Automated Maze Robot

by

Mohammad Izzuddin bin Jalil

Dissertation submitted in partial fulfilment of

the requirements for the

Bachelor of Technology (Hons)

(Information and Communication Technology)

JANUARY 2011

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Automated Maze Robot

By

Mohammad Izzuddin Bin Jalil

10727

A project dissertation submitted to the

Information and Communication Technology Programme

Universiti Teknologi PETRONAS

in partial fulfillment of the requirement for the

BACHELOR OF TECHNOLOGY (Hons)

(INFORMATION AND COMMUNICATION TECHNOLOGY)

Approved by,

(Dr. Jafreezal Bin Jaafar)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

JANUARY 2011

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

MOHAMMAD IZZUDDIN BIN JALIL

Abstract

An autonomous maze robot is a robot that can solve a linear maze autonomously. The aim of this project is to develop an autonomous robot for navigation in an unknown maze environment. The problems in this project are the problems in autonomous robot navigation and the problems in navigating in an unknown maze environment. The methodology used in this project is the prototyping methodology. The findings in this project are the maze that is being used, the linear maze; the robot that is being used, the Pololu 3pi Robot; the algorithm that has been chosen to be implemented, the Wall Following Algorithm; the implementation of the project, how the robot operates; the tests that had been ran, the Fault Injection Test, the Non-Functional Test and the Integration Test; and the test results, the success rate and the failure rate. At the end of this report, the author concluded the project and explains what can be done for expansion and continuation.

Acknowledgements

First, the author would like to say Alhamdulillah and thank you to God, for giving him the strength and health to do this project until it is done within the time frame. Not forgetting thank you to the authors' family for providing a lot of things such as money to buy anything that are related to this project, which is the most needed for this project. They also supported him and encouraged him to complete this task so that the author will not procrastinate in doing it.

Then the author would like to thank his supervisor, Dr. Jafreezal Bin Jaafar for guiding him throughout this project. He had helped the author a lot in doing this project in terms of giving advice on how to do the project and what should be done. The author appreciates his dedication and sacrifice towards helping him finishing this project.

Next, the author wants to thank his friends who were helping him in doing this project and sharing out ideas. They were very helpful that when combined and discussed together, the author had this task done even faster and better.

Last but not least, all who had helped the author directly and indirectly in finishing this project. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background of Study:

Exploration is the act of searching or travelling around a terrain including space, for the purpose of discovery of resources or information. Exploration occurs in all non-sessile animal species, including humans [1]. In robotics, the exploration problem deals with the use of a robot to maximize the knowledge over a particular area. The exploration problem arises in mapping and search & rescue situations, where an environment might be dangerous or inaccessible to humans [2]. These environments can be either known or unknown to people in terms of its geographical condition or its accessibility. Due to these factors and uncertainties, we can associate these conditions as a labyrinth or an unknown maze.

And a maze is a complex structure with a series of interconnecting pathways. The term is also used to refer to a graphical puzzle which replicates the maze on a two dimensional medium. A maze is viewed as a puzzle which must be solved, and the solver must work his or her way from the entrance of the maze to an exit, or another location. Getting through a maze can be difficult, leading to the use of "maze" as a slang term for a complex process [3]. There are many types of mazes that can be found nowadays such as the Barnes maze and the Radial Arm maze. And for the record, both of these mazes are used in psychological experiments to study spatial navigation and learning. However, these experiments are tested on rats and mice; a living being that has motor control, vision, smell and pattern recognition and all of these attributes work together as one to help the tested animal to navigate within the maze.

Compared to a robot that does not have all of those attributes, navigating in a maze will be a problem. The robot needs to be 'intelligent' in order to navigate within the maze. And to make the robot smart enough to navigate in the maze, the robot must be well equipped with the right amount and the right type of sensors. The sensors will act as detectors for external input or the robot's surroundings as compared to eyes and ears for rats and mice. Furthermore, the algorithms sets used for the robot that act as a decision maker are also important in making decisions and processing when combined with the sensors. From this, an issue came up in the author's mind questioning about how does the robot functions in an unknown maze and also the urge to explore about the knowledge of this field of study. Besides that, the author also found out that this field of study will help a lot in understanding a real exploration expedition and it will also help others to further understand the complexity and problems of unknown exploration using a robot.

1.2 Problem Statement

Autonomous robot navigation can encounter a lot of problems. The problems can range from whether the resources for the robot are adequate to power the robot or not; are the number of input for the robot is enough; or whether the memory allocation for the robot's code is enough. Besides that, there are also problems in navigating in an unknown maze environment. What are the action selections of the robot when it has no experience of what so ever about the maze? Can it manage itself within the maze?

1.2.1 Problem Identification

Explorations nowadays are vast as the need to find natural resources are prioritized by a lot of country. Unfortunately these resources are usually kept by Mother Nature in an environment where it will be surely hard for men to disembark and also unknown to them. So sending robots to these places rather than men are considered more reliable and gives them more advantages in terms of costs and efficiency. But to send a robot to these unknown environments rather than men, other problems will surely come up such as whether the robot would be able to think, act and react like a human intelligently. From here, we can see there will be problems in making a machine do what a normal person can do in exploring an unknown environment.

1.2.2 Significant of the Project

Using a robot to solve a line maze autonomously will help a lot in understanding on how a real robot will react in an unknown environment. People who dwell in this area of research will have a clearer view of what to expect and clearer understanding about what are the possibilities and problems that they could and would encounter during a real exploration. Besides that, this project also helps people to understand about artificial intelligence and computer sciences.

1.3 Objectives:

1.3.1 To develop an autonomous robot that can navigate in an unknown maze environment.

1.3.2 To chose an algorithm that can be implemented to solve the maze.

1.3.3 To investigate the potential of using a chosen algorithm on an autonomous robot in an unknown maze environment.

1.3.4 To implement the chosen algorithm for an unknown maze environment.

1.3.5 To test the success rate of the chosen algorithm in an autonomous robot navigation.

1.3.6 To learn on how to transfer information from an algorithm base to a coding base.

1.3.7 To understand the concept of exploration and the reaction of a robot to its unknown environment.

1.4 Scope of Study

The author's main purpose is to choose, develop, and experiment with the chosen algorithm that can be use on an autonomous robot to navigate and solve a linear maze. By doing so, the author is actually using a robot to act based on its environment, thus implementing artificial intelligence and furthermore, through testing and simulation, the author will be able to gather and analyze the tangible and intangible data that are valuable.

1.5 The Relevancy of the Project

This project will show whether the chosen algorithm that is implemented will be able to solve the line maze or not. This is to cater the objectives that had been specified before for the project. And by doing so, the author will assist people to understand about the concept of exploration, for example; what are the problems that people could face during exploration and what are the possibilities in exploring unknown territories. Besides that, this project will also help people to know and understand other field of studies due to the nature of this research that involves other field of studies.

1.6 Feasibility of the Project within the Scope and Time Frame

This project is feasible within the scope and time frame that has been specified because of the nature of the project. Rather having a broad scope, the project only focuses more on a small scale such as fewer requirements and fewer objectives, thus making it possible for the author to proceed with the project.

CHAPTER 2

Literature Review

2.1 A definition on mazes by dictionary.reference.com.

The definition of a **maze** is a confusing network of intercommunicating paths or passages. Besides that, a **maze** can also be defined as any complex system or arrangement that causes bewildered, confusion. (dictionary.reference.com, 2010)[4]

2.2 A definition on algorithm by

2.2.1 In mathematics and computer science, an **algorithm** is an effective method expressed as a finite list of well-defined instructions for calculating a function. **Algorithms** are used for calculation, data processing, and automated reasoning.

(www.wikipedia.com, 2011)[10]

2.2.2 A formula or set of steps for solving a particular problem. To be an **algorithm**, a set of rules must be unambiguous and have a clear stopping point. **Algorithms** can be expressed in any language, from natural languages like English or French to programming languages like FORTRAN.

(www.webopedia.com, 2011)[5]

2.3 A definition on artificial intelligence by

The branch of computer science concerned with making computers behave like humans. The term was coined in 1956 by John McCarthy at the Massachusetts Institute of Technology. **Artificial intelligence** includes:

2.3.1 Games playing: programming computers to play games such as chess and checkers.

2.3.2 Expert systems: programming computers to make decisions in real-life situations (for example, some expert systems help doctors diagnose diseases based on symptoms.

2.3.3 Natural language: programming computers to understand natural human languages

2.3.4 Neural networks: Systems that simulate intelligence by attempting to reproduce the types of physical connections that occur in animal brains

2.3.5 Robotics: programming computers to *see* and *hear* and react to other sensory stimuli

(www.webopedia.com, 2010)[5]

2.4 A definition on artificial intelligence by www.its.bldrdoc.gov.

The capability of a device to perform functions that are normally associated with human intelligence, such as reasoning and optimization through experience. **Artificial intelligence** is the branch of computer that attempts to approximate the results of human reasoning by organizing and manipulating factual and heuristic knowledge. Areas of **AI** activity include expert systems, natural language understanding, speech recognition, vision, and robotics. (www.its.bldrdoc.gov, 1996)[6]

2.5 A definition on autonomy and robots (particularly in a military context) by the US Department of Navy, Office of Naval Research.

2.5.1 **Robot**

A powered machine that senses, thinks (in a deliberative, non-mechanical sense), and acts.

(ethics.calpoly.edu/ONR_report.pdf, 2008)

2.5.2 **Autonomy** (in machines)

The capacity to operate in the real-world environment without any form of external control, once the machine is activated and at least in some areas of operations, for extended periods of time.

(ethics.calpoly.edu/ONR_report.pdf, 2008)

2.6 A Definition of Exploration by Wikipedia – the free encyclopedia

Exploration is the act of searching or travelling a terrain (including space, see space exploration) for the purpose of discovery of resources or information. **Exploration** occurs in all non-sessile animal species, including humans. In human history, its peak is seen during the Age of Discovery for Europe's contact with the rest of the world, and Major explorations after the Age of Discovery for scientific **exploration** in the modern era.

(www.wikipedia.com, 2010)[10]

CHAPTER 3

METHODOLOGY

3.1 Research Methodology

The author has done some studies about robots and found out that there are a lot of things that need to be taken into consideration about choosing which methodology to implement. Due to the fact that there many uncertainties in doing this research especially in implementing the algorithm for the robot by coding, the author has chosen Evolutionary Prototyping. By using the Evolutionary Prototyping, The project is developed by evolving the initial prototype (code) to the final version [7]. The reason the author chose this methodology because:

1. Evolutionary Prototyping allows the project to be flexible; meeting the changing constraints or requirements [8]. This will help a lot in doing this project because basically this project is focused more on testing and simulating; both activities are basically to ensure that the code tallies with the specified algorithm for the project. If there are any requirements that need to be change or altered in the code, it is easier for the author to change and specify new requirements without disturbing what is planned in the methodology.

2. Evolutionary Prototyping allows participant to reflect on lessons learned during the project timeline. That means, the author will learn a lot as soon as the project starts. For example, for an IR sensor usage, the author needs to learn how to program the sensor to work and function on board the robot within the project timeline. By prototyping, the author can code and test whether the code that has been developed can communicate with the sensor.

3.2 Project Activities

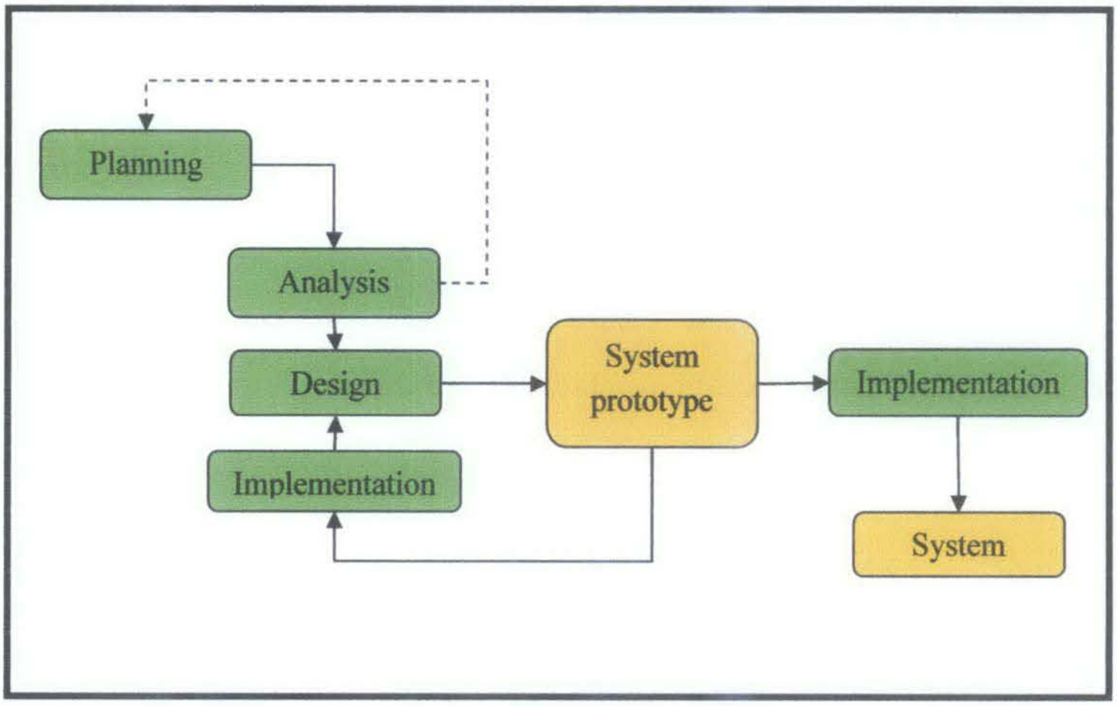


Figure 1: Prototyping Model

3.2.1 Planning

The author finds the main reason why this project should be done as well as understanding the requirements of this project. The author also collects as many information possible regarding the project and clarify what are the main objectives in doing this project. It is necessary for the author to be aware of the different field of studies that need to be accounted in doing this research; for example artificial intelligence (1 field of study) and electrical and electronics (1 field of study). It is also necessary for the author to understand the project so that the author will have a clearer view on what he is going to do afterwards.

3.2.2 Analysis

The author starts to analyze the information gathered. The analysis of the information gathered will lead the author to what should be done and should not be done for this project. Besides that, by analysis it will also help the author to predict and identify possible problems that will and can occur along the development of the project. Furthermore, by analysis, the author will be able to specify different types of tests that should be done to determine the success rate of the robot solving the maze. If anything goes out of plan in the analysis phase, the author can still go back to the planning phase and plan back.

3.2.3 Design

The author specifies a lot of designs associated with the project. For example, the author needs to design the test cases that will be use to test the success rate of the robot solving the maze and also the design of the algorithm for the robot in code form. Designing should take into account the data that has been gathered and analyzed so that all of the design for the project will be optimized to perfection. And also in this phase

3.2.3.1 System Prototype

The algorithm is coded and tested on the robot. Testing also falls under this stage because this is the suitable stage to code and test the code on the robot. As mentioned before, the author uses evolutionary prototyping, thus this will help the author to refine the coding for the robot from time to time until the code is good enough for formal implementation.

3.2.4 Implementation

The implementation of the refined code is done in this phase. The code then will be tested with a number of tests to determine the performance. The code that has been tested will be implemented to the robot for results and discussions purposes. All of the results will be documented and treated like they are a formal result so that they will be no tempering with the actual result of the project. The results can also be use to determine whether the project is a success or not. If there are any problems with the prototype, the author can still move back to the drawing board and redesign or design anew (for example; error in the code). And this situation will loop on until the targeted result has been achieved by the author or the resources for the project have gone out.

3.3 Gantt Chart

Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Selection of Project Topic	x	x													
Preliminary Research Work		x	x	x											
Submission of Preliminary Report				x											
Seminar 1 (optional)					x	x	x								
Project Work					x	x	x								
Submission of Progress Report								x							
Seminar 2 (compulsory)									x	x	x	x			
Project work continues								x	x	x	x	x			
Submission of Interim Report Final Draft													x		
Oral Presentation														x	

Table 1: Timeline for FYP I

Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Project work continues	x	x	x	x	x	x	x	x							
Submission of Progress Report								x							
Project Work Continues								x	x	x	x	x			
Pre-EDX											x				
Submission of Draft Report												x			
Submission of Dissertation (soft bound)													x		
Submission of Technical Paper													x		
Oral Presentation														x	
Submission of Project Dissertation (Hard Bound)															x

Table 2: Timeline for FYP II

3.4 Tools

3.4.1 Hardware:

Laptops or desktops that support the USB technology, 4 or more AAA batteries, AVR ISP programmer with 6 pin connector, and Pololu 3pi Robot (a robot created by POLOLU Robotics and Electronics), white card board, 3/4" black electrical tape to create lines for your robot to follow, a pencil and other drawing tools such as a ruler/straightedge.

3.4.2 Software:

WinAVR: a free open source suite of development tools for the AVR family of microcontrollers, including the GNU GCC compiler for C or C++.

AVR Studio: Atmel free integrated development environment (IDE) that natively works with WinAVR's free GCC C or C++ compiler. AVR Studios include AVR ISP software that will let you upload your programs to the 3pi.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Findings

Under observations, research, comparison, and reviews of data from other journals and websites, the author had found a lot regarding the project that should be taken into account. The critical parts in this project are:

4.1.1 The Maze

As the author has a time frame that needs to be taken into consideration as a constraint, the author has decided to use a linear maze rather than any other mazes. A linear maze is a maze where all of the walls inside the maze are connected to the maze wall and there are no free-floating walls inside the maze. There are no loops in a linear maze, where if the robot comes back to its starting place, it will use the same path that it used before. Below is an example picture of a linear maze:

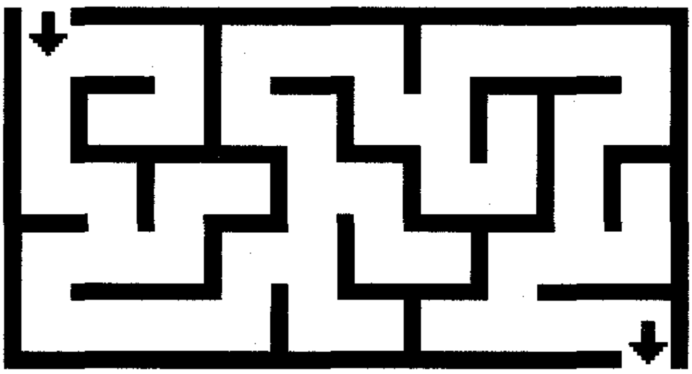


Figure 2: A linear Maze

4.1.2 The Robot

The robot that will be use as a prototype in this project should take into account about these 4 critical things:

1. **The chassis**
2. **The motor**
3. **Robot dynamics**
4. **The sensors**

4.1.2.1 The Chassis

The chassis of the robot should be suitable with the maze that is built. The size of the chassis should not be too big that it disturbs the movement of the robot and it should not be too small that it makes mounting the batteries, motors, controllers and sensors on the robot difficult. Furthermore, the chassis should also be strong and light to make testing and handling the robot much easier.

4.1.2.2 The Motor

The motor of the robot should adequate the weight of the robot. This is to ensure that the speed of the robot is optimized and there are no problems with the robot's movement.

4.1.2.3 Robot Dynamics

There are many things that fall under robot dynamics, for example, how fast could the robot go, how much acceleration and torque is enough to make the robot moves and also how fast the robot can take turns at corners. All of this factors if it is not taken into account can cause failures when running and testing the robot. One factor for example is the friction between the robots' tyres and the surface of the floor. If the surface is too slippery and probably will make the robot slip, a wrong input will probably be taken or no input taken by the sensors and this will generate a wrong output and temper with the testing results.

4.1.2.4 The Sensors

The robot will need sensors that can tell the robot about itself and its environment. This is to ensure that the robot receive sufficient input to process it as an output.

As all the points mentioned above are critical to this project success, the author will be using a robot that is manufactured by Pololu Robotics and Electronics. Rather than manufacturing a robot from scratch, the author will buy a readymade robot so that the author can focus more on the algorithm and programming. Problems regarding the robot can be less as the robot will probably adequate all the critical points. But the author will still take precautions though the robot seems to be adequate enough.

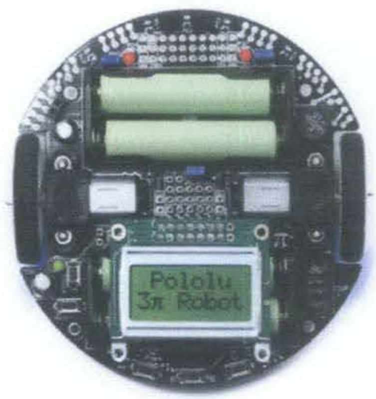


Figure 3: The upper view of the Pololu 3pi Robot

Figure 4: Pololu 3pi robot on a 3/4" black line.

The following are only the important specifications in using the Pololu 3pi robot. Other non-important specifications are not listed but can be use an included into the project (e.g. LEDs, piezo buzzer, and user potentiometer).

- 2 30:1 Micro Metal Gear motors
- Removable 8X2 character LCD
- On/Off power button
- Reset button
- User pushbuttons (3 buttons)
- ISP programming connector pin
- Light weight plastic ball caster
- 5 Integrated QTR-RC reflectance sensors

All of this components are connected to a user programmable AVR microcontroller (ATmega328 microcontroller running at 20 MHz) that measures approximately 3.7 inches (9.5 cm) in diameter and weighs 2.9 oz (83 g) without batteries.

4.1.3 The Algorithm

The algorithm that is chosen in this project is the Wall Follower Algorithm. Since the walls in the maze is connected and the objective of this research is for effectiveness, by using the wall following algorithm, the robot is guaranteed to visit all parts of the maze until it reaches the goal. If it does not reach the goal, it will come back to its starting point using the same path travelled. It is also easier and takes less more time to code.

4.2 Data Gathering / Data Analysis

4.2.1 The Line Course and Line Maze

The Line Course

Due to testing reasons, the author needs to develop a line course first in order to test the robot whether it can manoeuvre (simple line following, avoid obstacles) as what has been specified in the code. This will also help the author to learn key factors on how to write the code and makes the robot understands it. For example; delays in robot programming, motor speed.

The Line Maze

The author has decided to use a linear maze rather than any other mazes. A linear maze is a maze where all of the walls inside the maze are connected to the maze wall and there are no free-floating walls inside the maze. There are no loops in a linear maze, where if the robot comes back to its starting place, it will use the same path that it used before.

Building the line course and the line maze:

- Sketch the course on a piece of paper first to help plan for the building of the real course. The precise dimensions are included in the sketching.
- Pencil the course of the card board by following the specifications in the sketching.
- Stretch the electrical tape along the pencil line drawn on the card board.

Figure 5: A line course example

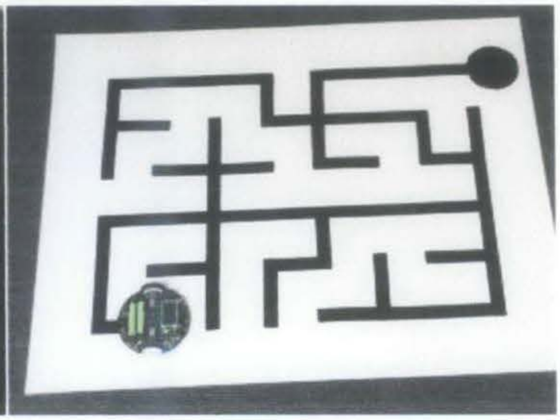


Figure 6: A line maze example

4.2.2 Wall Following Algorithm

The wall following algorithm is a simple algorithm that can be use in solving a simple maze. Though the drawback of using a wall following algorithm is it is impossible for it to solve complex mazes but as mentioned before, the author is going to use a linear maze; thus using the wall following algorithm will not be a problem. Besides that, based on PSU Robotics, (PSU stands for

Pennsylvania State University) the wall following algorithm is easier and takes less time to code.

The algorithm is as followed:

1. Move forward
2. Check for collisions on the front and both sides of the robot
3. If the preferred turning side is open (left)
   3.1. Turn 90 degrees towards that side
   3.2. Go to step 1
4. If the front is blocked
   4.1. Check which side is open (left or right)
   4.2. If the preferred turning side is open (left)
      4.2.1. Turn 90 degrees towards that side
      4.2.2. Go to step 1
   4.3. If the preferred turning side is blocked
      4.3.1. Turn 90 degrees towards the opposite preferred side
      4.3.2. Go to step 1
   4.4. If both sides are blocked
      4.4.1. Turn 180 degrees
      4.4.2. Go to step 1
5. If the goal's requirement is fulfill
   5.1. Stop

The algorithm in flow chart:

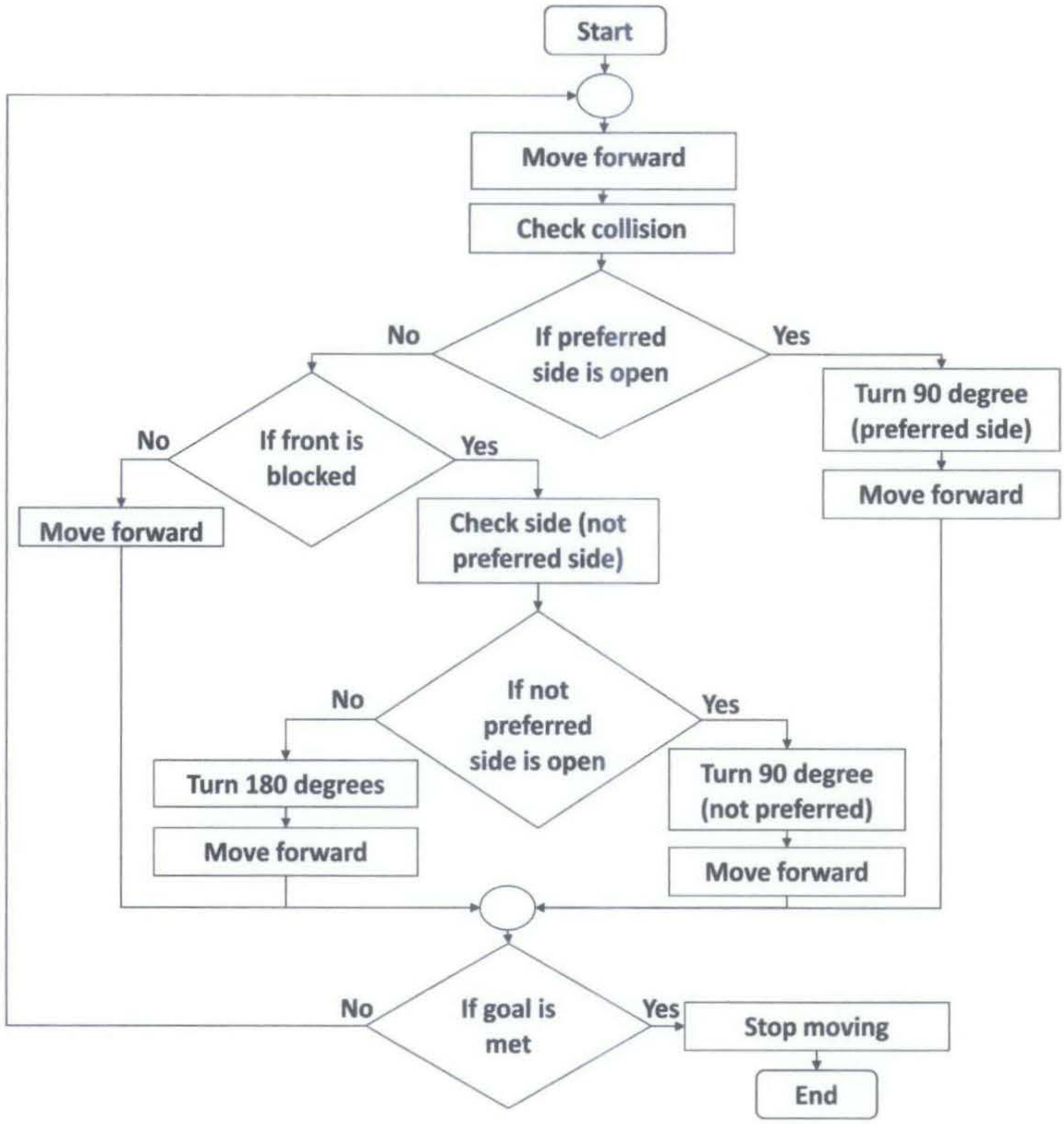
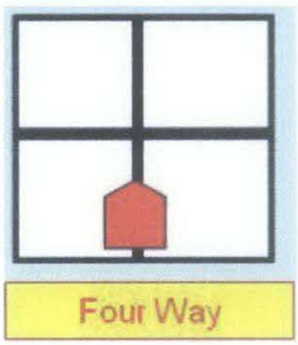
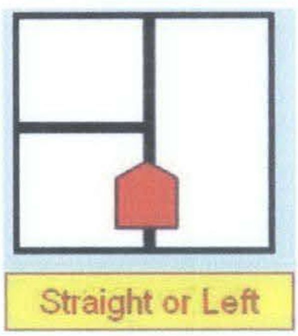
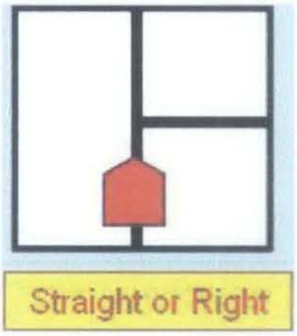
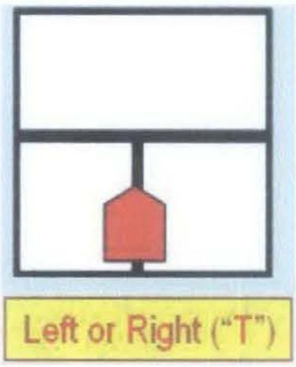
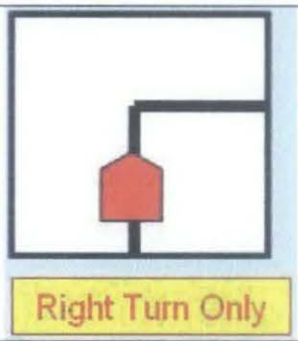
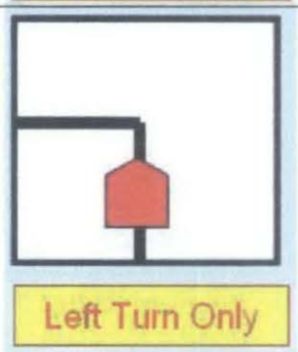


Figure 7: The Wall Following Algorithm in flow chart form

4.2.3 Probability Sets

The robot will navigate itself in the maze and there are a lot of conditions that the robot will encounter within it. These are the conditions and the actions of the robot that are predetermined by the author. In this example, the preferred turning side for this robot is left. All conditions are already taken into consideration and the algorithm specified should be able to cater for all the condition in a linear maze.

No	Conditions	Description
1.		**Condition**: All of the pathways; front and both sides are open. **Action**: The robot will turn 90 degrees to the preferred turning side (left) and move forward to explore the left junction first rather than going straight or turning right.
2.		**Condition**: There is a pathway open at the left side of the robot. **Action**: The preferred turning side is open; the robot will turn 90 degrees to the preferred turning side (left) and move forward to explore the left junction first rather than going forward along the straight line.

3.		**Condition**: There is a pathway open on the right side of the robot. **Action**: Though there is pathway open, the open pathway is not the preferred turning side. The robot will go along the straight line first rather than turning right.
4.		**Condition**: The front is blocked but both sides are open. **Action**: The preferred turning side is open; the robot will turn 90 degrees to the preferred turning side (left) and move forward to explore the left junction first rather than turning right.
5.		**Condition**: The front and the left side is blocked **Action**: The robot will turn 90 degrees to the right (opposite preferred turning side) and follow along the line as that is the only pathway open.
6.		**Condition**: The front and the right side is blocked **Action**: The robot will turn 90 degrees to the preferred turning side (left) and follow along the line as that is the only pathway open.

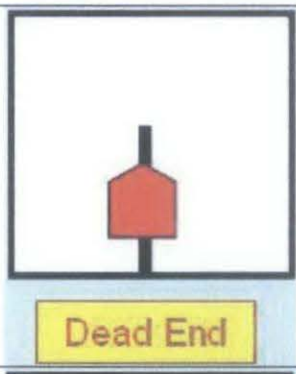
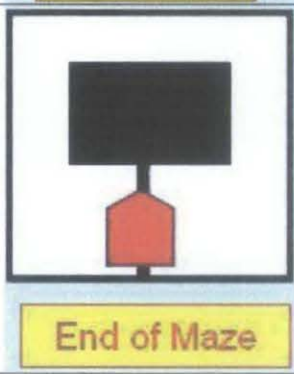
8.		**Condition**: All of the pathways in front and the sides of the robot are blocked **Action**: The robot will turn 180 degrees move forward (equivalent to reverse).
9.		**Condition:** All of the pathways are open, but at the same time all of the sensors detect a line. **Action**: The robot will stop as this will be a sign to the robot that it already reaches its goal.

Table 3: The conditions in the maze and the actions that will be taken by the robot

4.2.4 How the Robot Operates

The robot uses 5 sensors as its input. These sensors will look directly down on the track and then be read by the program to determine the correct next action. When the sensor detects black it will become 1 and if it detects white it will become 0. Using a 1 to mean "Sensor sees black" and 0 to mean "Sensor sees white", a robot travelling along the black line will produce several patterns. For example:

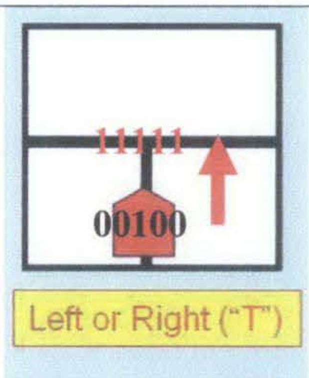
	The initial detection reading of the robot at the moment is **00100**. When the robot moves forward and reaches the "Left or Right" or T-junction, the sensors will detect as **11111**

Table 4: An example of the sensors detection numbers

With five sensors that can each be a one or a zero, there are 2^5 or 32 possible combinations. We will be walking through many of these, but also be aware some of these are impossible or highly unlikely in the mazes described here. For example, you would not expect to see these particular combinations in a line maze: **10101** or **11011** or **10011** and many more. These impossible combinations are disregarder in this project.

The robot, most of the time, will be involved in one of these following behaviors:

1. Following the line, looking for the next intersection.
2. At an intersection, deciding what type of intersection it is.
3. At an intersection, making a turn.

These steps continue looping over and over until the robot senses the end of the maze. This continuation is vital and the robot needs to be taught the correct behavior depending on the type of turn or intersection it encounters every time. The following are the action selections of the robot respective to the probabilities and sensors readings.

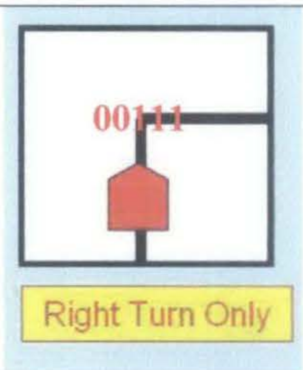
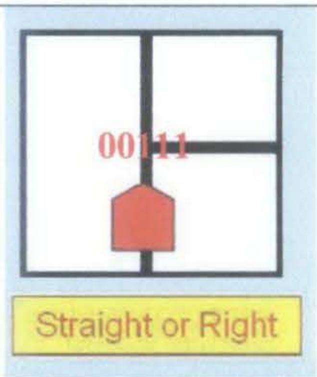
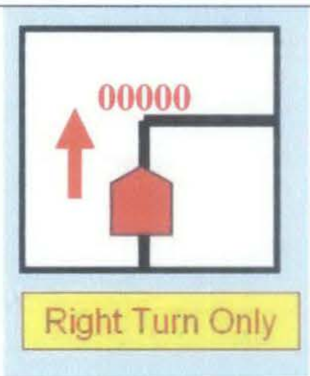
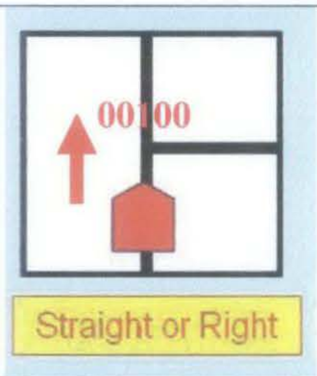
<table>
<tr><th colspan="2">"Right" or "Straight or Right"</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">The initial detection of the sensors at the junctions is 00111. How does the robot differentiate the "Right Turn Only" and "Straight or Right"?</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>The robot will move forward for about 3 cm. If the sensors detect 00000, that means the robot is at a "Right Turn Only".
Action: The robot will turn right.</td><td>The robot will move forward for about 3 cm. If the sensors detect 00100, that means the robot is at a "Straight or Right".
Action: The robot will go straight.</td></tr>
</table>

Table 5: Action Selections in "Right Turn Only" and "Straight and Right"

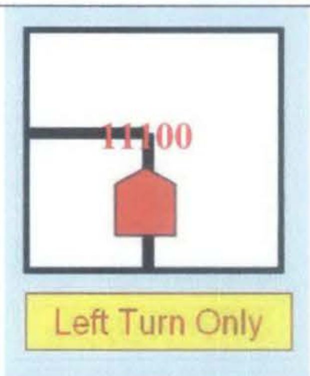
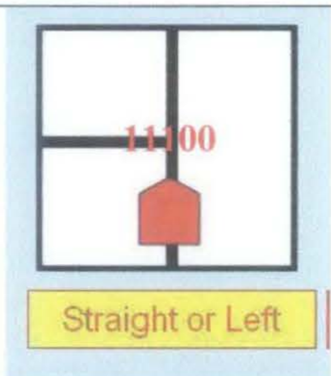
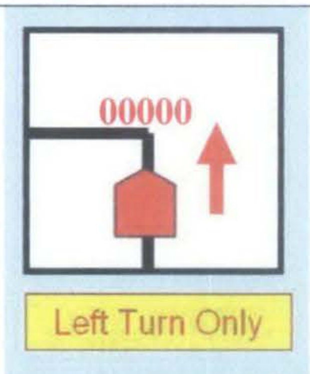
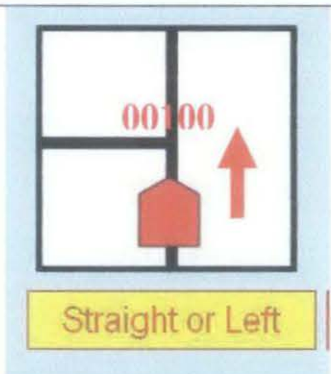
<table>
<tr><th colspan="2">“Left” or “Straight or Left”</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">The initial detection of the sensors at the junction is 11100. How does the robot differentiate the “Left Turn Only” and “Straight or Left”?</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>The robot will move forward for about 3 cm. If the sensors detect 00000, that means the robot is at a “Left Turn Only”.
Action: The robot will turn left.</td><td>The robot will move forward for about 3 cm. If the sensors detect 00100, that means the robot is at a “Straight or Left”.
Action: The robot will turn left.</td></tr>
</table>

Table 6: Action Selections in “Left Turn Only” and “Straight and Left”

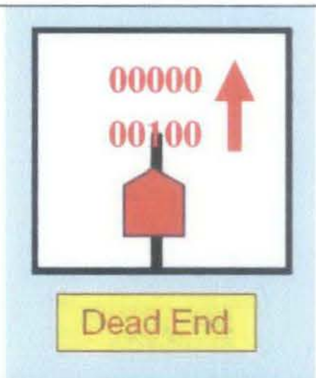
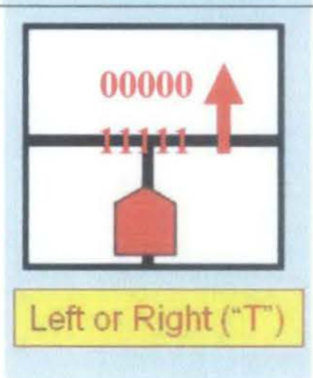
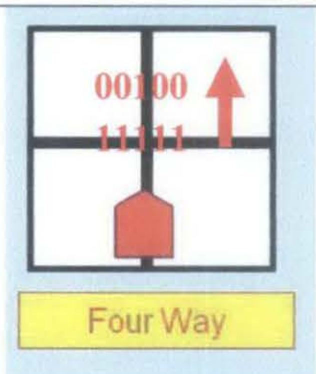
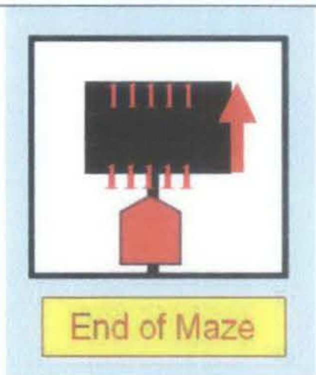
	The initial detection of the sensors at the junction is **00100**. The robot will move forward for about 3 cm. If the sensors detect **00000**, that means the robot is at a "Dead End". **Action:** The robot will make a U-turn
	The initial detection of the sensors at the junction is **11111**. The robot will move forward for about 3 cm. If the sensors detect **00000**, that means the robot is at a "Left or Right" or T-junction. **Action:** The robot will turn left
	The initial detection of the sensors at the junction is **11111**. The robot will move forward for about 3 cm. If the sensors detect **00100**, that means the robot is at a "Four Way" or crossroads. **Action:** The robot will turn left
	The initial detection of the sensors at the junction is **11111**. The robot will move forward for about 3 cm. If the sensors detect **11111**, that means the robot is at a "End of Maze" or goal. **Action:** The robot will stop

Table 7: Action Selections in "Dead End", "Left or Right", "Four Way", and "End of Maze"

4.2.5 Performance Requirements

These are the requirements that should be taken into consideration in implementing the Wall Following Algorithm on the robot.

- Detection accuracy; how accurate can the sensors on the robot detect the input from the environment (for example: detect the black lines).
- Decision Accuracy: how accurate can the robot make a decision based the input taken (when to move; based on the input).
- Decision Time (time taken for the robot to make a decision); how long does it takes to make a decision based on an input.
- Memory Allocation; how optimize can the memory be use to store programming data in a single PIC (use less memory or more memory to use the same type of function)

All of these requirements will determine the performance and accuracy of the robot. The performance and accuracy of the robot will be further tested in the test sets that had been develop to determine the performance and accuracy of the robot.

4.3 Experimentation/Modelling

4.3.1 Implementation

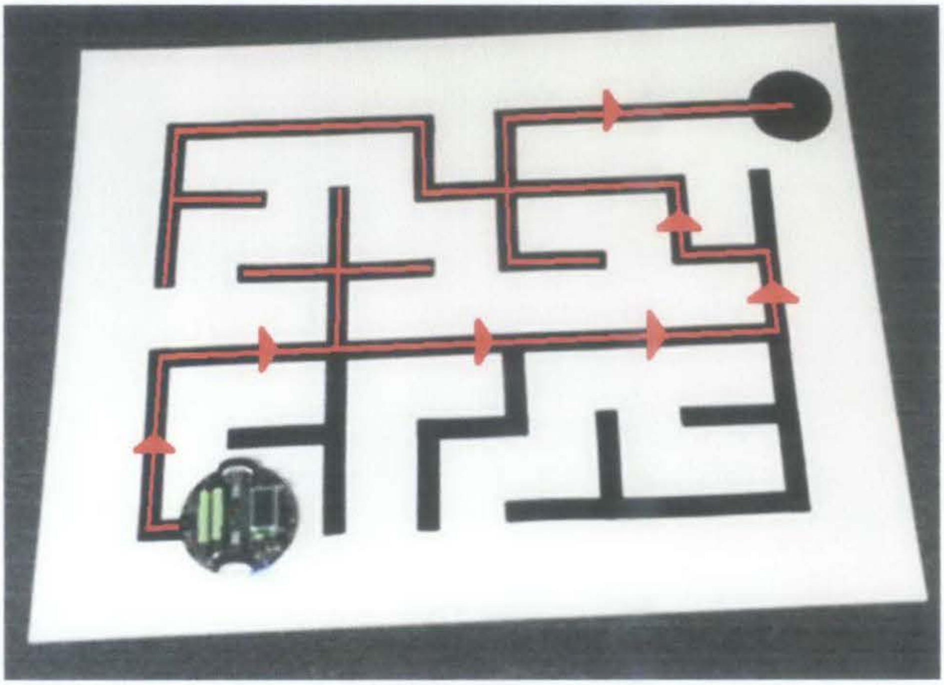


Figure 8: The pathway taken by the robot

The red coloured lines in the maze show the pathway taken by the robot in the specified maze. The position of the robot at the moment is the starting position of the robot. The round black spot in the maze has been specified as the goal.

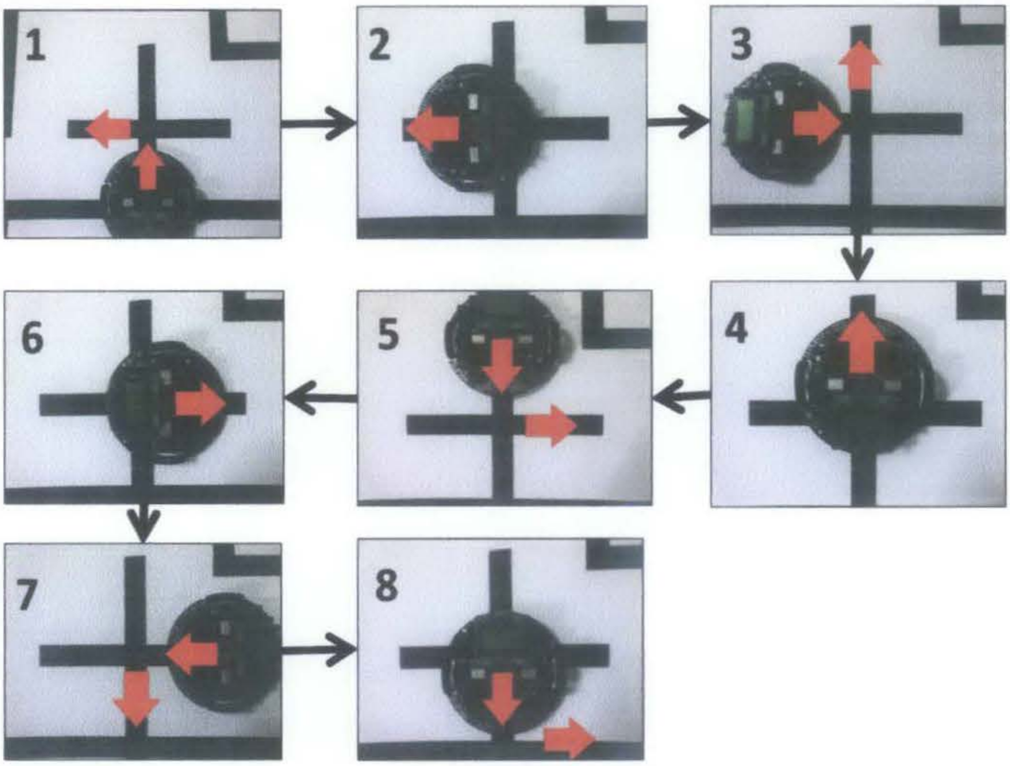


Figure 9: the robot entering the crossroad and exiting it

The figures numbered 1 to 8 above shows the actions of the robot at a crossroads. The descriptions are as followed:

1. The robot enters the crossroad and turns left at the crossroads.
2. The robot enters the junction and goes forward.
3. The robot makes a U-turn and enters the next junction by turning left.
4. The robot enters the junction and goes forward.
5. The robot makes a U-turn and enters the next junction by turning left.
6. The robot enters the junction and goes forward.
7. The robot makes a U-turn and enters the next junction by turning left.
8. The robot leaves the crossroads after exploring all the junctions in the crossroads.

4.3.2 Test and Acceptance Requirements Models

4 tests had been run in order to test the accuracy and the performance of the robot. The tests are as followed:

Test 1	Fault Injection Testing
Input	Use a low power battery to test whether the lack of resources will affect the robot's performance; input, decision making, output.
Case	The lack of resources could disturb the performance of the hardware.
Output	The output is recorded so that the data could be analyze and help us conclude whether the lack of resources will affect the robot's performance.

Table 8: Fault Injection Testing

Test 2	Non-functional testing
Input	The robot is tested on different surfaces, to see whether it will affect its input readings, decision making or its output.
Case	Introducing different surfaces to the robot can test whether it will face any problems if there are any unwanted objects in the maze that can disturb the performance of the robot. **For example;** smooth surfaces may cause the robot to move more than specified in the code. Rather than a 90 degree turn, it can be a 100+ degree turn.
Output	The output for all of the surfaces should be recorded so that any problems regarding the maze used by the robot can be trace.

Table 9: Non-Functional Testing

Test 3	Integration Testing
Input	The modules in the code are integrated 1 by 1 from top to bottom to test whether the module is working or not.
Case	Testing whether the modules specified can work together or not in the code.
Output	By assigning counters, passing values and by visual evidence, we can see whether the modules in the code are integrated to one another or not.

Table 10: Integration Testing

Test 4	Fault Injection Testing
Input	Lines that have different colours • Blue line • Red line • Yellow line
Case	Introducing different coloured lines will show if the sensors can differentiate between the actual colour for the line input (black line) or not.
Output	**Black lines** – follow the line as usual and act according the conditions predetermined. **Other coloured lines** – move forward until the robot can receive the actual input specified; or the robot run out of resources; or the robot is turn off.

Table 11: Fault Injection Testing

4.3.3 Tests Results

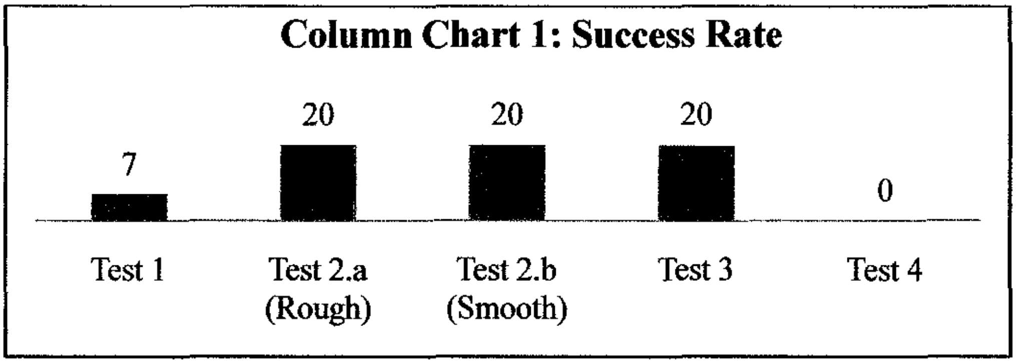


Figure 10: Column Chart 1: Success Rate

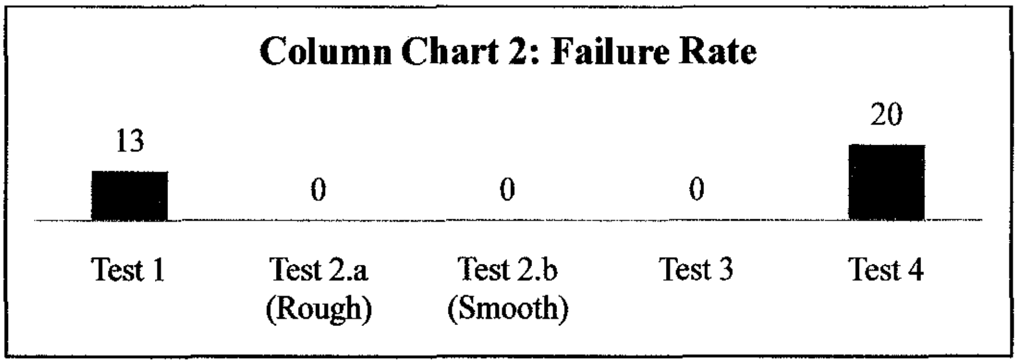


Figure 11: Column Chart 2: Failure Rate

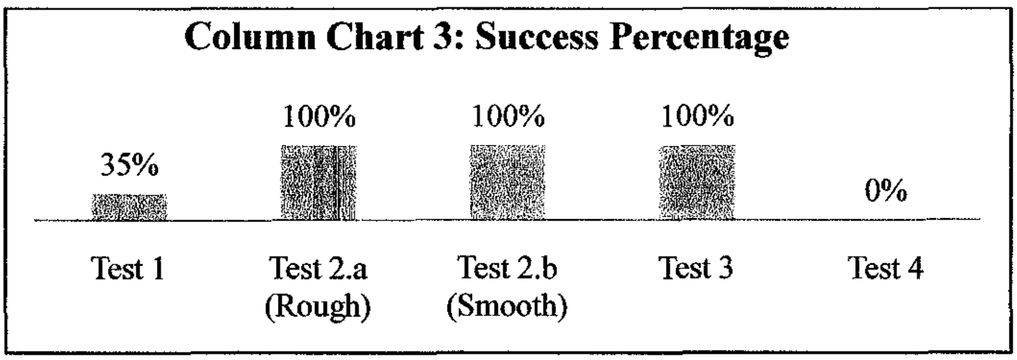


Figure 12: Column Chart 3: Success Percentage

4.3.4 Result Discussions

Test 1 – Test 1 has a success rate of 7/20 and a failure rate of 13/20. This is because at the 12th run of the robot, the robot stops as there no more power or resources to power the robot to move. Thus this test proves that the lack of resources in the robot will affect the performance and accuracy of the robot.

Test 2 – Test 2 for **a** and **b** has a success rate of 20/20 and a failure rate of 0/20. This is because the different surfaces that are introduced to the robot does not effect the performance and accuracy of the robot. The rubber tyres on the robot works perfectly on both of the surfaces and does not affect the readings taken by the sensors, thus leading to no misread of the sensors and no wrong decision making.

Test 3 – Test 3 has a success rate of 20/20 and a failure rate of 0/20. This is because there are no problems in the integration of the code. Though there are problems in the develpoment of the code. The test is solely for the code after it has already finished.

Test 4 – Test 4 has a success rate of 0/20 and a failure rate of 20/20. This is because the robot cannot differentiate the different coloured lines (e.g. yellow lines, red lines) with the original input specified, the black lines. The true problem of this test can be encountered in the coding by specifying the range of the input. In the code, the author should specify specifictly the range of the preferred input rather taking a random range of the sensors of 0 and 1.

CHAPTER 5

CONCLUSIONS AND RECOMENDATIONS

5.1 Relevancy to the Objective.

As a conclusion, though the robot reaches its goal in implementing the Wall Following Algorithm and solving the line maze as specified, there are still many things that should be taken into consideration in implementing this project in terms of performance and accuracy. All of this can be noted by the tests that had been run in the project. Fortunately, by running these tests, the author manages to obtain valuable data that can be used for future work or continuation.

5.2 Suggested Future Work for Expansion and Continuation.

The author would suggest in furthering this project, learning and memorization could be implemented in the algorithm and the code. Though the robot manages to maneuver itself in the maze and reaches the goal, the robot still lack in learning capabilities and the ability to remember the path that it had taken. Plus, the robot is only able to maneuver in simple line maze. A complex maze will cause a lot of problems to the robot in terms of decision making. In a nutshell, if this project continues, these are the things that can be taken into consideration and gives a purpose and objectives in furthering the project.

REFERENCES

[1] http://en.wikipedia.org/wiki/Exploration

[2] http://en.wikipedia.org/wiki/Exploration_problem

[3] http://www.wisegeek.com/what-is-a-maze.htm

[4] Dictionary.reference.com

[5] www.webopedia.com

[6] www.its.bldrdoc.gov

[7]http://cslinux.cs.apu.edu/~sliang/Spring2009/CAPS470/Handouts/Prototype%20Pros%20and%20Cons.pdf.

[8] http://www4.ncsu.edu/~jbearp/epramTechReport.pdf.

[9] http://www.micromouseonline.com/book/micromouse-book

[10]Wikipedia, the free encyclopedia at <http://en.wikipedia.org>

Stamenova, Stiliyana, "Solving the Maze: Robot Localization Using the Monte Carlo Localization Algorithm and Shape Context" (2009). *Honors Projects.* Paper 15. http://digitalcommons.macalester.edu/mathcs_honors/15

Patrick Lin, George Bekey, Keith Abney, "Autonomous Military Robotics: Risk, Ethics, and Design" (2008). ethics.calpoly.edu/ONR_report.pdf.

The Pennsylvania State University – Robotics Club: Making the World Autonomous Since 2004 at <http://www.psurobotics.org/wiki/index.php?title=Main_Page>

Welcome to Cytron Technologies – Robot, Head to Toe: PIC training Malaysia, PIC Microcontroller<http://www.cytron.com.my/>

Richard T. VannoyII, "Teaching a Robot to Solve a Line Maze" (2009)

Pololu Corporation, "Pololu AVR Library Command Reference" (2001 – 2010)

Jan Malášek, "Understanding battery capacity: Ah is not A" (2010) at <http://www.pololu.com/blog/2/understanding-battery-capacity-ah-is-not-a>

Pololu Corporation, "Building Line Following and Line Maze Courses" (2001 – 2009)

Pololu Corporation, "Sample Project: 3pi Wall Follower" (2001–2010)

Pololu Corporation, "Pololu 3pi Robot User's Guide" (2001–2010)

Pololu Corporation, "Pololu AVR C/C++ Library User's Guide" (2001 – 2010)

IEEE, "Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 18, No. 4" (2010)