MAZE SOLVER

Prepared by
Deep Desai (17IT018)
Manush Parikh (17IT061)

Under the supervision of

Prof. Nehal Patel

Prof. Sagar Patel

A Report Submitted to

CharotarUniversity of Science and Technology

for Partial Fulfillment of the Requirements for the

Degree of Bachelor of Technology

in Information Technology

IT244 Software Group Project-I (3rdSem)

Submitted at

DEPARTMENT OF INFORMATION TECHNOLOGY

Chandubhai S. Patel Institute of Technology

At: Changa, Dist: Anand – 388421

November 2018

CERTIFICATE

This is to certify that the report entitled "**Maze Solver**" is a bonafied work carried out by **Mr. Manush Parikh (17IT061)** under the guidance and supervision of **Prof. SagarPatel** for the subject **Software Group Project-I (IT244)** of **3rd** Semester of Bachelor of Technology in **Information Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate **himself**, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under supervision of,

Prof. Sagar Patel
Assistant Professor
Dept. of Information Technology
CSPIT, Changa, Gujarat.

Dr. Parth Shah
Head & Associate Professor
Department of Information Technology
CSPIT, Changa, Gujarat.

Chandubhai S Patel Institute of Technology

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

CERTIFICATE

This is to certify that the report entitled "**Maze Solver**" is a bonafied work carried out by **Mr. Deep Desai (17IT018)** under the guidance and supervision of **Prof. Nehal Patel** for the subject **Software Group Project-I (IT244)** of **3rd** Semester of Bachelor of Technology in **Information Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate **himself**, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under supervision of,

Prof. Nehal Patel
Assistant Professor
Dept. of Information Technology
CSPIT, Changa, Gujarat.

Dr. Parth Shah
Head & Associate Professor
Department of Information Technology
CSPIT, Changa, Gujarat.

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my Prof. Nehal Patelas well as our HOD Dr. ParthShah who gave me the golden opportunity to do this wonderful project on the topic Maze Solver which also helped me in doing a lot of Research and i came to know about so many new things I am really thankful to them.

Secondly i would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

ABSTRACT

This project will undertake the construction and implementation of a wheeled robot that is capable of solving a maze constructed as a line. The structural, mechanical, and electronic components of the bot will be assembled in a manner that will make the bot possible to solve the maze as quickly as possible.
The bot will be rotate independently which will be driven by a motor IC circuit. Information about tracks, the dead ends, and turns will be obtained from sensors on the device. The line of tracks will be determined by IR sensors and information from the sensors will be send to the Arduino microcontroller. To take the correct turns and to avoid dead ends, next time robot is operated, the memory of the microcontroller will also be used.

TABLE OF CONTENTS

List of Figures

CHAPTER 1: INTRODUCTION

1.1 Project Overview

A project on maze solver where the machine will be finding the shortest path to be out of the maze. Concepts of **"Q-learning"** will be applied for solving the maze. Main applications of such automated maze solving robots are, automated crate moving within a warehouse, military applications may involve, robots traversing unknown terrain while avoiding or moving obstacles out of the way and in rescue or emergency scenarios whereby robots are required to go into hazardous situation.

1.2 Problem and Solution Statement

The problem is solving a maze and finding the shortest path to achieve the result is difficult and time consuming which in turn uses a lot of resources and man power and understanding. Thus, all these consequences that we face can be solve by intending the machine to find the shortest path from the maze by applying the basic knowledge of **machine learning** specifically **reinforcement learning**.

1.3 Benchmark (How this solution is better?)

There are various methods to solve the problem and get the shortest path. The method used here is based on **reinforcement learning** and its sub part "**Q-learning**" with is based on the pervious output and updating the learned path on the next condition.

Chapter 2: System Analysis

2.1 User Characteristics

We design the robot to reduce the work of men and reduce the manual effects in hazardous areas. It works on exploring the path and identifying the shortest path. Users do not need any special requirement to use this they just need to define a particular destination point.

2.2 Tools and Technology

2.2.1 Arduino and circuit elements

We will be using:

- 2 Micro Gear Motors –
  - DC Motors with gears and 5v power supply requirement will be used to run the wheels and the Machine in the maze.
- Pair of Motor Brackets –
  - The brackets are sold in pairs, and each bracket includes two M2 screws for securing the motor to the bracket.
- Pair of Wheels and Wires –
  - Wheels will be used for the movement of the robot and wires to connect the connections.

- Ball Caster –

  - Ball casters are a type of caster that features a spherical-shaped roller ball wheel, to provide specific maneuvering abilities.

- Analog Reflectance Sensor Array –

  - The QTR-8A reflectance sensor array is intended as a line sensor, but it can be used as a general-purpose proximity or reflectance sensor. The module is a convenient carrier for eight IR emitter and receiver (phototransistor) pairs evenly spaced at intervals of 0.375" (9.525 mm).

- Arduino Uno –

- The Arduino UNO is a widely used open-source microcontroller board.
- The board features 14 Digital pins and 6 Analog pins.
- It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts.

- 4AAA Rechargable Batteries and Holder –

- This will be the power supply for the circuit to function. The cells are rechargeable and thus can be used times. The battery holder holds the batteries and thus makes it simple.

2.2.2 Q-Learning

The algorithm use for solving the problem is “Q-learning” which is based on reinforcement learning that involves learning based on past experiences and improvising the memory accordingly. The “Q-learning” is algorithm which involves development of the matrix based on the readings and then updating the shortest path with maximum rewards. These is the best algorithm based on problem solving without supervision on the robot.

Example:

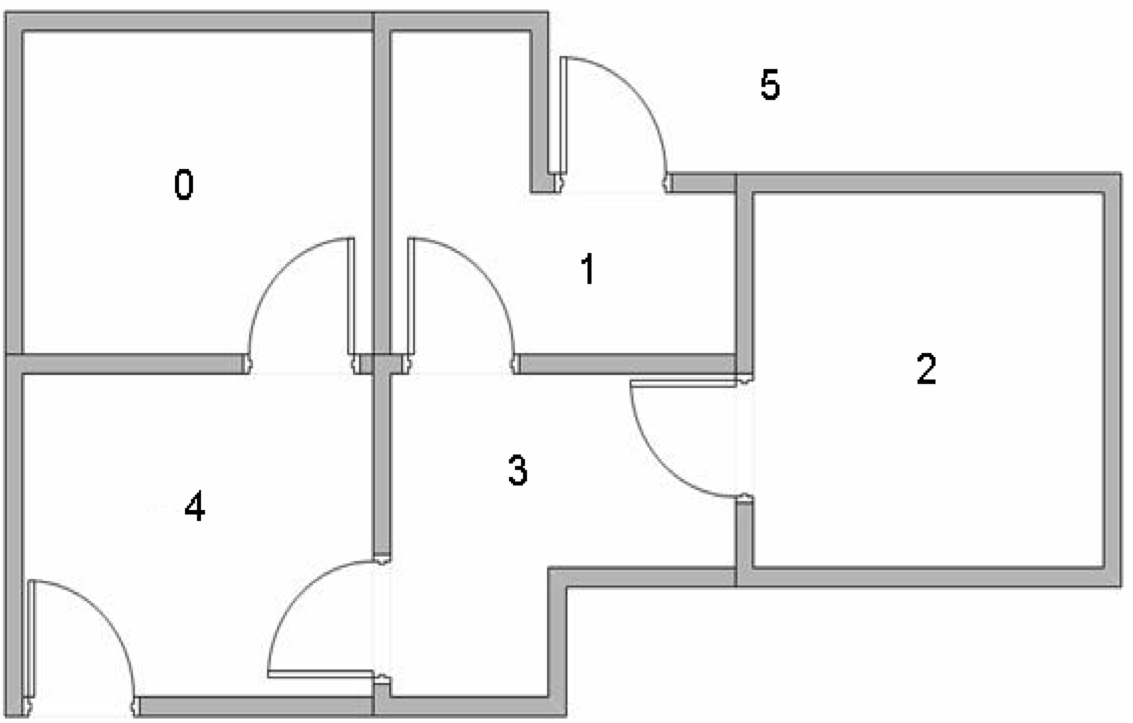

Figure 1. Q-learning Problem

Graphical representation of the problem:

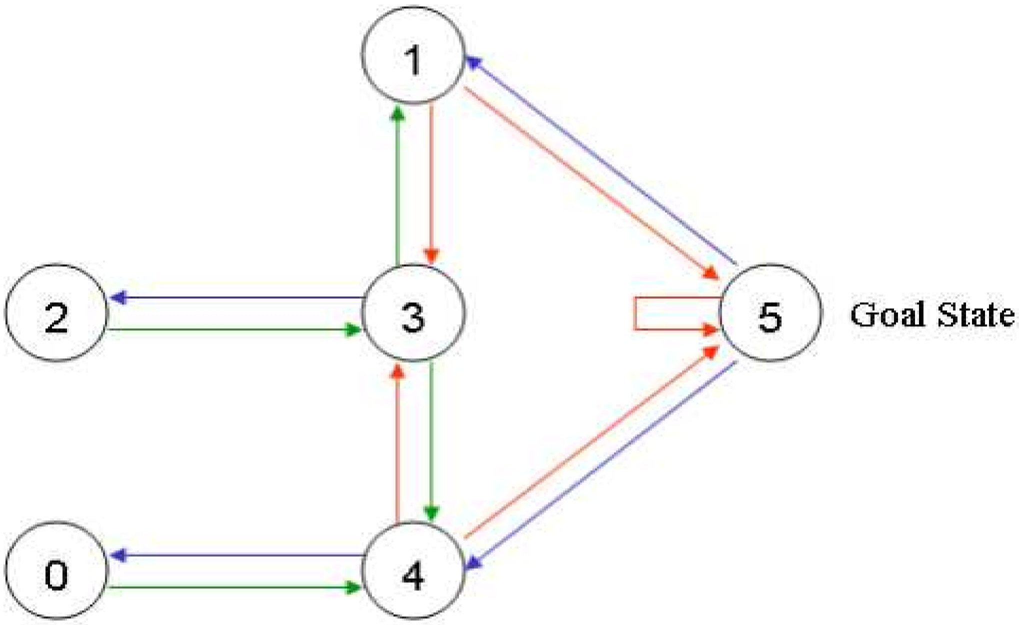

Figure 2. Q-learning Graph-1

Weighted Graph:

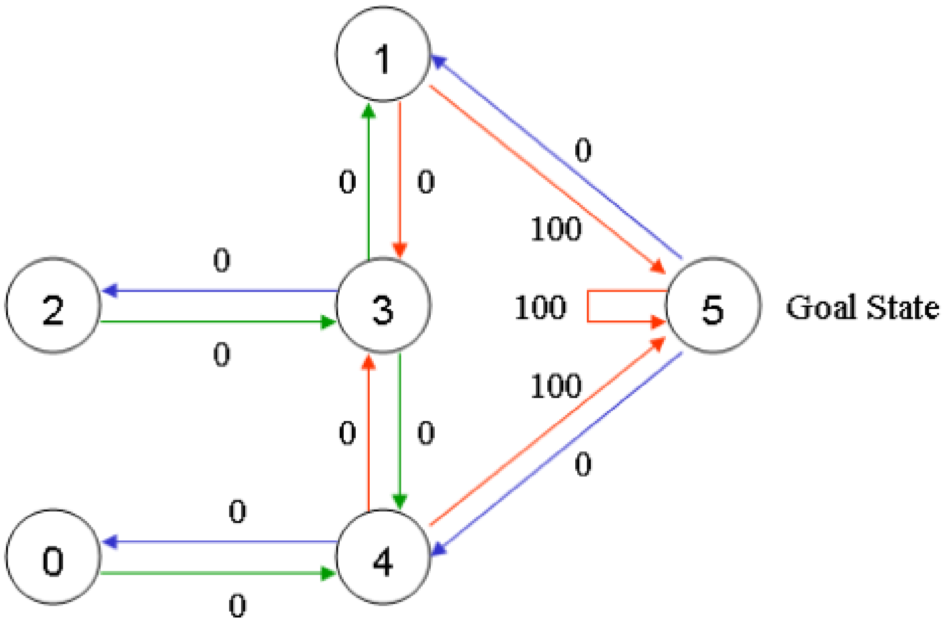

Figure 3. Q-learning Graph-2

Starting the development of r matrix:

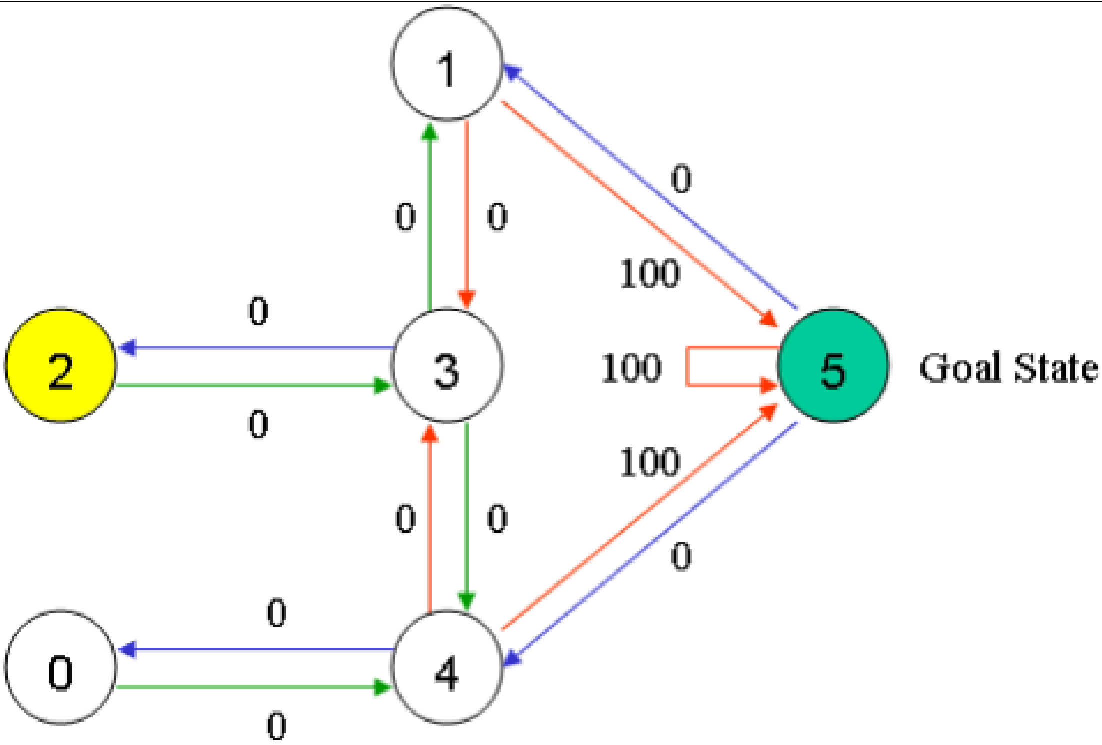

Figure 4. Q-learning Graph-3

Matrix based on 2 as initial stage:

$$
R= \begin{array}{cc}
 & \textbf{Action} \\
\textbf{State} & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\begin{bmatrix}
-1 & -1 & -1 & -1 & 0 & -1 \\
-1 & -1 & -1 & 0 & -1 & 100 \\
-1 & -1 & -1 & 0 & -1 & -1 \\
-1 & 0 & 0 & -1 & 0 & -1 \\
0 & -1 & -1 & 0 & -1 & 100 \\
-1 & 0 & -1 & -1 & 0 & 100
\end{bmatrix}
\end{array}
$$

Figure 5. Q-learning Matrix-1

Formula for Q-Matrix :

Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]

Q- Matrix :

Initialize matrix to zero:

$$Q = \begin{array}{cc} & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Figure 6. Q-learning Matrix-2

Developed Matrix:

$$Q = \begin{array}{cc} & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 400 & 0 \\ 0 & 0 & 0 & 320 & 0 & 500 \\ 0 & 0 & 0 & 320 & 0 & 0 \\ 0 & 400 & 256 & 0 & 400 & 0 \\ 320 & 0 & 0 & 320 & 0 & 500 \\ 0 & 400 & 0 & 0 & 400 & 500 \end{bmatrix} \end{array}$$

Figure 7. Q-learning Matrix-3

Solution :

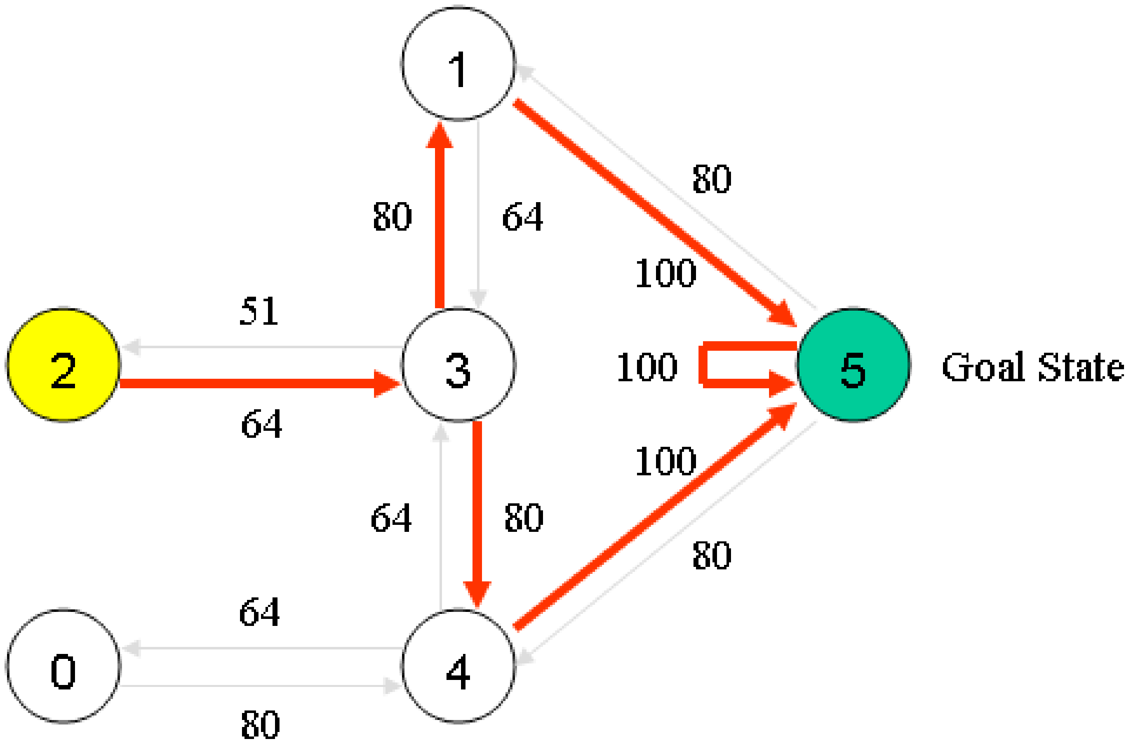

Figure 8. Q-learning Matrix solution

CHAPTER 3: SYSTEM DESIGN

3.1 FLOW OF PROJECT

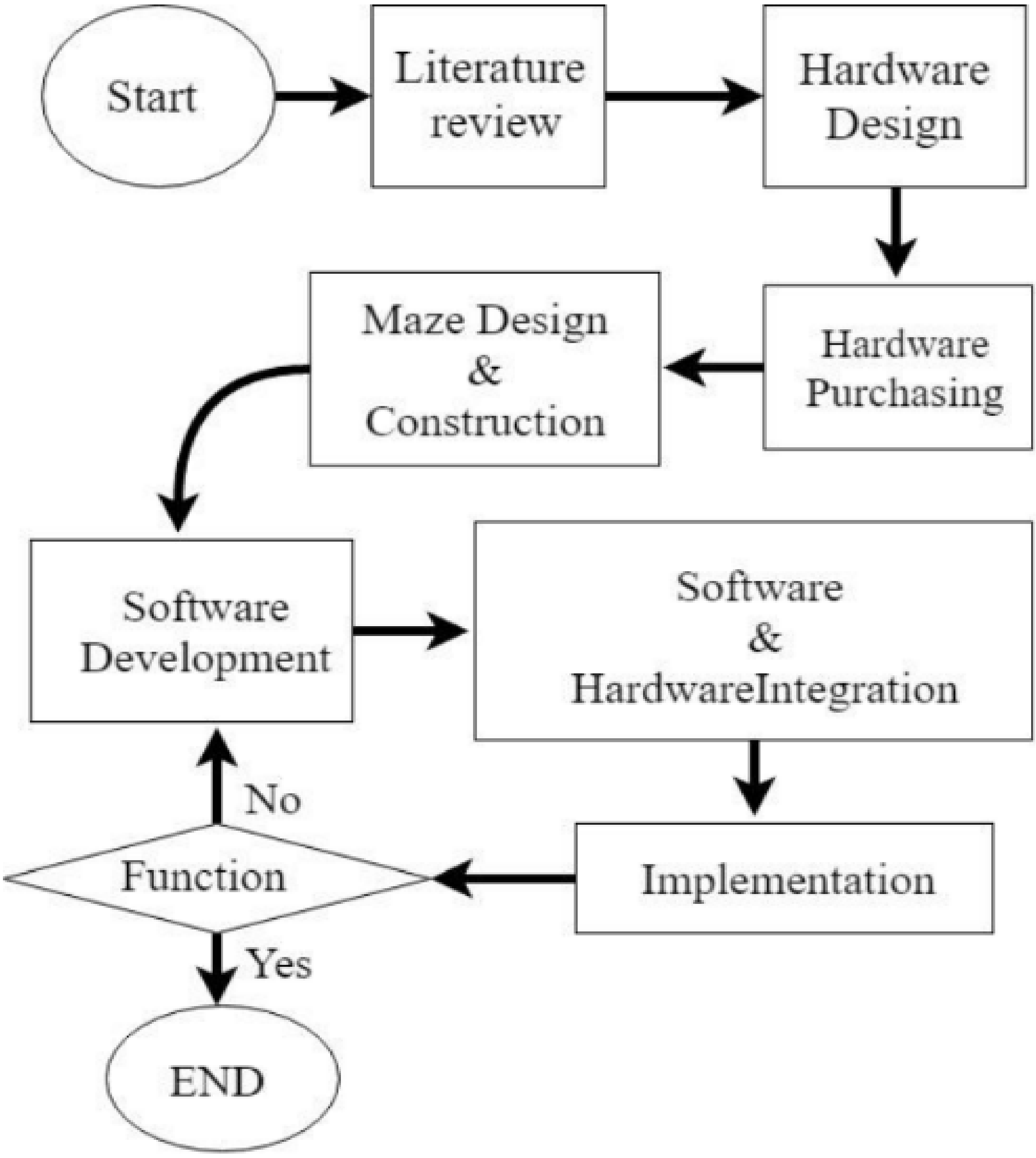

Figure 9. Project flow chart

3.2 Major Functionality

- Easy to setup the robot.
- User friendly as the user just needs to define the destination point.
- Solves the path problem and identifies the path using the IR array.
- Forms updated path and solves the maze using the Q-learing Algorithm.
- Drives through the shortest path.

Chapter 4: Implementation

4.1 Implementation Environment

The Robot and the algorithm has non-GUI based application and can be use in the in various physical application to reduce the men power and rates of failure due to loss of human life.

4.2 Module Specification

The workflow will start with learning the basic concepts of machine learning, reinforcement learning and “Q-learning”. Further on we will apply our learned concepts GUI based maze and find the shortest path for it. Then we will try to combine our concepts and prepared GUI solution to the machine.

4.3 Coding Standards

```
        int mot1=9;
int mot2=6;
int mot3=5;
int mot4=3;

int left=13;
int right=12;

int Left=0;
int Right=0;

void LEFT (void);
void RIGHT (void);
void STOP (void);

void setup()
{
  pinMode(mot1,OUTPUT);
  pinMode(mot2,OUTPUT);
```

```
  pinMode(mot3,OUTPUT);
  pinMode(mot4,OUTPUT);

  pinMode(left,INPUT);
  pinMode(right,INPUT);

  digitalWrite(left,HIGH);
  digitalWrite(right,HIGH);


}

void loop()
{

analogWrite(mot1,255);
analogWrite(mot2,0);
analogWrite(mot3,255);
analogWrite(mot4,0);

while(1)
{
  Left=digitalRead(left);
  Right=digitalRead(right);

  if((Left==0 && Right==1)==1)
  LEFT();
  else if((Right==0 && Left==1)==1)
  RIGHT();
}
}

void LEFT (void)
{
   analogWrite(mot3,0);
   analogWrite(mot4,30);


   while(Left==0)
   {
    Left=digitalRead(left);
    Right=digitalRead(right);
```

```
    if(Right==0)
    {
      int lprev=Left;
      int rprev=Right;
      STOP();
      while(((lprev==Left)&&(rprev==Right))==1)
      {
        Left=digitalRead(left);
        Right=digitalRead(right);
      }
    }
    analogWrite(mot1,255);
    analogWrite(mot2,0);
   }
  analogWrite(mot3,255);
  analogWrite(mot4,0);
}

void RIGHT (void)
{
   analogWrite(mot1,0);
   analogWrite(mot2,30);

   while(Right==0)
   {
    Left=digitalRead(left);
    Right=digitalRead(right);
    if(Left==0)
    {
      int lprev=Left;
      int rprev=Right;
     STOP();
      while(((lprev==Left)&&(rprev==Right))==1)
      {
         Left=digitalRead(left);
         Right=digitalRead(right);
      }
    }
    analogWrite(mot3,255);
    analogWrite(mot4,0);
    }
   analogWrite(mot1,255);
   analogWrite(mot2,0);
}
void STOP (void)
{
analogWrite(mot1,0);
analogWrite(mot2,0);
```

```
analogWrite(mot3,0);
analogWrite(mot4,0);

}
```

4.4 Snapshots of project

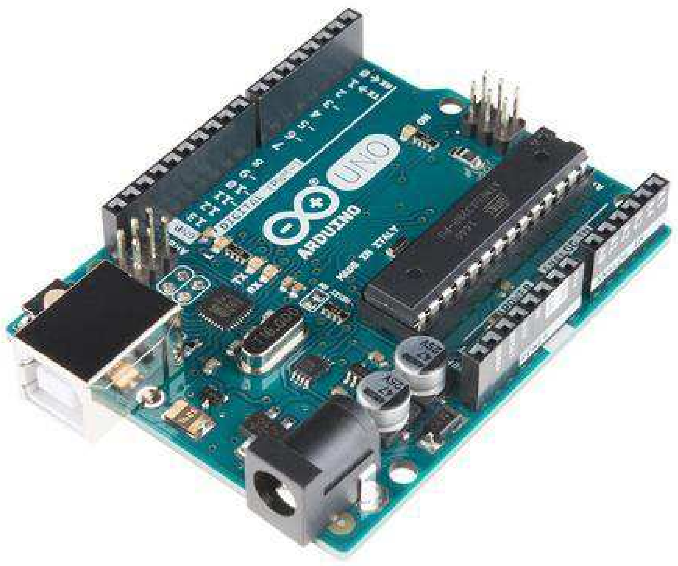

Figure 10. Hardware Component -1

Figure 11. Hardware Component -2

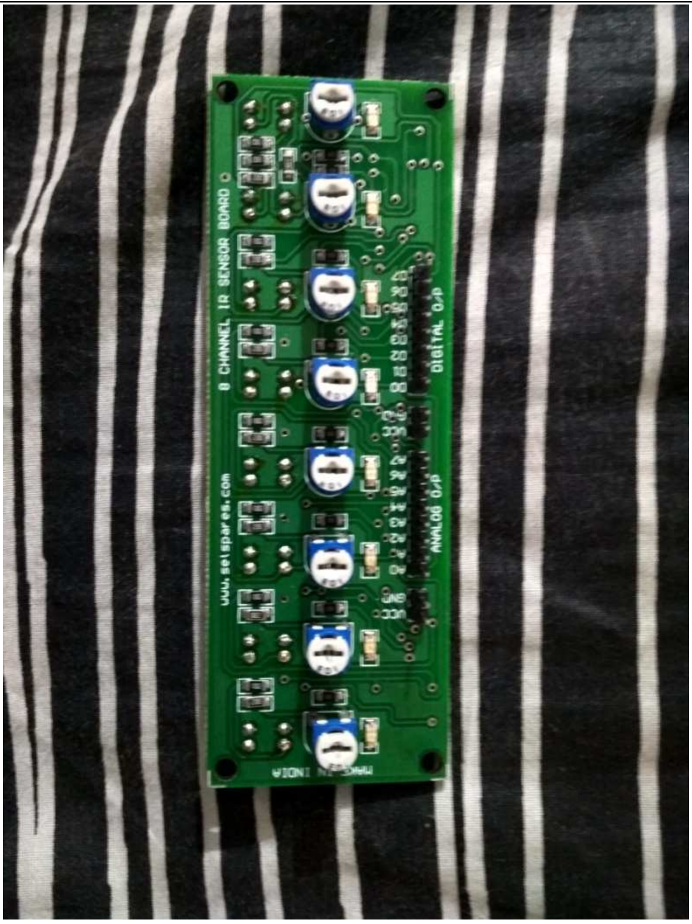

Figure 12. Hardware Component -3

Figure 13. Hardware Component -4

Figure 14. Hardware Component -5

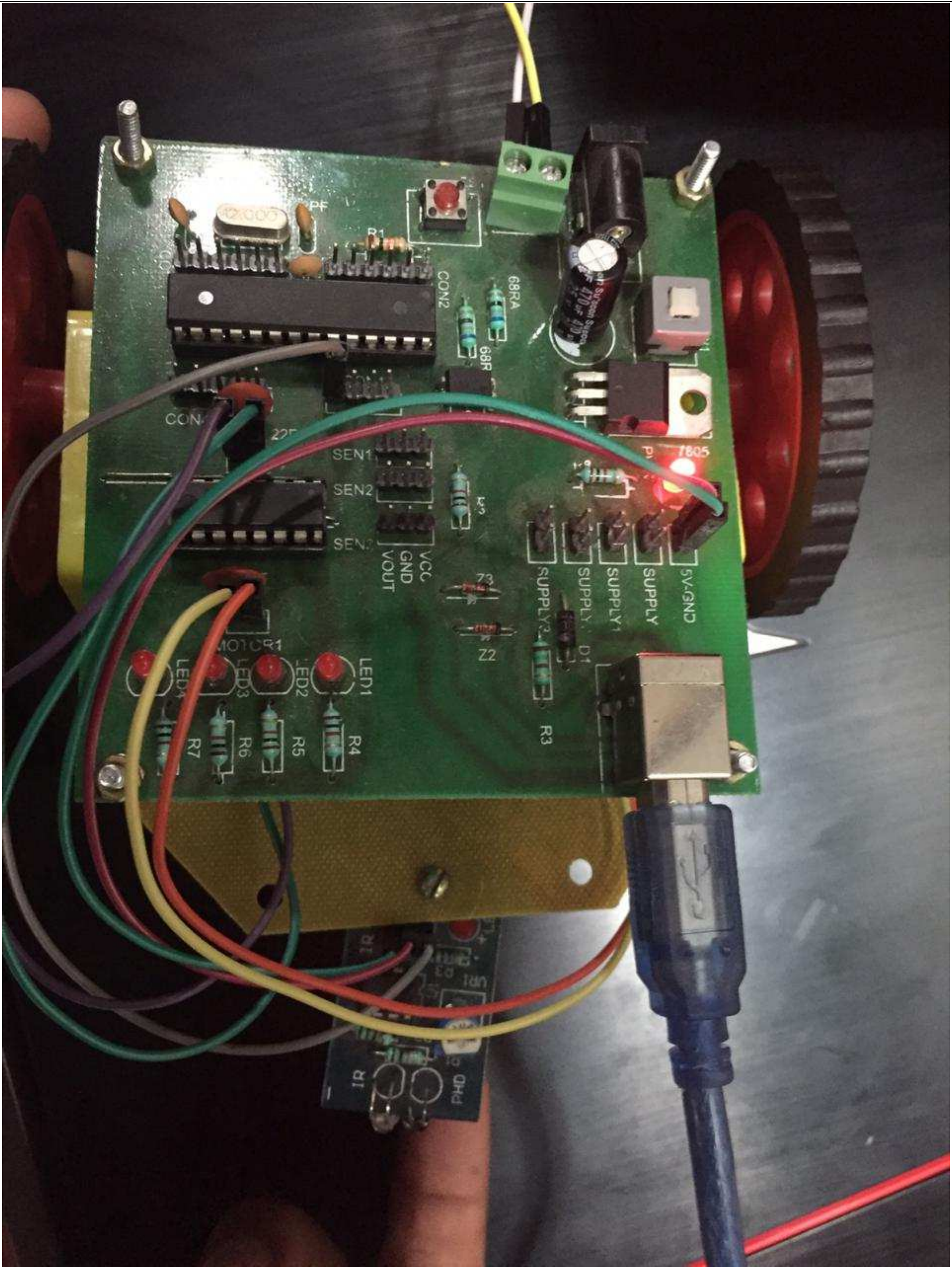

Figure 15. Robot -1

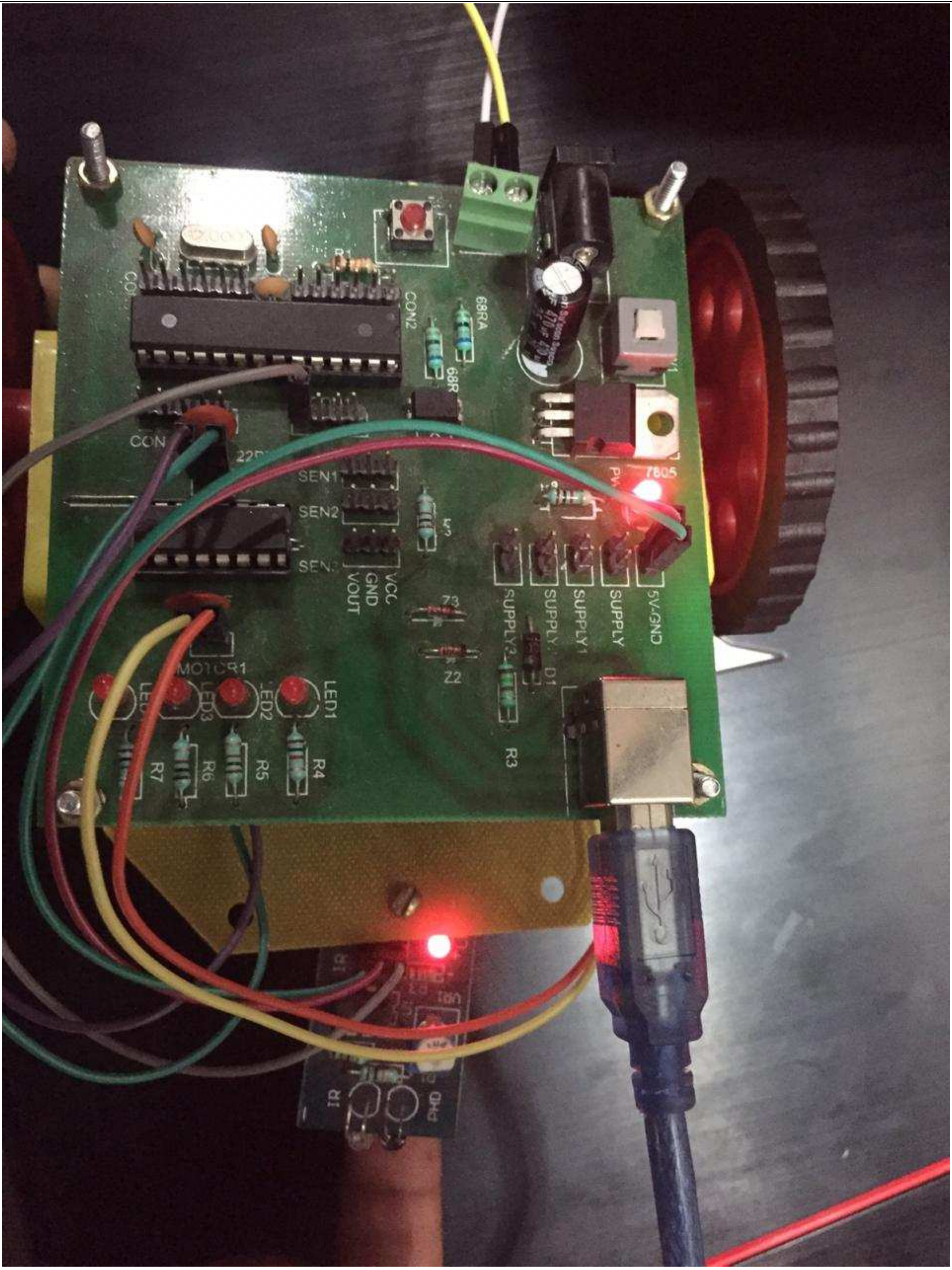

Figure 16. Robot -2

Chapter 5: Constraints and Future Enhancement

The robot will be made more faster and efficient. The placing of tools will be adjusted properly and would make it neat and clean, thus can be used in purposes. In this project, the IR sensor is used for mapping the maze. Other efficient navigating sensors can be used. At last, it needs development in its wheels and body to make it comfortable in any type of surface.

Chapter 6: Conclusion

As a conclusion, the maze solving algorithm has successfully been implemented in the robot and the objectives of the project have been achieved. The robot was tested well and was implemented on different tracks for the best performance of the bot.

This project helps to improve various important information about robotics, knowledge about many decision making algorithms. It's also helped to learn about many electronics components such as motor driver, sensors, etc. This gained knowledge will have a significant impact on future work.

REFERENCES

- https://www.instructables.com/id/Maze-Solving-Robot/
- https://www.engineersgarage.com/contribution/arduino-based-maze-solving-robot
- https://www.zagrosrobotics.com/shop/custom.aspx?recid=69
- https://www.instructables.com/id/Maze-Solving-Robot/
- http://patrickmccabemakes.com/hardware/Maze_Solving_Robot_V3/
- http://mnemstudio.org/path-finding-q-learning-tutorial.html