current to control two motors separately.

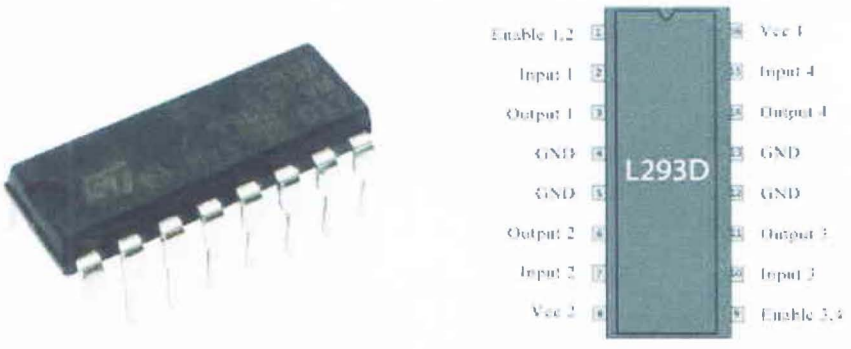


Figure 3-11: L293D motor driver IC [7] [8]

The above figure shows the pin configuration of L293D. It has two channels to control two motors separately. These are [9]

1. Channel 1: Pin 1 – Pin 8
2. Channel 2: Pin 9 – Pin 16

The L293D motor driver IC has following features:

1. Wide supply voltage range: 4.5V to 36V
2. Separate input logic supply
3. Thermal shutdown
4. Input current 600mA
5. Peak output current 1.2A per channel

L293D is characterized for operation from 0°C to 70°C. On the other hand, we need at least 500mA current for normal movement of the robot. The L293D can supply a maximum of 1.2A current to the motors from a single channel.

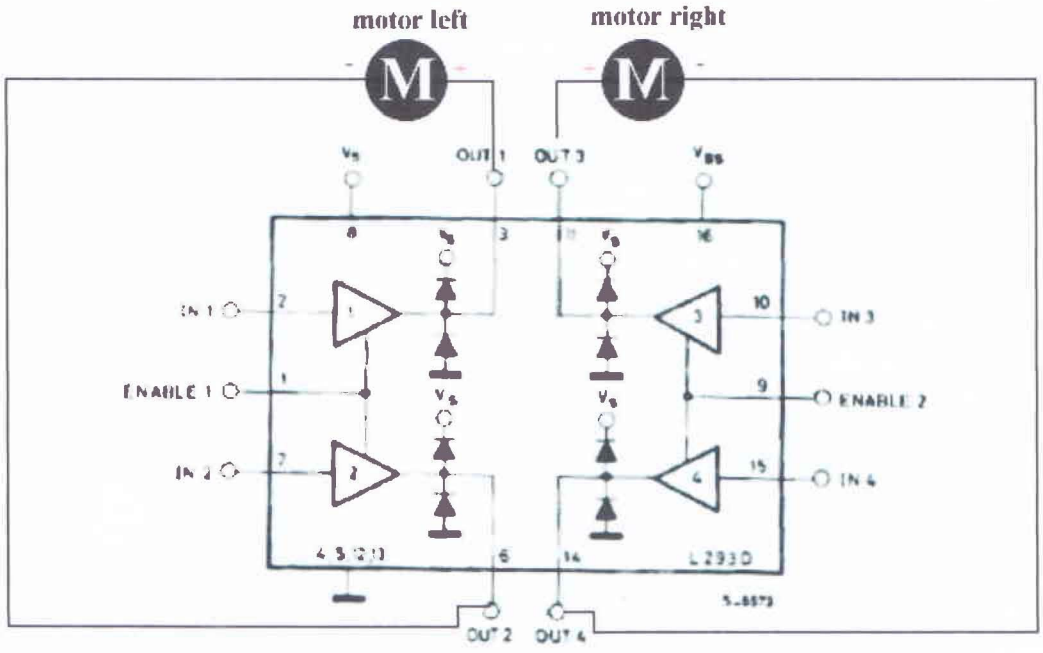


Figure 3-12: Block diagram of operation of L293D[10]

In our project, the motor driver IC was used at the output of the microcontroller. Thus, the two motors run according to the program stored in the microcontroller.

3.4 Seven segment display

To display the temperature, we have used seven segment displays. We have used a 4 digit common cathodes seven segment display which is suitable for programming and to arrange in circuit.

Figure 3-13: 4 digit common cathode 7- segment display [11]

3.5 Microcontroller

Microcontroller works as the brain for our robot. All the features of our robot are programmed inthe microcontroller. There are two families of microcontrollers available in the market. They are PIC series from Microchip Co and AVR series from Atmel Co. In our project we chose ATmega16A from the AVR series.

Figure 3-14: ATmega16A microcontroller [12] [13]

3.5.1 Why AVR

In our project, we chose AVR microcontrollers because they are faster than the PIC series microcontrollers. When PIC and AVR run at the same clock frequency, then AVR executes instructions four times faster than the PIC. This is because the PIC needs four clock cycles to execute an instruction while the AVR need one clock cycle [14]. Architecture of the AVR series microcontroller is also more developed than the PIC series microcontrollers. It supports C programming language and is also available in the local market.

3.5.2 Kinds of AVR

There are five kinds of AVR families available in the market. Three kinds of them are more popular. These are [15]

1. Tiny AVR (ATtiny series)
    a. 0.5 – 8kB program memory
    b. 6 – 32 pin package
    c. Limited peripheral set
2. Mega AVR (ATmega series)
    a. 4 – 256 kB program memory
    b. 28 – 100 pin package
    c. Extended instruction set
    d. Extensive peripheral set
3. Xmega (Atxmega series)
    a. 16 – 384 kB program memory
    b. 44 – 64 – 100 pin package
    c. Extended performance features, such as DMA, "Event system" and cryptography support
    d. Extensive peripheral set with DACs

In our project, we have used the mega series of the AVR family, ATmega16A. It is a fast microcontroller to execute the program so that the instructions can be tracked quickly.

3.6 Power supply

Power supply is an important issue for out project. For free movement of the robot, we used a battery as the power supply. Further, we experienced that the motors we used in our robot needs at least 0.5A current for normal movement. We have used a 6V 4.5Ah Led Acid battery as power supply for the robot.

The ICs we used in our project need 5V voltage for operation. We have used a voltage regulator IC (7805) which can limit the voltage at 5V. It can supply maximum load current of 500mA.

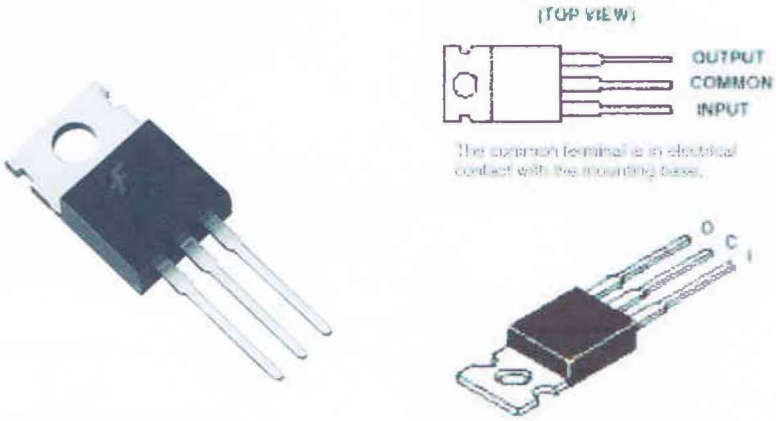


Figure 3-15: Voltage regulator (7805)[16]

The complete Simulation Circuit diagram is given at Appendix B.

Chapter 4 Construction

4.1 Introduction

It is always very interesting to implement any theoretical solution practically. Making a robot is interesting but a bit difficult too. The main problem we faced to make the robot is to collect the equipment. As robot is an electromechanical device, we needed to collect components from both electrical and hardware stores. After collecting all the components it is needed to assemble them. It is also very important to assemble all the parts properly. We did many trial and error corrections too.

4.2 List of Equipment

In our Line Following Robot we used the following components:

1. ATmega16L (micro controller)
2. LM324 (Comparator IC)
3. L293D (DC motor controller)
4. Inverter (7404)
5. Voltage regulator (7805)
6. LM35 temperature sensor
7. Potentiometers (1KΩ, 50Ω)
8. Resistors (1KΩ)
9. Capacitors (1nF)
10. 4 digit common cathode7- segment display
11. Push button
12. On/Off button

13. Light Dependent Resistor (LDR)
14. White and red Light Emitting Diodes (LED)
15. Hard board (3mm thick)
16. Vero board
17. DC gear motors
18. Controlling wheel
19. Rotating wheel
20. 6V – 4.5Ah rechargeable battery
21. Wires and connectors

4.3 Procedure

4.3.1 Step 1: Making the base

In our project we have made the base of the robot with a 3mm thick hard board. It is rectangular in size. To attach the motors with the board, first we attached small pieces (motor size) of the hard board with the main board. The pieces are attached with glue.

Figure 4-1: Small pieces of hard board attached with main board

4.3.2 Step 2: Attaching the motor

Next we have to attach the motors with the base. To attach the motors with the base, first we attached the motors to a similarly sized piece of hard board and then we attached it with the main board just beside the board pieces that we attached before. We attached the motors with the hard board pieces using screws.

Figure 4-2: Motor attached with small pitches of hard board by screw

4.3.3 Step 3: Attaching wheels

After attaching the motors with the main board, we attached the wheels with the motors. To attach wheels we used two small pieces of old pens. First we emptied the ink tube of the pen. Then we cut down the nib and cleared the tube using water. After that we cut the tube in small pieces according to our need. We attached one side of the tube with motor and the other side with the wheel. We collected the wheels from a toy car.

Figure 4-3: Wheels from toy car and supporting wheels

Figure 4-4: Attached wheel to the motor

4.3.4 Step 4: Placing the third wheel and connecting the wires to the motors

Next we placed the third wheel which is a rotating wheel. This wheel is responsible for balancing the robot and also helps to turn it in any direction. The process of placing the wheel has been described in chapter 3, section 3.2.1. After that we connect wires to the motors, the lower side of the robot is complete.

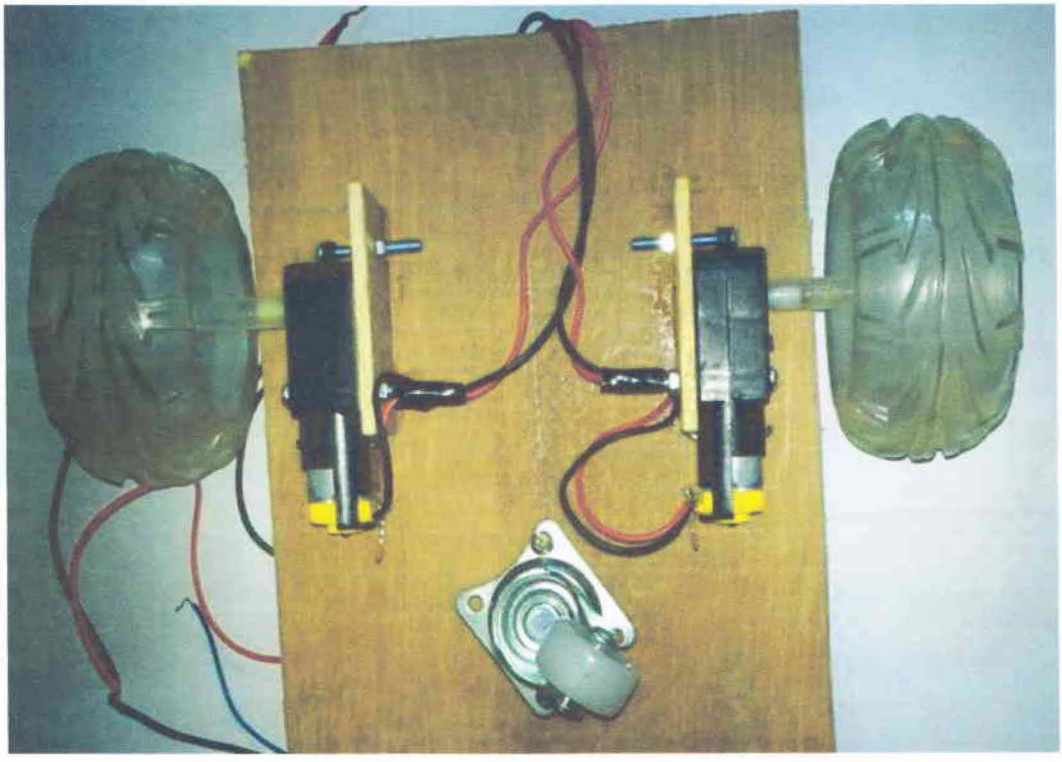


Figure 4-5: Complete Lower side of the robot

4.3.5 Step 5: Making the sensor array to detect path

So far we worked with the mechanical part. Now we start working on the electrical part. We made the sensor array to detect the line by Light Emitting Diode (LED) and Light Dependent Resistor (LDR). The circuit diagram and process of construction are discussed in chapter 3, section 3.1.6.

To make the sensor array, first we constructed the circuit in a bread board for simulation. After the simulation was over, we constructed the circuit in Vero board.

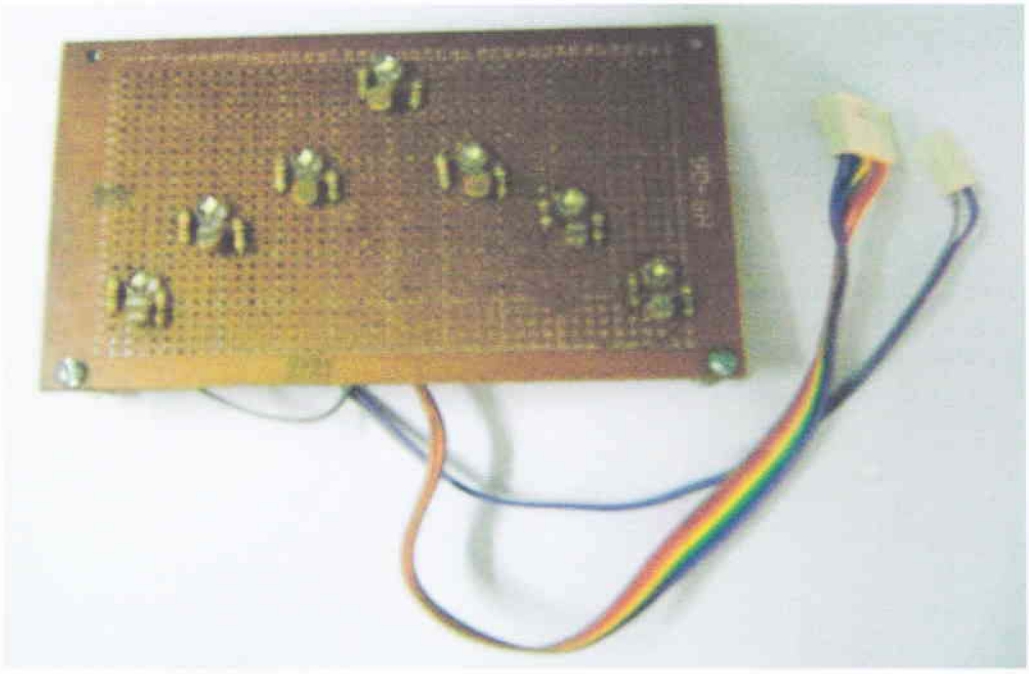


Figure 4-6: Sensor array to detect path (front view)

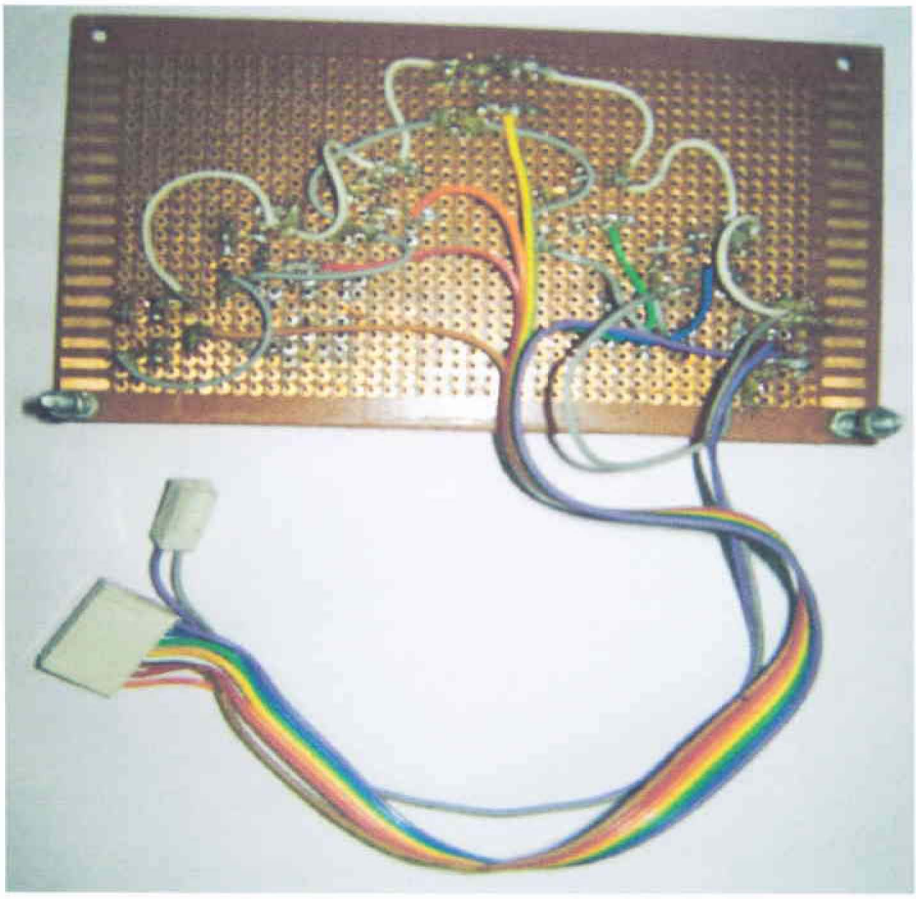


Figure 4-7 Sensor array to detect path (back view)

4.3.6 Step 6: Construction of the main circuit

After making the sensor array to detect the path, we constructed the main circuit in Vero board.

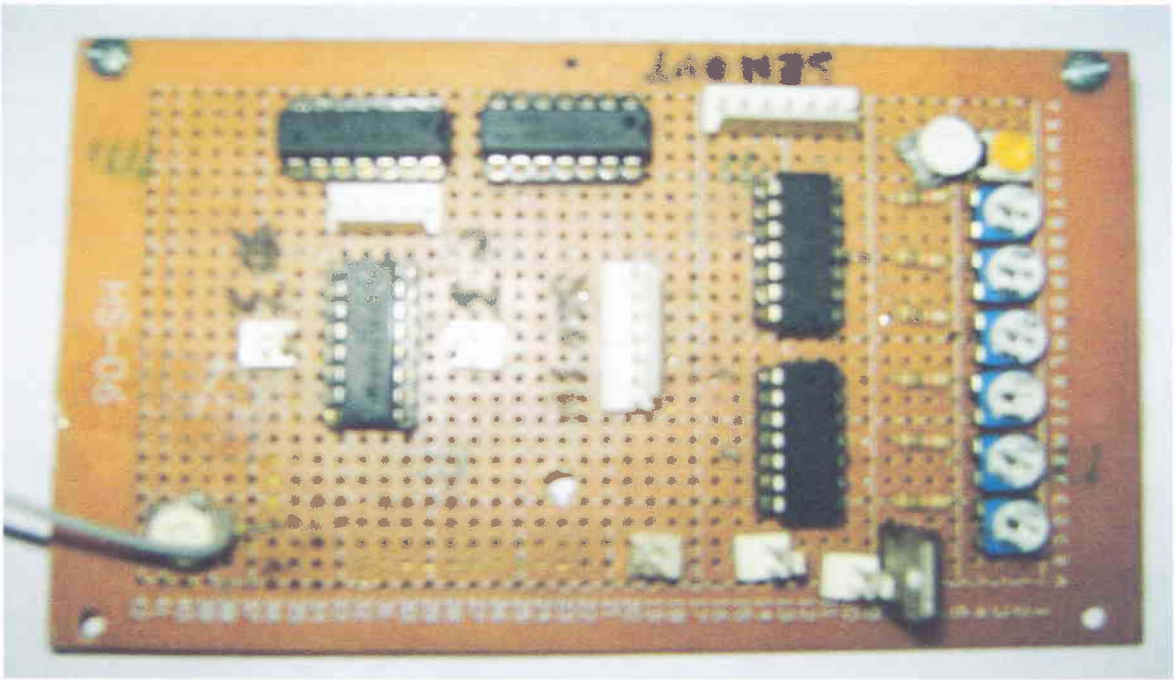


Figure 4-8: Main circuit containing potentiometer, LM324, L239D and 7805

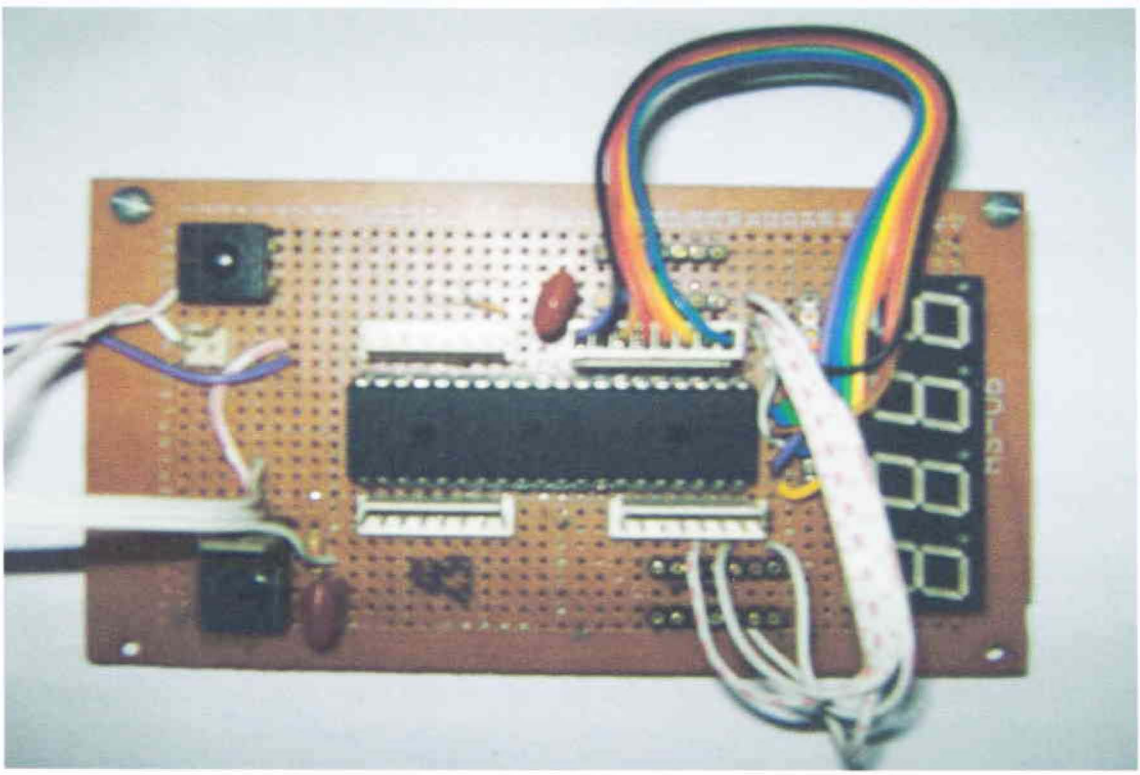


Figure 4-9 Main circuit containing microcontroller, switches and display

4.3.7 Step 7: Placing the circuit on the robot's base

After constructing the circuit in Vero board as the simulation circuit diagram given in Appendix B, we fixed it to the robot's basewith screws. We placed the sensor array a bit lower than the baseso that it can sense the path accurately.

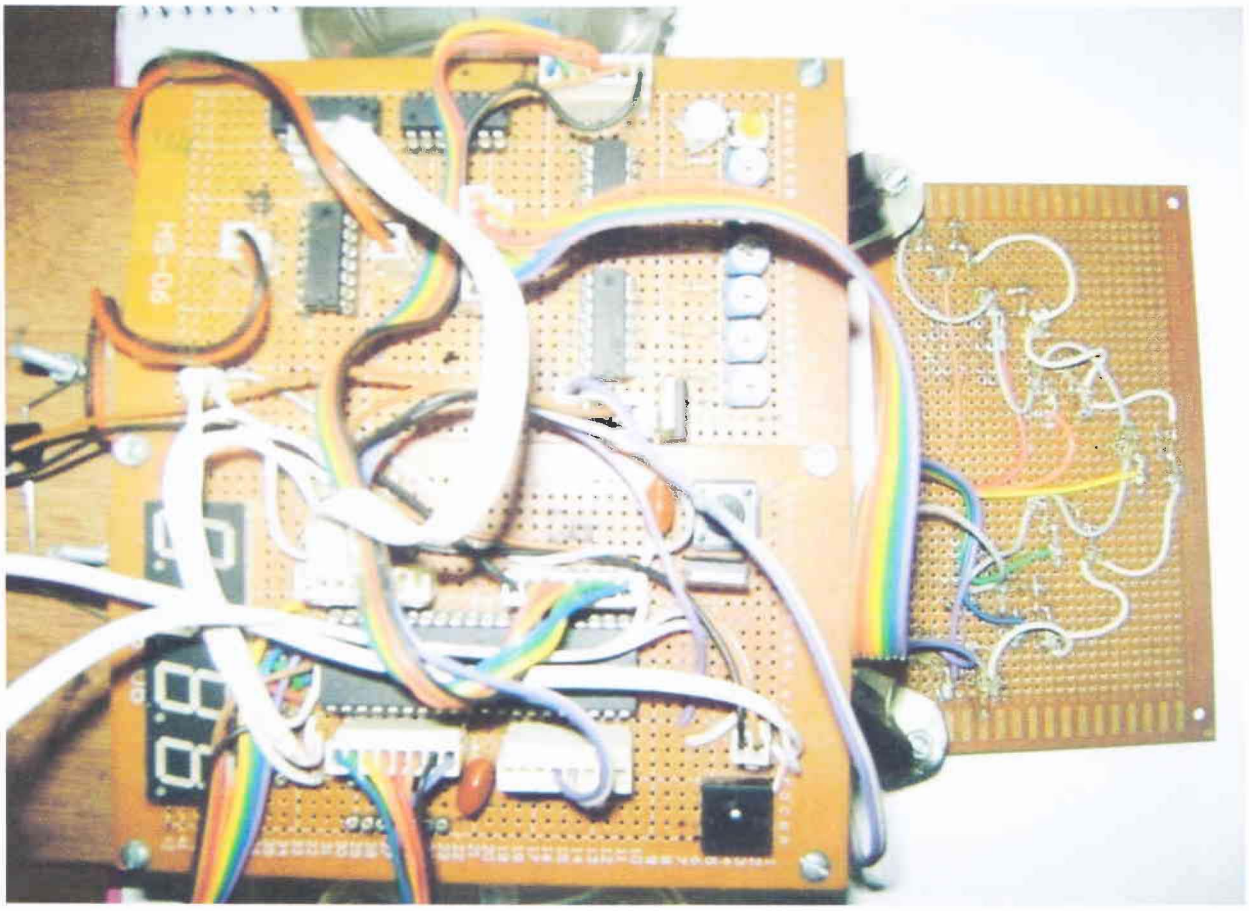


Figure 4-10: Complete circuit board

4.3.8 Step 8: Making a cover for the robot

After placing the circuit, we made a cover for the robot to make the outlook better. We supplied the power of the robot by separate battery. Thus the entire robot is complete.

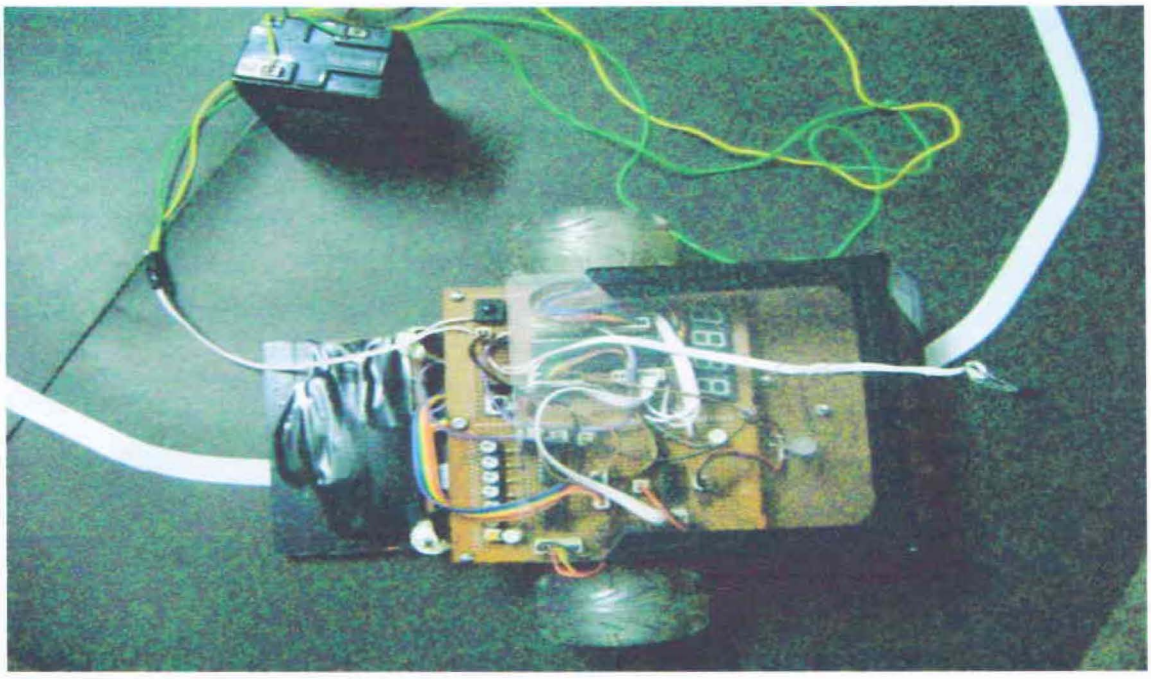


Figure 4-11 Complete robot (top view)

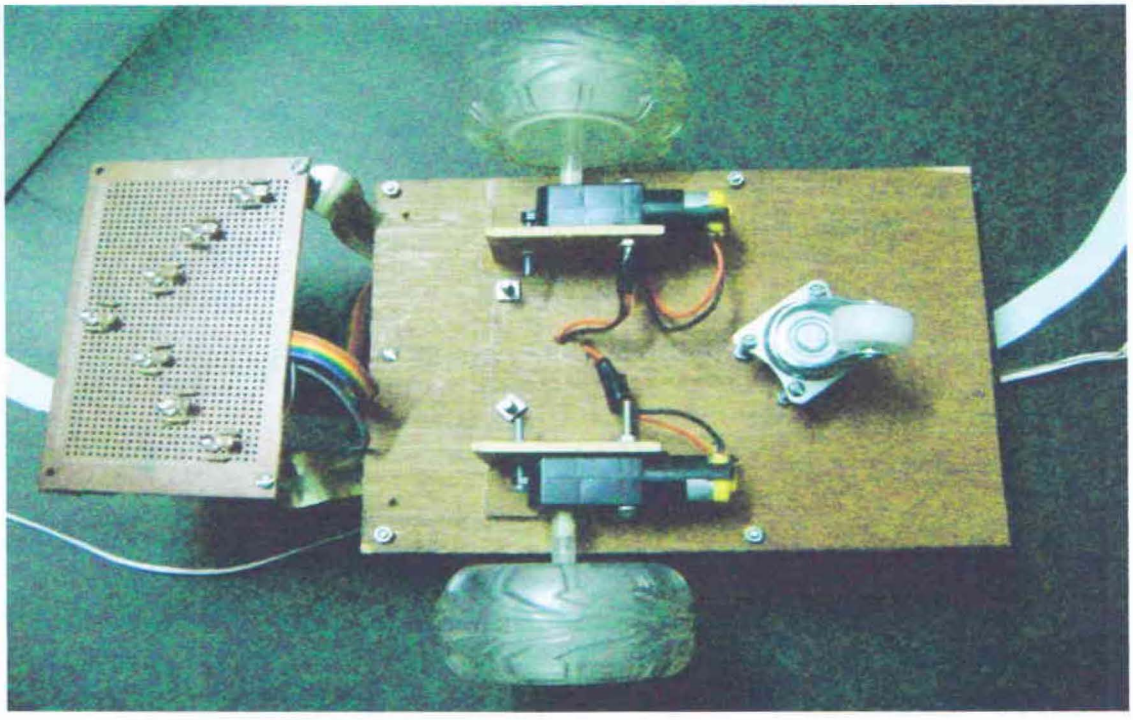


Figure 4-12 Complete robot (bottom view)

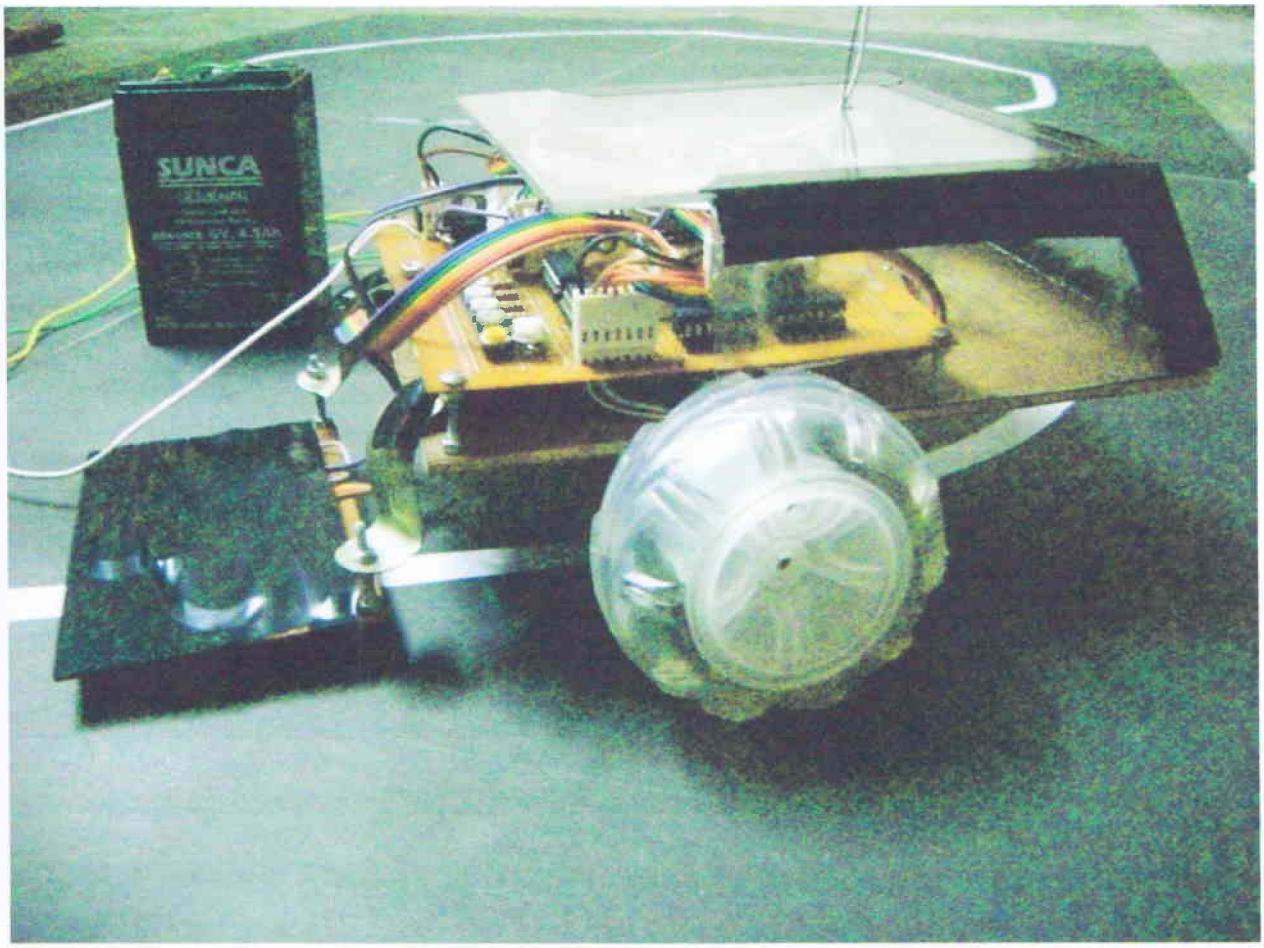


Figure 4-13: Complete robot

Chapter 5 : Discussion

Line following robot plays a very important role in robotic technology based systems. There are some robots which need not move from one place to another. Rather, it can perform its job by some rotational movements standing in a particular place. But when the concept to move robots from one place to another was developed, the question of what path the robot will follow arose. Here, the line following robot can be an answer of this question.

There are some other ways to move robots from one place to another. Those ways may be done by image processing or using the principle of sound reflections. In these cases the robots have to be programmed to use particular images to take decisions to run, or in the echo system, it has to be programmed for a specific reflection frequency range of sound. But both the cases are very difficult to implement. However, different techniques can be followed to run a robot in a desired direction. These techniques can be applied to different situations.

In this project we tried to make a line following robot which can follow a white path on a black surface and also sense the temperature on its path. The aim of our project was to introduce a different kind of concept to detect the temperature of a room and also issue a warning if there is any mismatch of room temperature with the desired one. The idea was developed because of the some practical problems faced by many cold storages companies around the world. In a cold storage usually there are a number of cooling machines around a large room. As the room is very large and the spaces between the selves are very narrow, it is not possible to get information if the temperature in all the parts of the room is the same or not. It is very important because, if any cooling system becomes out of services then the foods near that system will become rotten. If we

use permanent sensors on those areas, the sensors will only sense temperature of the particular area and unable to sense the temperature of those areas where the sensors are not present. But if we can monitor the temperature of the room using a line following robot, then it will be very easy to sense the temperature of any area of that large room and it will be also cost effective.

5.1: Problems we faced

During construction of the robot we faced some problems. These are,

1. We suffered due to the unavailability of proper motors for a long time during the construction of the robot. First, we used 9V DC motors without gears. The motors were heavy and their torque was low. So it couldn't take the load of the robot. To solve this problem, we used 12V DC gear motors. The advantage of these motors are that they are DC motors with gear systems which is easy to control and the gear mechanism also helps to produce high torque, so the motors can run with sufficient high load. It also draws less current than the motors without gears.
2. Attaching wheels to the motors were also a challenge. Here we used a ball point pen as the wheel axis.
3. To detect the path, first we tried to use IR (infra-red) sensors instead of LDRs, but the performances of IR sensors were not satisfactory. Further, we used an IR sensor to detect any obstacles in the path. We used six LDR sensors to detect the path and one IR sensor to detect obstacles. But later, we needed seven sensors to detect the path. So the feature to detect obstacles in its path was canceled.

5.2: Suggestions for Future Development

Though we met the necessary demand to complete our project, there are some limitations of our robot. We hope that further development of it can make this robot more professional and user friendly. Here we are suggesting some features which can be make the robot more user- friendly.

1. The battery (i.e. power source) of the robot is not attached to it. Since the motor (and other components) draw a large amount of current (0.5mA), it is not possible for a 9V dry cell battery to supply this current. So we used a 6V 4.5Ah sealed lead acid rechargeable battery. But the battery is so heavy that the robot cannot carry it without damaging the robot's structure. If we use steel boards for base instead of hard board for the robot the problem can be solved.
2. The robot shows the temperature on a display mounted on the circuit board. So it is not so user friendly. If we can make a wireless remote communication system then it could be possible to monitor the temperature from remote locations.
3. The robot cannot detect any obstacles in front of it. If we can use an IR sensor to detect obstacles, and program what should be done in that situation then it could be better.
4. Another problem of our robot is that it is cannot follow complex paths, such as cross paths or sharp turn. This problem can be fixed by adding more sensors or by using advance sensors and modifying the code of the microcontroller

Though there are some limitations in our robot, we hope further work on it can fix these problems and will make it more users friendly. More features may also be added to this robot.

Chapter 6 : Conclusion

Line following is very important for autonomous operation of robot. By this technique we can send robots to a particular destination in a defined way. There are many applications of line following robot. They can be used as waiter in a restaurant to serve food to the customers from kitchen. In a library we can use this to bring books from selves. In office and house we can use this robot to clean the floor etc.

In our project we have used the line following robot to sense temperature in temperature sensitive areas. A reference temperature is set to the robot for comparison of current temperature. If it finds the current temperature more than the reference one it stops on there and warn by a blinking light.

Our project combines two systems, i.e. temperature sensing technique and line following robot, to autonomously monitor the local temperature in the robot's path. This robot can be very useful in temperature sensitive areas like cold storages. Further works on it can develop its features and make it more efficient.

References

1. http://www.made-in-china.com/showroom/costargroup/product-detailXouxBFYvhGpA/China-Cds-Photoconductive-Cell-Photoresistor-LDR.html
2. http://www.solarbotics.com/products/lm324/
3. http://ecee.colorado.edu/~ecen4618/lm324.htm
4. http://www.engineersgarage.com/electronic-components/74ls04-datasheet
5. http://www.national.com/mpf/LM/LM35.html
6. http://www.facstaff.bucknell.edu/mastascu/elessonshtml/Sensors/TempLM35.html
7. http://www.solarbotics.com/products/l293d/
8. http://www.engineersgarage.com/electronic-components/l293d-motor-driver-ic
9. http://robokits.co.in/datasheets/l293d.pdf
10. http://www.linuxfocus.org/English/July2003/article297.shtml
11. http://www.siliconray.com/components/discrete-component/leds/4-digit-7-segment-common-cathode-leds-0-56-inch.html
12. http://www.preisvergleich.eu/suche211981.html
13. http://www.datasheetdir.com/ATMEGA16A+AVR-microcontrollers
14. http://extremeelectronics.co.in/avr-tutorials/part-ii-getting-started/
15. http://en.wikipedia.org/wiki/Atmel_AVR
16. http://tom-itx.dyndns.org:81/~webpage/how_to/atmega168/mega168_howto_main_index.php

Appendix A

Program code

```
/*
 * linefollow.c
 *    Final Version
 */

#include<avr/io.h>
#include<avr/interrupt.h>
#defineF_CPU1000000UL

#include<util/delay.h>
/*---------Variable Defining---------*/
      int   sens_dat,
            sens_Lft,
            sens_Rgt,
            sens_Cntr,
            val[3],
            Go_left,
            Go_right;

Volatileint tmp,tmp1;

double      Res;//????????

/*--------------PORT Initialization ----------------------------*/

Voidport_init(){
      // --------I/P------------------//
      DDRA=0x00;//0xff;
      PORTA=0x00;// pull up enable
      DDRB=0x80;// PORTB is input EXCEPT PB7
      PORTB=0x7f;// pull up enable

      // --------O/P------------------//
      DDRC=0xFF;// 7 Seg
      DDRD=0xFF;// 7 Seg digit select and L298 motor drive
      PORTC=0x00;// for common anode display
      PORTD=0x00;// motor stationary, 7 seg off
}

/*----------Microcontroller Initialization------------*/
Voiduc_init(){
      SFIOR=0x00;//Special Function I/O Register disabled.
}
/*----------Analog to Digital conversion Initialization---------*/
Voidadc_init(){
      ADCSRA=0b10000011;// ADEN=1, ADSC=0, clock=1M/8=125K
      ADMUX=0b01000100; // use AVCC with external cap at AREF
}
```

```
/*--Display pattern for common cathode 7-segment to display 0-9 and C,E,R--*/
Unsignedshortmask (intnum) {
      switch (num) {
            case0:return0b00111111;//0x3F;Display "0".
            case1:return0b00000110;//0x06;Display "1".
            case2:return0b01011011;//0x5B;Display "2".
            case3:return0b01001111;//0x4F;Display "3".
            case4:return0b01100110;//0x66;Display "4".
            case5:return0b01101101;//0x6D;Display "5".
            case6:return0b01111101;//0x7D;Display "6".
            case7:return0b00000111;//0x07;Display "7".
            case8:return0b01111111;//0x7F;Display "8".
            case9:return0b01101111;//0x6F;Display "9".
            case10:return0b00111001;//0x39;//display C
            case11:return0b01111001;//0x79;//display E
            case12:return0b01110111;//0x77;//display R
            }
}

/*--------Displaying Temperature at 7-segment--------*/

Voiddisp_data(){
      for(inti=0;i<=40;i++){
      /*---Display the left most digit----*/
            PORTD|=((0x01<<PD3)|(0x01<<PD2)|(0x01<<PD1));
            PORTC=mask(val[1]);
            PORTD&=~(0x01<<PD0);
            _delay_ms(15);

      /*---Display the 2nd digit----*/
            PORTD|=((0x01<<PD3)|(0x01<<PD2)|(0x01<<PD0));
            PORTC=mask(val[2]);
            PORTD&=~(0x01<<PD1);//Display the 2nd digit.
            _delay_ms(15);

            /*---Display the 3rd digit----*/
            PORTD|=((0x01<<PD3)|(0x01<<PD1)|(0x01<<PD0));
            PORTC=mask(val[3]);
            PORTC&=~(0x01<<PC7);
            PORTD&=~(0x01<<PD2);
            _delay_ms(15);

            /*---Display C(Centigrade ) at the Right most digit----*/
            PORTD|=((0x01<<PD2)|(0x01<<PD1)|(0x01<<PD0));
            PORTC=mask(10);
            PORTD&=~(0x01<<PD3);
            _delay_ms(15);
      }
}

/*----Displaying ERROR----*/

Voiddisp_error(){
      for(inti=0;i<=40;i++){
            PORTB|=((0x01<<PB7));//To Blink the light at PORTB-7
```

```
            /*-----Display E at 2nd digit---*/
            PORTD|=((0x01<<PD3)|(0x01<<PD2)|(0x01<<PD0));
            PORTC=mask(11);
            PORTD&=~(0x01<<PD1);
            _delay_ms(15);
            /*-----Display R at 3rd digit---*/
            PORTD|=((0x01<<PD3)|(0x01<<PD1)|(0x01<<PD0));
            PORTC=mask(12);
            PORTC&=~(0x01<<PC7);
            PORTD&=~(0x01<<PD2);
            _delay_ms(15);
      }
}
/*-----Reference Temperature Initialization-----*/
VoidrefTemp_init(){
      uint16_tadc_value;
      ADCSRA=0b10000011;// ADEN=1, ADSC=0, clock=1M/8=125K
      ADMUX=0b01000011; // use AVCC with external cap at AREF
      //read temp, update display
      ADCSRA|=(0x01<<ADSC);// start conversion
      while(ADCSRA&(0x01<<ADSC));// wait for the conversion to end
      adc_value=ADC;// get ADC result
      tmp1=adc_value;//Convert to degree Centigrade
      Res=(tmp1*100.0*(5.0/1024));  // Multiply by 100 to get only the higher
3 digits

      val[0]=(int)(Res/1000);//1000th value, this value is not used at
displaying data.
      val[1]=(int)(Res/100);//100th value
      val[2]=(int)((Res/10))%10;//10th value
      val[3]=((int)Res%100)%10;//unit value
      disp_data();//display reference temperature
      PORTD|=((0x01<<PD3)|(0x01<<PD2)|(0x01<<PD1)|(0x01<<PD0));   //stop the
car.
      _delay_ms(400);

}

/*-----Initialization of Timer-1------*/
Voidtimer_init(){
      TCCR1A=0b00000000;//Timer/Counter1 Control Register A normal operation
activated.
      TCCR1B=0b00001101;//Timer/Counter1 Control Register B set to pre-scalar
/1024
      OCR1A=0x7271;//30 sec delay @1Mhz
      TIMSK|=(0x01<<OCIE1A);//Timer/Counter Interrupt Mask Register with
Timer/Counter1, Output Compare A Match Interrupt Enabled.
}

/*------Display Current temperature at each 30 sec. delay------*/
ISR(TIMER1_COMPA_vect){
uint16_tadc_value;
/*-----check if the car is on straight path or not------*/
      while((Go_left==1)|(Go_right==1)){
            PORTD&=(~(0x01<<0)&~(0x01<<1)&~(0x01<<2)&~(0x01<<3));
            PORTC=0xC0;
            sens_dat=PINB;
```

```
            sens_Cntr=sens_dat&0x08;//???????
            sens_Lft=(sens_dat&0x70)>>4;//???????
            sens_Rgt=(sens_dat&0x07);//??????

            run();
      }
      /*----convert from analog to digital data, check for the error
condition and display result on 7-segment------*/
      _delay_ms(150);
      PORTD&=(~(0x01<<7)&~(0x01<<6)&~(0x01<<5)&~(0x01<<4));//Stop
      //read temp, update display
      ADCSRA|=(0x01<<ADSC);// start conversion
      while(ADCSRA&(0x01<<ADSC));// wait for the conversion to end
adc_value=ADC;// get ADC result
      tmp=adc_value;//Convert to degree Centigrade
      Res=(tmp*100.0*(5.0/1024));    // Multiply by 100 to get only the higher
3 digits

      val[0]=(int)(Res/1000);//1000th value
      val[1]=(int)(Res/100);//100th value
      val[2]=(int)((Res/10))%10;//10th value
      val[3]=((int)Res%100)%10;//unit value
      /*-------Temp. and error showing conditions------*/
      if(tmp<=tmp1){
            disp_data();
            PORTD|=((0x01<<PD3)|(0x01<<PD2)|(0x01<<PD1)|(0x01<<PD0));
            _delay_ms(400);
}
      else{
            disp_data();
            PORTD|=((0x01<<PD3)|(0x01<<PD2)|(0x01<<PD1)|(0x01<<PD0));
            _delay_ms(400);

            for(;;){
                  disp_error();
                  PORTD|=((0x01<<PD3)|(0x01<<PD2)|(0x01<<PD1)|(0x01<<PD0));
                  PORTB&=~(0x01<<PB7);
                  _delay_ms(400);
            }
      }
}

/*-----Car Running conditions-----*/
Voidrun(){
            if((sens_Cntr==0)&(sens_Lft==sens_Rgt)){//Go straight
                  PORTD|=((0x01<<4)|(0x01<<6));
                  PORTD&=(~(0x01<<5)&~(0x01<<7));
                  Go_left=0;
                  Go_right=0;
            }
            elseif((sens_Cntr==1)&(sens_Lft<sens_Rgt)){// Go left
                  PORTD|=(0x01<<6);
                  PORTD&=(~(0x01<<4)&~(0x01<<5)&~(0x01<<7));
                  Go_left=1;
                  Go_right=0;
            }
            elseif((sens_Cntr==1)&(sens_Lft>sens_Rgt)){// Go right
```

```
                PORTD|=(0x01<<4);
                PORTD&=(~(0x01<<5)&~(0x01<<6)&~(0x01<<7));
                Go_left=0;
                Go_right=1;
            }
}
/*-------------------7-segment Display checking at the beginning ------------
------------*/
Voiddisp_chk(){
      PORTD=0xF0;
      for(inti=0;i<4;i++){
            PORTC=0x00;
            for(intj=0;j<7;j++){
                  PORTC|=(0x01<<j);
                  _delay_ms(100);
            }
            _delay_ms(200);
      }
      PORTC=0x00;
      _delay_ms(200);
}

Intmain(void){
      cli();
      uc_init();
      port_init();
      disp_chk();

      refTemp_init();
      adc_init();
      sei();
      timer_init();

      while(1){
      sens_dat=PINB;
            sens_Cntr=(sens_dat&0x08)>>3;//Getting the center sensor data
            sens_Lft=(sens_dat&0x70)>>4;//Getting the left sensors data
            sens_Rgt=(sens_dat&0x07);////Getting the Right sensors data

            run();
}
      return0;
}
```

Appendix B

Circuit Diagram

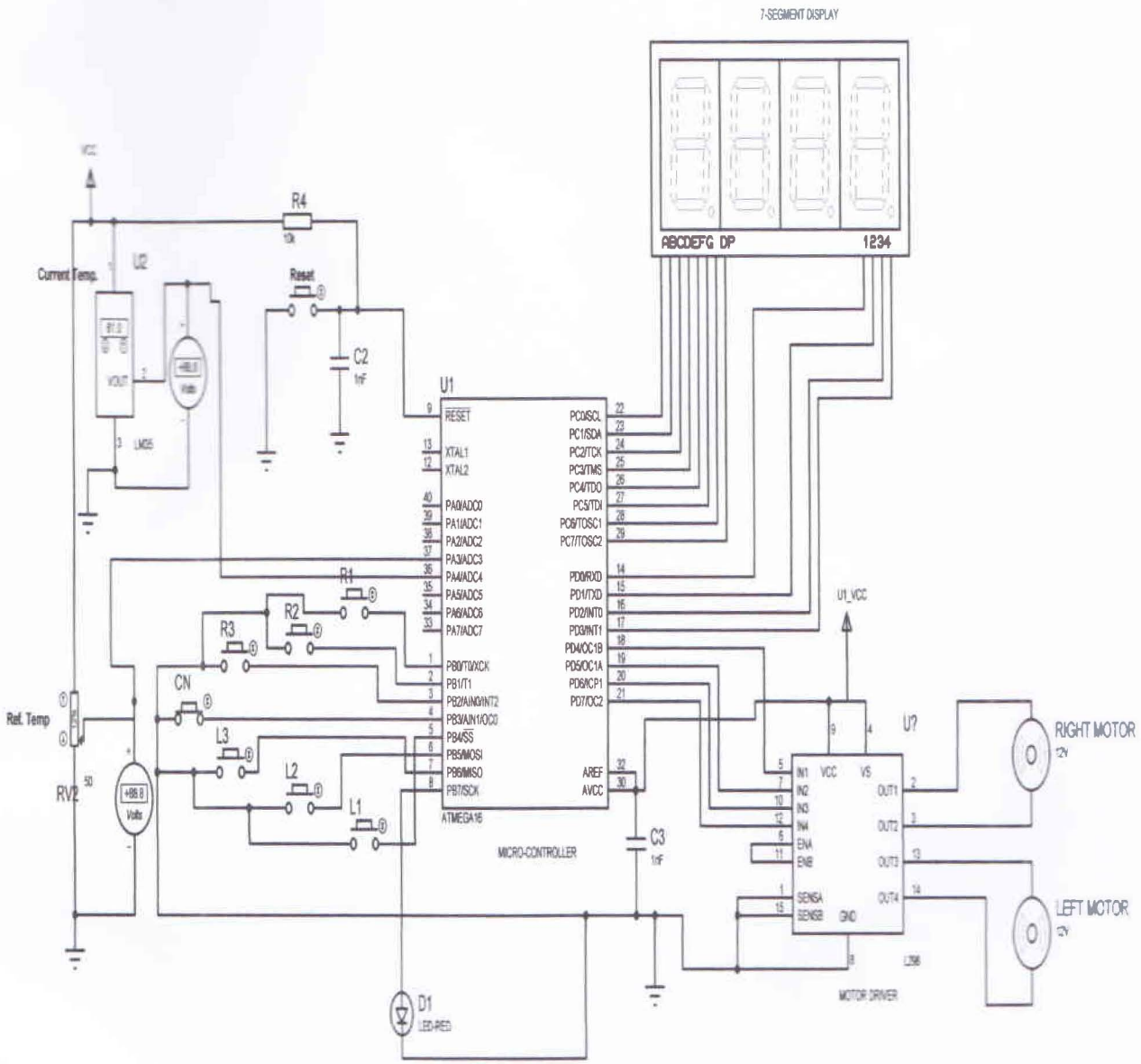


Figure B1: Circuit Diagram.