MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(Affiliated to OU)

Hyderabad - 500034

A MINI-PROJECT REPORT

ON

“MAZE SOLVING BOT”

Submitted by

SAKHIB NAWAB	1604-21-735-018
MOHD SHOYAB	1604-21-735-019
SYED TAJUDDIN	1604-21-735-020

Department of Electronics and Communications

Muffakham Jah College of Engineering and Technology

MINI PROJECT CO-ORDINATOR AND AFFILIATION

Subuhangi Saxena (Assistant Professor, ECED, MJCET)

MAZE SOLVING BOT

Table of Contents

CHAPTER-1

ABSTRACT

This project presents the development and implementation of a maze-solving robot, designed to autonomously navigate through complex labyrinthine structures. The maze-solving bot integrates various sensors, actuators, and algorithms to efficiently explore and find the optimal path from the entrance to the exit of the maze. The primary focus of this project is to design a versatile and adaptable robotic system capable of handling different maze configurations and environmental conditions.

The maze-solving bot employs a combination of sensor technologies, including ultrasonic and infrared sensors, to perceive its surroundings accurately. These sensors enable the robot to detect walls, obstacles, and open pathways, providing essential data for navigation. Through a fusion of sensor data and advanced algorithms, the robot constructs a map of the maze and plans its movements accordingly.

To navigate through the maze, the robot utilizes a pathfinding algorithm, such as Dijkstra's algorithm or A* algorithm, to determine the shortest path from its current position to the maze's exit. These algorithms consider factors such as distance, obstacles, and possible paths to optimize the route. Additionally, the robot employs real-time decision-making mechanisms to adapt its path based on dynamic changes in the maze environment.

The maze-solving bot is equipped with motorized actuators, such as wheels or tracks, enabling it to move efficiently within the maze. The robot's movement is controlled through a microcontroller or a single-board computer, which orchestrates the interaction between sensors, actuators, and algorithms. This centralized control system ensures coordinated navigation and smooth operation of the robot.

Furthermore, the project explores the integration of machine learning techniques to enhance the maze-solving capabilities of the robot. By leveraging reinforcement learning or neural networks, the robot can learn from its experiences and optimize its navigation strategies over time. This adaptive learning approach enables the robot to improve its performance and adapt to new maze environments with minimal human intervention.

The maze-solving bot is designed with modularity and scalability in mind, allowing for easy customization and expansion. The robot's hardware and software components are designed to be interchangeable, facilitating upgrades and modifications as needed. This flexibility enables the robot to tackle a wide range of maze configurations and challenges, making it suitable for educational, research, and practical applications.

In conclusion, the maze-solving bot presented in this project demonstrates an effective and efficient solution for autonomous navigation in maze-like environments. Through the integration of sensor technologies, algorithms, and machine learning techniques, the robot exhibits robust navigation capabilities and adaptability to varying maze conditions. This project lays the foundation for further advancements in robotics and autonomous systems, with potential applications in areas such as search and rescue, exploration, and industrial automation.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to everyone who contributed to the successful completion of this mini project on the MAZE SOLVING BOT.

I would like to thank our HOD DR. AYESHA NAAZ who has always been the source of inspiration

I am deeply thankful to Miss. SHUBHANGI SAXSENA, our project supervisor, for the invaluable guidance, support, and encouragement throughout the project. Their expertise and insights played a pivotal role in shaping the project and enhancing its quality

I extend my appreciation to my fellow team members who collaborated tirelessly, bringing diverse skills and perspectives to the project. Our collective efforts and dedication have culminated in the successful implementation of the MAZE SOLVING BOT

I would like to acknowledge MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY providing the necessary resources and conductive environment for the project work

Thank you everyone involved for making this project rewarding and enriching experience.

INTRODUCTION

The fascination with maze-solving algorithms and robotics has long captivated both enthusiasts and researchers alike. Mazes, intricate structures of pathways and dead-ends, serve as a compelling testing ground for developing autonomous navigation systems. The ability to navigate through complex environments autonomously is a fundamental challenge in robotics, with applications ranging from search and rescue operations to industrial automation. This project delves into the design and implementation of a maze-solving robot, a versatile robotic platform equipped with sensors, actuators, and intelligent algorithms capable of autonomously navigating through intricate maze configurations.

The maze-solving robot represents a convergence of various disciplines, including robotics, computer science, and engineering. At its core, the project aims to address the fundamental question of how a robot can efficiently explore and navigate an unknown maze environment. By leveraging sensor technologies, advanced algorithms, and machine learning techniques, the robot endeavours to emulate the decision-making processes inherent in human navigation while surpassing human limitations in efficiency and precision.

The significance of maze-solving robots extends beyond mere entertainment or academic curiosity. In practical terms, such robots hold immense potential in scenarios where human intervention is impractical or hazardous. Search and rescue operations in disaster-stricken areas, exploration of unknown territories, and inspection of hazardous environments are just a few examples where autonomous maze-solving robots can play a pivotal role. Furthermore, the insights gained from developing these robots contribute to the broader field of robotics, advancing our understanding of autonomous navigation and paving the way for future innovations.

The development of a maze-solving robot involves tackling several key challenges, each requiring innovative solutions. Firstly, the robot must possess the ability to perceive its surroundings accurately. This necessitates the integration of various sensors, such as ultrasonic and infrared sensors, enabling the robot to detect walls, obstacles, and open pathways within the maze. Accurate perception forms the foundation for effective navigation, allowing the robot to construct a mental map of the maze and plan its movements accordingly.

Navigation within the maze presents its own set of challenges, requiring the robot to employ sophisticated algorithms for pathfinding and decision-making. Algorithms such as Dijkstra's algorithm or A* algorithm are commonly utilized to determine the shortest path from the entrance to the exit of the maze while considering factors such as distance, obstacles, and potential pathways. Real-time decision-making mechanisms further enhance the robot's adaptability, enabling it to dynamically adjust its path in response to changing environmental conditions.

Moreover, the project explores the integration of machine learning techniques to augment the maze-solving capabilities of the robot. By leveraging reinforcement learning or neural networks, the robot can learn from its experiences and refine its navigation strategies over time. This adaptive learning approach imbues the robot with the ability to optimize its performance and adapt to new maze environments with minimal human intervention.

In summary, the maze-solving robot project represents a multifaceted endeavour at the intersection of robotics, artificial intelligence, and engineering. By designing a versatile robotic platform capable of autonomously navigating complex mazes, the project aims to advance our understanding of autonomous systems while unlocking new possibilities for real-world applications. Through innovative sensor technologies, algorithms, and machine learning techniques, the maze-solving robot exemplifies the potential of robotics to tackle challenging problems and push the boundaries of human achievement.

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 successful 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.

LIMITATIONS

The system has restricted to the following limitations

• 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.

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.

This report 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.

This report also 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.

This also contains the process explanation with working algorithm, flowchart and sketch of the Arduino.

And contains all the results obtained from the software experiments that include the algorithm implemented in a program.

Finally, this report will summarize the 3rd year mini project. The conclusion, suggestions or recommendations for improvements that can be implemented in future are discussed within this chapter.

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 colour. 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 limitations of efficiency and reliability. Robots are made to replace dependency of human force partially. The project is somehow designed to perform the similar task.

CHAPTER-2

Technology and Literature Survey

The maze solver bot is a self- operating robot that detects and follows a line that is drawn on the floor and through that solves a maze. 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

BASIC OPERATION

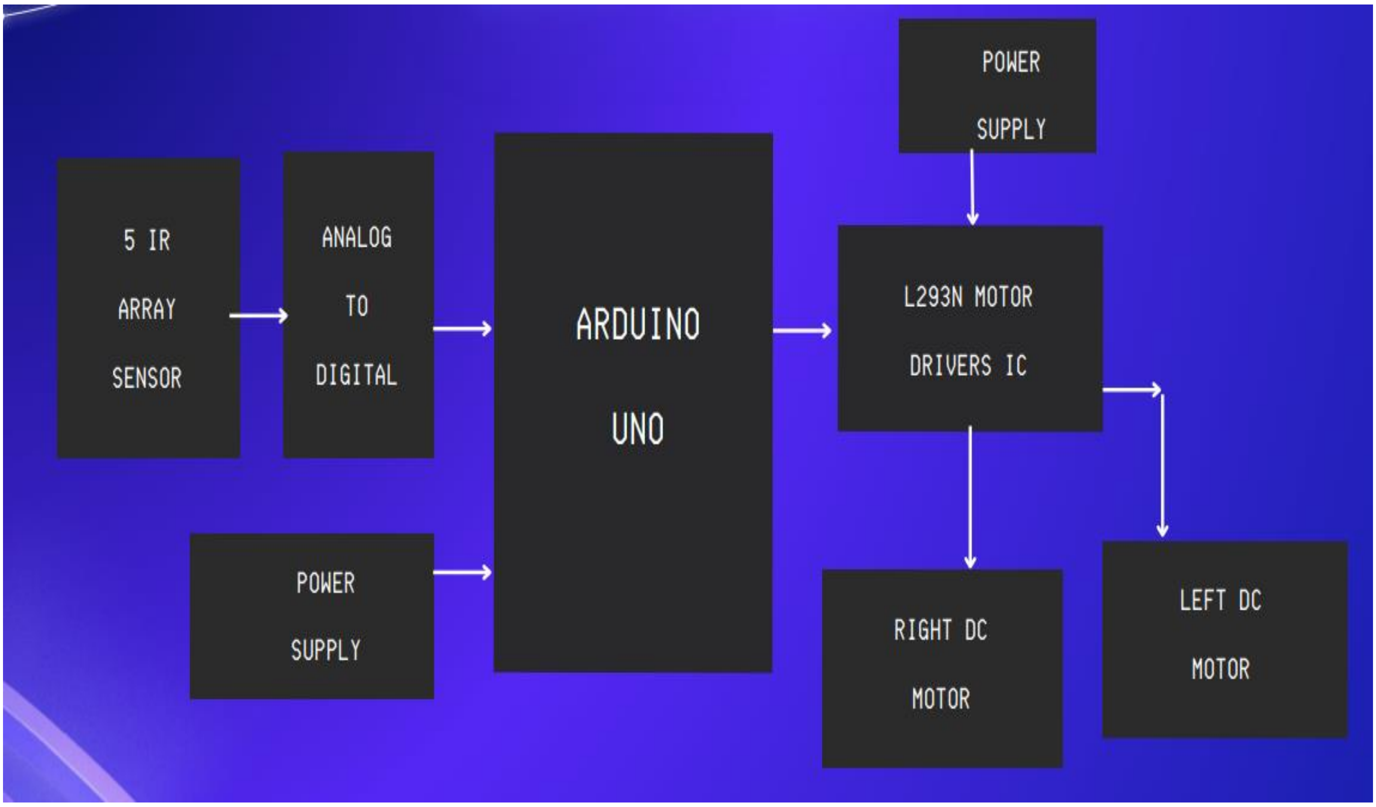
The basic operations of line follower are as follows:

• 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 makes 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

BLOCK DIAGRAM



CHAPTER-3

Hardware Required:

The hardware required is divided in the following category:

Input System

The detail of each component is discussed below.

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.

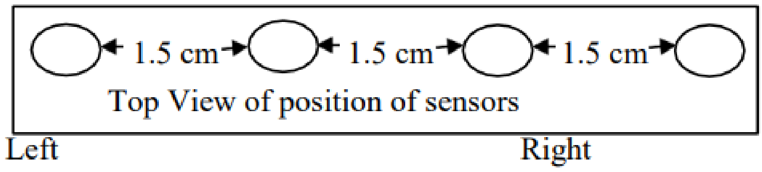


Figure 2-2: Arrangements of the Sensor

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. 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.

3. 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)

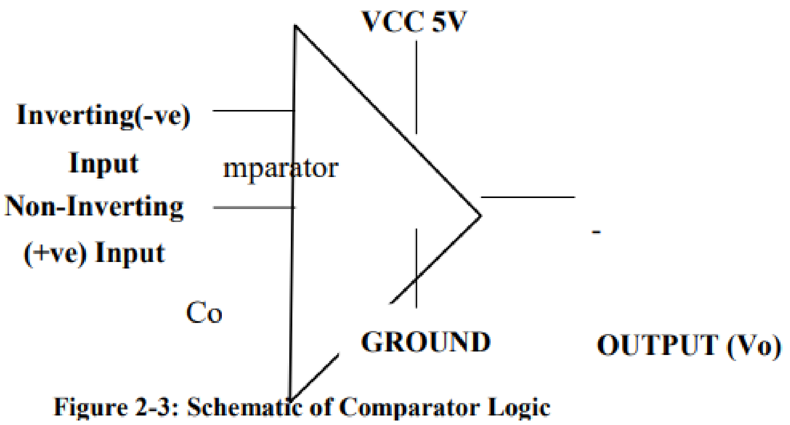


Figure 2-3: Schematic of Comparator Logic

4. Optical sensor:

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

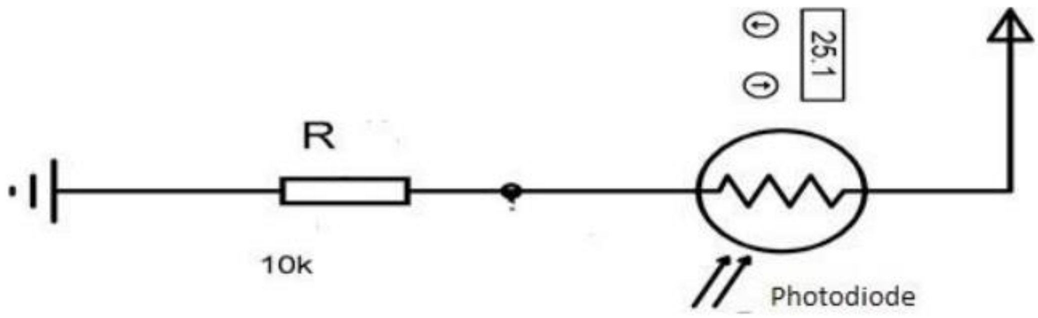


Figure 2-4: Optical Sensor schematic

5. 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 AVR1.

6. 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 Memory, 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

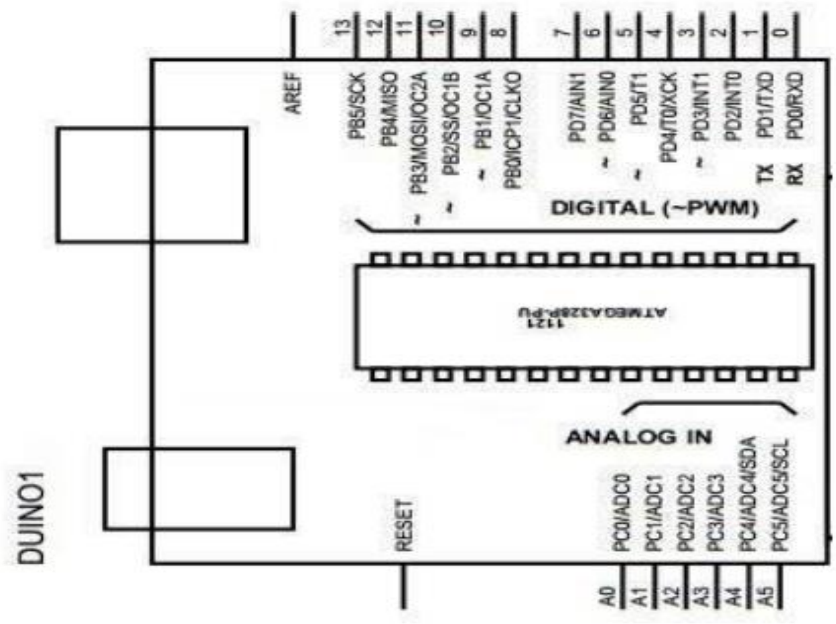


Figure 2-5: Arduino Uno Schematic

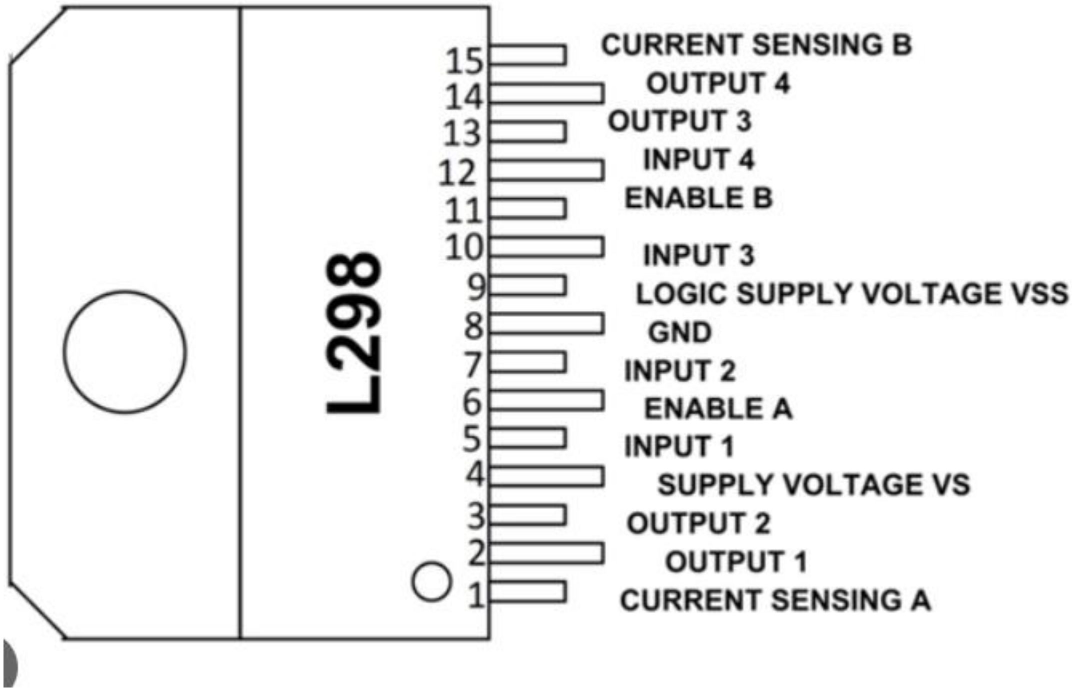
7. Output System:
   The output system is designed with the function of the following components

8. Motor Driver:

   Motor driver is a current enhancing device; it can also be acting 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.

9. IC L298N

   This is a motor driver IC that can drive two motors 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 L298N has 2 Channels. One channel is used for one motor.



10. TT 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.

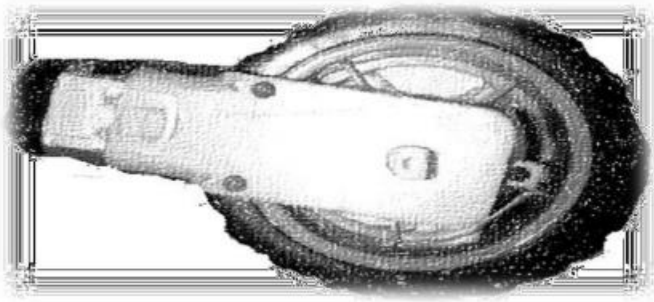


Figure 2-7: Low Volt DC Gear Motor attach with Wheel.

11. Proximity Sensor

The combination of IR- LED and Photodiode is used as the reflective optical sensor. It generates 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.

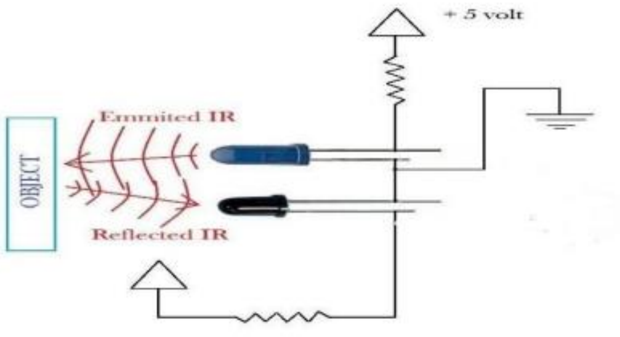


Figure 2-9: Proximity Sensor

12. Software Required

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

13. Summary

The system is completed in to the three division 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) are 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-4

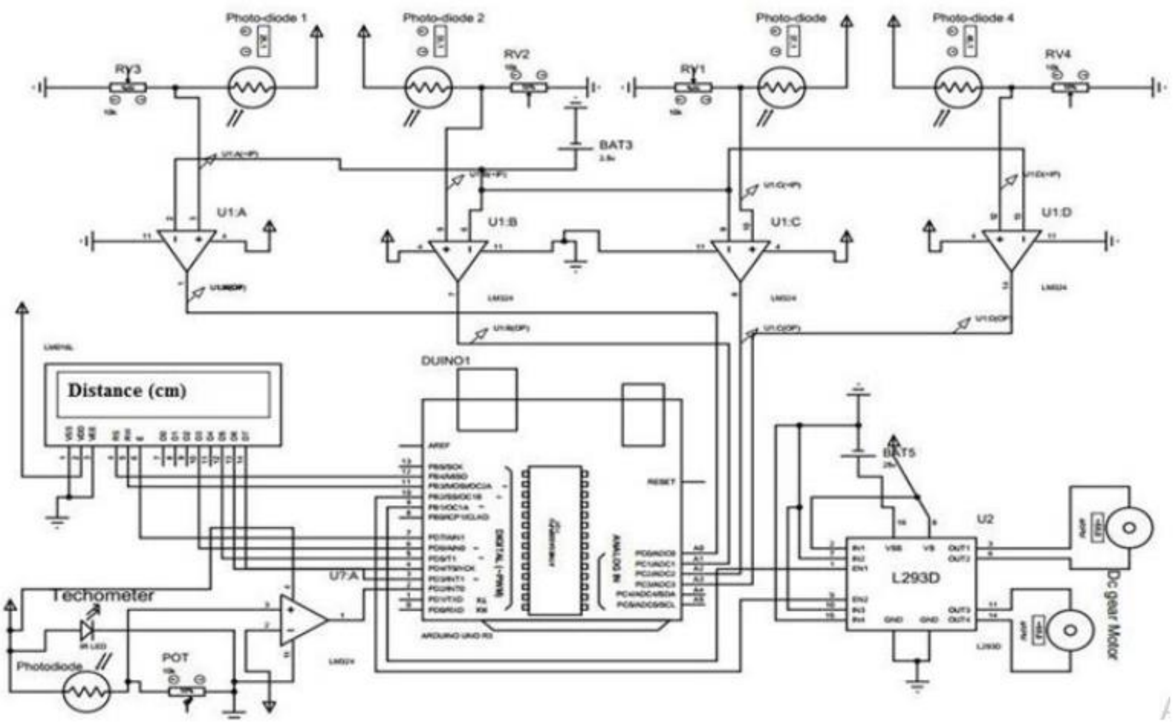
DESIGN AND IMPLEMENTATION

1. Schematics

The schematic of the “Maze Solving 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:

- Arduino 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 tachometer

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 below



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 12
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

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 motor driver which finally drive the motor in such way that, it produces 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-5

CONCLUSION

The line following robot is automobile system that has ability to recognize its 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 Electromechanically robot dimension is 192×100×70 mm3 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 its 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.

FUTURE WORK

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

So, the development features in brief:

- Use of colour sensor.
- Use of ccd camera for better recognition and precise tracking the place

CHAPTER-6

BIBLOGRAPHY

[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]

CHAPTER-7

APPENDICES:

PROGRAM CODE:

```
int s1 = 0;
int s2 = 0;
int s3 = 0;
int s4 = 0;
int s5 = 0;
#define led 13
#define lpwm 10
#define rpwm 11
int spd = 125;
const int lmf = A5;
const int lmb = A4;
const int rmf = A2;
const int rmb = A3;
void setup()
{
  for (int i = 2; i <= 5; i++)
  pinMode(i, INPUT);
  pinMode(12, INPUT);
  pinMode(A5, OUTPUT);
  pinMode(A4, OUTPUT);
  pinMode(A3, OUTPUT);
  pinMode(A2, OUTPUT);
  pinMode(lpwm, OUTPUT);
  pinMode(rpwm, OUTPUT);
}
void loop()
```

```
{
  s1 = digitalRead(2);
  s2 = digitalRead(3);
  s3 = digitalRead(4);
  s4 = digitalRead(5);
  s5 = digitalRead(6);
if (s3 == 0)
  {
    forward();
    digitalWrite(led, LOW);
  }
if (s4 == 0 || s5 == 0)
  {
    right();
    digitalWrite(led, LOW);
  }
 if (s1 == 0 || s2 == 0)
  {
    left();
    digitalWrite(led, LOW);
  }
 if (s1 == 0 && s3 == 0 && s5 == 0)
  {
    analogWrite(led, 255);
    delay(5);
  }
 if (s1 == 0 && s2 == 0 && s3 == 0 && s4 == 0 && s5 == 0)
  {
    stop();
  }
}
```

```
void forward()
{
  analogWrite(lpwm, spd);
  analogWrite(rpwm, spd);
  digitalWrite(lmf, HIGH);
  digitalWrite(lmb, LOW);
  digitalWrite(rmf, HIGH);
  digitalWrite(rmb, LOW);
  delay(10);
}
void right()
{
  analogWrite(lpwm, spd);
  analogWrite(rpwm, spd);
  digitalWrite(lmf, LOW);
  digitalWrite(lmb, HIGH);
  digitalWrite(rmf, HIGH);
  digitalWrite(rmb, LOW);
  delay(10);
}
void left()
{
  analogWrite(lpwm, spd);
  analogWrite(rpwm, spd);
  digitalWrite(lmf, HIGH);
  digitalWrite(lmb, LOW);
  digitalWrite(rmf, LOW);
  digitalWrite(rmb, HIGH);
  delay(10);
}
void stop()
```

```
{
  analogWrite(lpwm, 0);
  analogWrite(rpwm, 0);
  digitalWrite(lmf, LOW);
  digitalWrite(lmb, LOW);
  digitalWrite(rmf, LOW);
  digitalWrite(rmb, LOW);
  digitalWrite(led, HIGH);
}
```