

User Manual

SEP Group 1

Name

Signature

Lee Crosby

Craig Eales

Andreanna Humphrey

Vivian Lee

Zhiyang Ong

Version History

Version	Release Date	Edited By	Description	Changes since last
1	27 th October, 2003	Zhiyang Ong	Initial Version	None
2				
3		Craig Eales	Troubleshooting	Some modifications to the trouble shooting section.

Table of Contents

List of Figures

List of Tables

1.0 Preface

2.0 Getting Started

3.0 Mazebot User Interaction

4.0 Calibrating the Robot

5.0 Getting the Robot to Explore a Maze

6.0 Getting the Robot to Traverse the Maze

7.0 Trouble Shooting

8.0 Obtaining Technical Assistance

9.0 Reference Information

1.0 Preface

Congratulations on your new Mazebot robotic maze solver system!

The Mazebot system allows you to and the Mazebot robot to explore a wide range of mazes using a Linux based desktop computer. You can watch as the robot explores the mazes and they appear on the screen before your eyes. You can tell the robot where to go on the maze with just a click, be it the pizza bar, the pub or maybe the fuel station, and the robot will get there as quickly as it can.

The Mazebot system is designed to be easy to use. You do not need any training with computers or robots as the system will guide you through each stage of operation. You don't have to worry about getting into any trouble either as the system has an emergency stop if you can't keep up with the Mazebot's pace!

This user manual has been written to help you to start using the Mazebot mapper as soon as possible. Read the section on 'Getting started' if you want to jump right in, but if you want to learn more about the robot and the system read further about its capabilities. The manual also provides support if things don't seem to be doing as you would expect, so you can get back to exploring as soon as possible.

You can obtain more information about Mazebot by contacting the developers of the Mazebot software. Details about the limitations of the software and hardware involved as well as the physical environment the system is operated in can be found in the Software Requirements Specification (SRS), which you can get a soft copy of [Group1SEPG03a]. See Section 8.0 for more details about contact information.

1.1 Disclaimer

This manual and any examples contained herein are provided as-is and are subject to change without notice. Software Engineering and Project Group 01, 2003, does not express or imply warranty of any kind with regard to this manual and shall not be liable for any errors, accidents and/or consequential damage in connection to the performance or use of this manual and the examples herein.

Editors' Note

The Mazebot software is the intellectual property of Software Engineering and Project Group 01, 2003. Unauthorized reproduction, adaptation and/or translation of the Mazebot program or its associated documents are strictly prohibited.

The editor's have adopted the format of this User Manual from other user manuals referenced in Section 9.0 [Microsoft, Dell, CASIO].

CAUTION:

Please take due care when using electronic devices as intentional misuse or carelessness may lead to injury.

Also note that the motors on the robot may become hot to touch if used for long durations.

2.0 Getting Started

The Mazebot software system may be distributed in one of two forms. It may be stored as source files in the CVS repository on the Computer Science servers at the University of Adelaide, or as compiled programs in a directory hierarchy. If the software is stored in the CVS it must first be 'checked out' and then compiled into executable code before it can be run. This process is described below.

1. If the software is in the repository it must first be checked out. To do this you must first be logged onto the CS server and have access to the CVS. To create a working version, create a new directory for the program by using the unix/linux command `mkdir name`. Name is the name of the directory you want to create and store your Mazebot software in. Enter this directory and then checkout the release version of the Mazebot software as follows.
2. Type `mkdir Mazebot` then `cd Mazebot`
3. Type `cvs checkout project -r "Final" project`
4. This checks out the "Final" revision of the module 'project'. This project module contains all of the source code for the Mazebot system.
5. The java code is compiled using the Makefile. This is done by typing `make Mazebot`. *Before you type this command, ensure that the robot is connected to the computer via the serial link and the heart-beat led is blinking.*
6. Now there will be directories of Java class files that can be executed with the Java interpreter. Before executing the host software, the Makefile will first compile the bootloader program and the robot software.
7. Once compiled the bootloader will then load the robot software onto the robot. This requires that the robot be connected to the host computer via the serial link as previously mentioned. The code will be displayed on the screen as numbers and letters streaming up the screen. This process may take over 30 seconds.
8. Once the code is loaded onto the robot, the robot will display 'SEP—01' on its seven segment display and the Makefile will execute the host software.
9. The Mazebot title screen will appear for a brief moment while the graphical user interface is initialised. It will then disappear, leaving the graphical user interface on the screen.

Congratulations, the Mazebot system is now running on your PC. If you want to jump right in go ahead. You can click the coloured buttons in the button panel and the message panel will instruct you if you get lost. If you get confused at any time don't panic, just press the emergency stop button (a big red stop sign!) and then reset the robot by pressing the reset button on the robot body.

To learn more about how the Mazebot User Interface works, read on!

3.0 Mazebot User Interaction

The interactions between you and the Mazebot software are indicated as follows:

3.1 Summary of Interactions between the User and the Mazebot Software

Figure 3.1 shows a picture indicating the interactions between you and the Mazebot software through the buttons available on the Graphical User Interface (GUI).

Through the buttons available on the GUI, the user is able to manipulate the robot. You must calibrate the robot prior to any other operation since each small robot differ in terms of the ratio of left and right steps needed to be taken to keep the robot moving linearly.

Once the robot has been calibrated, the exploration of the maze can begin; after which, map exploration will be enabled. You can stop or pause the operation of the robot during map exploration and traversal. If the map exploration or traversal is paused, you may resume the paused operation.

You may make use of the display options available on the GUI to enlarge or zoom out map displayed on the GUI. You may also drag the map along the screen.

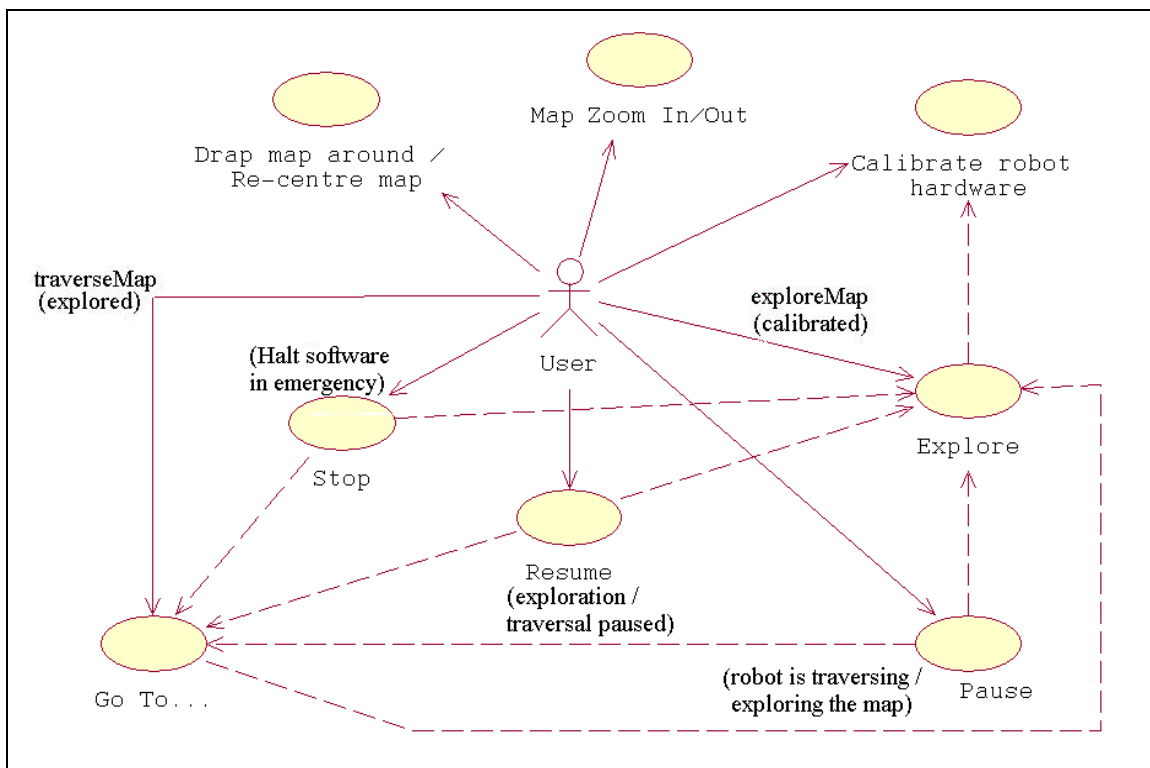


Figure 3.1 – A Use Case Diagram of the Mazebot software using UML notation.

- 3.2** Once you have run the Mazebot software, a GUI will appear on the monitor screen to allow you to interact with the system.
- 3.3** There are seven buttons on the GUI, in addition to the left and right mouse buttons, to provide nine features to you. Each button allows you to manipulate the robot in a fashion clearly labeled on the button. Details of each button are provided in the subsequent three Sections.

4.0 Calibrating the Robot

The procedure for calibrating the small robot is:

- 4.1** Set the robot on a piece of paper containing a black line, laser printed on white paper, with a width of 6 mm and click the “*CALIBRATE*” button on the GUI and the robot will begin to search for the nearest black line.
- 4.2** Note that apart from the “*CALIBRATE*” button, no other button will be enabled; neither will a right nor left click of the mouse bring any effects to the small robot.

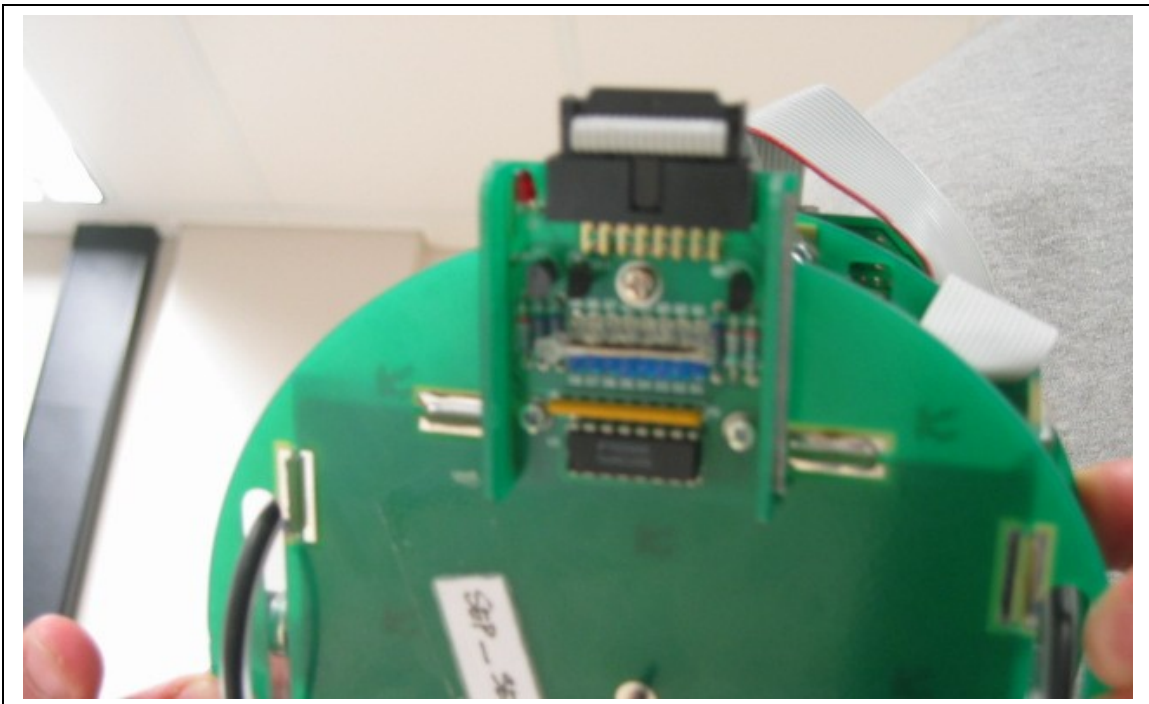


Figure 4.3a A photograph of the robot's sensors

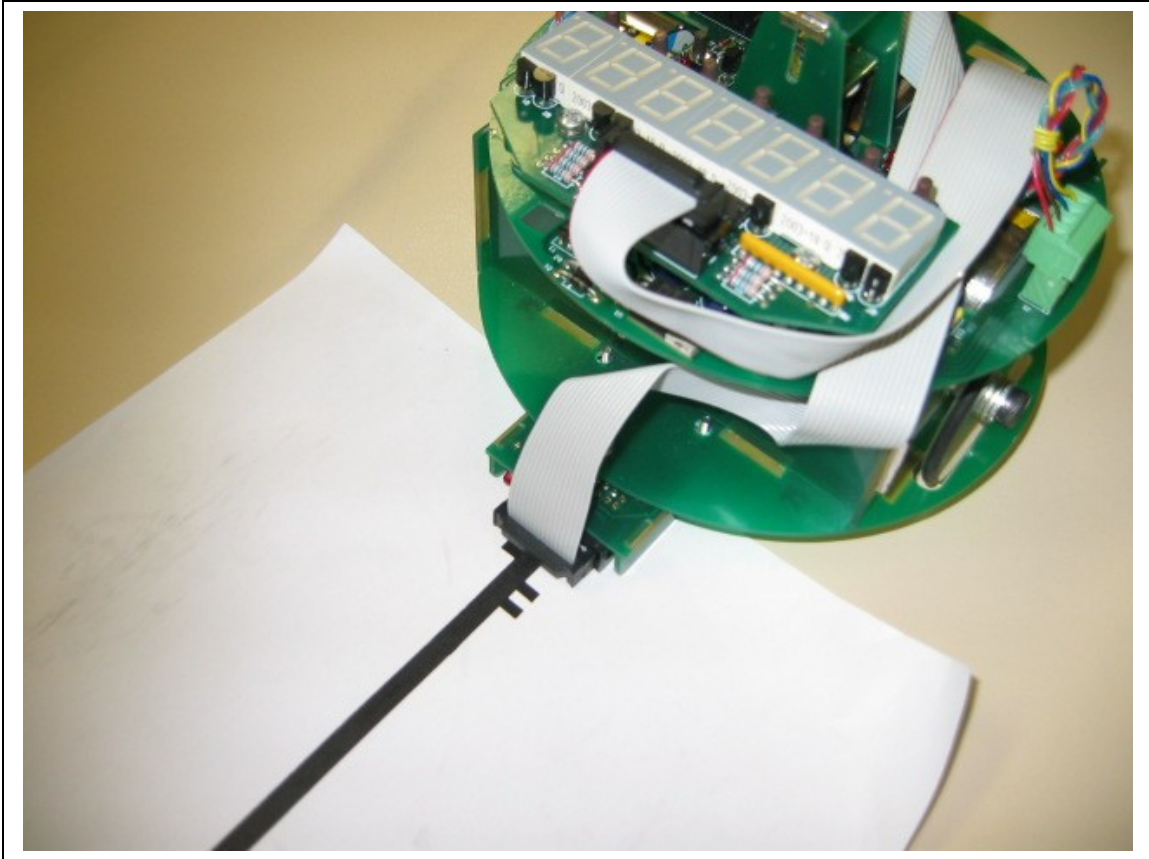


Figure 4.3b A photograph indicating how the robot should be placed on the calibration line.

- 4.3** If a “Pop-up window” appears with a message indicating “Line not found”, place the small robot hardware on the appropriate linear line on the piece of paper.

The robot will orientate itself back to the black line and trundle along the line searching for the start code placed adjacent to the line, which represents roads of the city. Once a start code has been found, the robot will count the number of steps taken till it reaches the end of the line, indicated by the end code that is also found adjacent to the line.

- 4.4** A “Pop-up window” appears to inform the user that the robot has been calibrated.
- 4.5** The user can proceed to exploring the maze and all other buttons are enabled with the exception of the “Go To...”, “ZOOM IN”, “ZOOM OUT” as well as left and right clicks of the mouse. This is because the maze has not been mapped onto the GUI and the map traversal as well as manipulation functions should be disabled as a result at this point.

5.0 Getting the Robot to Explore a Maze

This is the procedure to get the robot to explore a given maze.

- 5.1 Click the “*EXPLORE*” button on the maze to instruct the robot to map the maze.
- 5.2 As the robot explores the maze, you will see a scaled representation of the maze on the screen being drawn to scale real time, through the GUI.

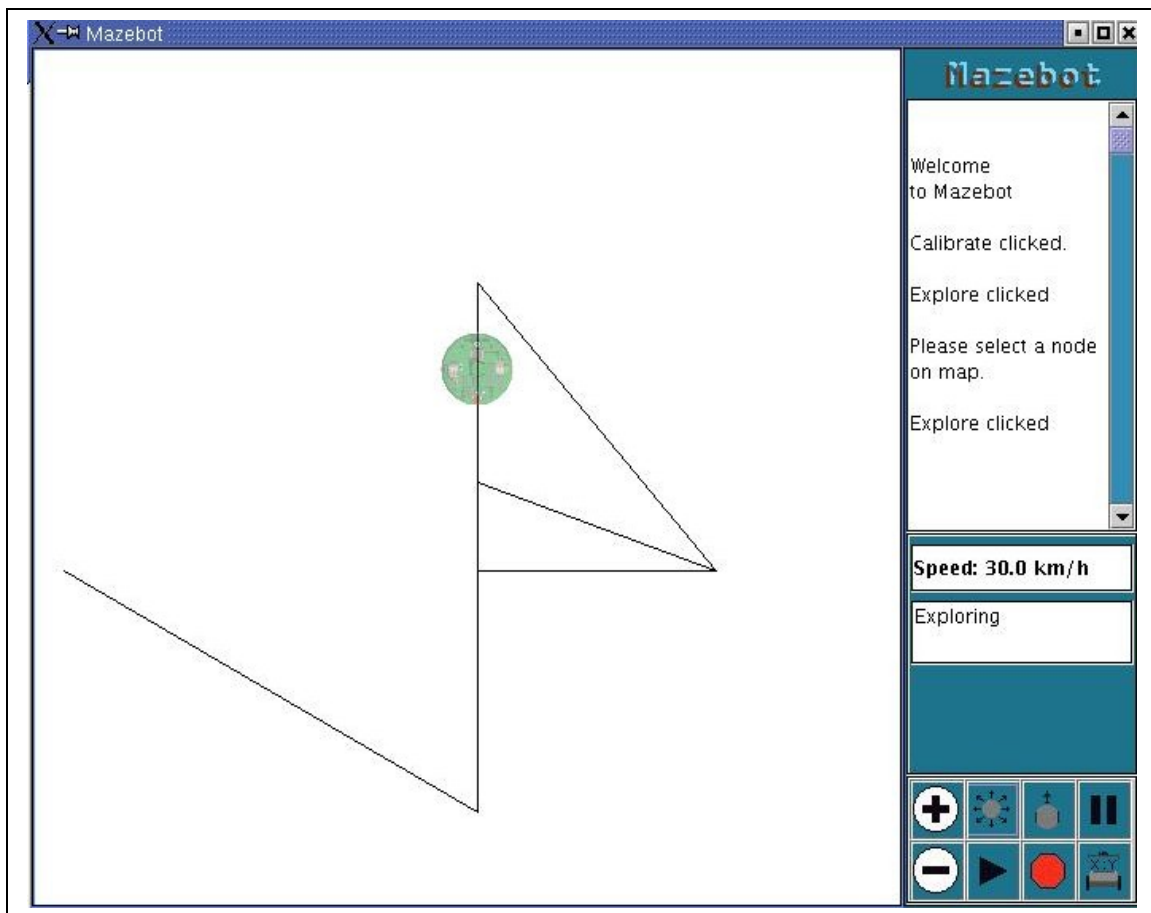


Figure 5.2 A picture of the GUI trundling along a half explored map.

- 5.3 You can pause this operation by clicking on the GUI’s “*PAUSE*” button. The exploration of the maze can be resumed by clicking on the “*RESUME*” button.

Note that resuming this operation and clicking on any button apart from the “*RESUME*” button subsequently after that will cause the exploration of the maze to cease. However, the “*EXPLORE*” button can be clicked again to continue exploration of the maze.

- 5.4** You can stop this operation by clicking on the “*STOP*” button. Note that this button is used in an emergency when the robot malfunctions and control of the robot is lost. This results in having to repeat the procedure for calibration and map exploration of the robot.
- 5.5** Clicking the “*CALIBRATE*” button has the effect of clearing the map from the screen and the robot will have to explore the map again.
- 5.6** Once the exploration of the maze has begun, the buttons as well as the left and right mouse clicks are enabled to allow map manipulation.
- 5.7** During the exploration of the maze, the “*ZOOM IN*” and “*ZOOM OUT*” buttons can be clicked to enlarge or reduce the size of the scaled map representation on the screen.
- 5.8** Click the right mouse button and drag the mouse to drag the map to move the map around on the screen to allow viewing of different areas of the maze.
- 5.9** Click the left mouse button to centre the robot on the middle of the screen and indicate the surrounding areas of the map where it is traversing.
- 5.10** Once the exploration of the map has been completed, the “*Go To...*” button will be enabled and a “Pop-up window” will appear to indicate the completion of map exploration.

6.0 Getting the Robot to Traverse the Maze

Follow this procedure to get the robot to go to a number of locations on the maze.

- 6.1** Click the “*EXPLORE*” button found on the GUI and a “Pop-up” window will appear to request the user to click on any junctions or landmarks (such as pizza joint or pubs) on the maze for the robot to traverse to.

Note that more than one location can be selected. There is a limit of 50 locations that the robot will traverse to in one traversal.

- 6.2** Click on the “OK” button on the “Pop-up window” to complete the selection of locations for the robot to traverse to.
- 6.3** As indicated in the previous Section, the map manipulation features are enabled to allow you to have different views of the map.
- 6.4** Similarly, the “*STOP*”, “*PAUSE*” and “*RESUME*” buttons available to stop or pause the map traversal by the robot or resume map traversing.

7.0 Trouble Shooting

The Mazebot system has been designed to provide the user a reliable and easy to use Maze mapping robot. If you should experience difficulties in the operation of your Mazebot system, please follow the following troubleshooting guide.

Problem:

The program stops and the screen displays an 'AssertionException' message.

Explanation:

The Mazebot system has developed an unrecoverable error due to software bugs.

Solution:

The Mazebot system should not ever generate AssertionExceptions. If this does occur, reset the robot and *then* restart the host software. Also, please contact Zero-One Systems to report the bug so we can find and fix the software errors for future updates.

Problem:

The robot displays 'LOST' on its seven segment display.

Explanation:

The robot sensors have moved away from the maze lines and cannot locate them again. The robot can no longer follow the maze lines to traverse or explore the maze. This may happen if the robot attempts to negotiate mazes that are not supported by the system.

Solution:

The robot must be reset before it can be used to explore or traverse the maze again. The robot should only be used on mazes that are supported by the system as the performance and behaviour of the robot on other mazes may be unpredictable.

Problem:

The robot displays 'LOST' on its seven segment display and the GUI displays the message 'Robot is lost'.

Explanation:

The robot sensors have moved away from the maze lines and cannot locate them again. The robot can no longer follow the maze lines to traverse or explore the maze. This may happen if the robot attempts to negotiate mazes that are not supported by the system.

Solution:

The robot must be reset before it can be used to explore or traverse the maze again. The robot should only be used on mazes that are supported by the system as the performance and behaviour of the robot on other mazes may be unpredictable.

Problem:

The robot suddenly slows down and comes to a halt. It displays 'COMMS' on the seven-segment display and the GUI displays the message 'Communications Timeout'.

Explanation:

The communications link may be suffering from a large amount of noise or the link may be broken. The communications software cannot maintain a link.

Solution:

Close the GUI and then check the connection between the serial cable and the robot. Take it out of the socket and examine it for dirt / dust or build up. Also look for any blackening as this may indicate that the communications socket is burnt out. Check for loose wires or other damage and then reinsert the plug into the socket. Gently wiggle the plug in the socket to check for looseness. Ensure a firm connection and then check the connection to the back of the host computer. Reset the robot and then run the Mazebot GUI again. If the problem repeats it is most likely that the serial connection on the robot or the serial cable is damaged and must be replaced.

Problem:

The robot image on the GUI does not seem to move as the robot on the map and is not drawing the correct map on the screen.

Explanation:

The Mazebot system will accurately display the robot behaviour and the maze on the GUI if the maze is within the supported specifications. Mazes that do not meet the required specifications may confuse the system and cause it to behave unpredictably.

Solution:

For reliable performance, operate the Mazebot system on mazes that meet the defined specifications.

8.0 Obtaining Technical Assistance

Mazebot Software Support Service is available via electronic mail to the developers of the Mazebot software. Technical assistance is available at no cost to the user.

The contact details for Zero-One-Systems can be seen below:

Email: zero-one.systems@cs.adelaide.edu.au

Note that these electronic mail addresses are subject to change. Please contact Zhiyang Ong via electronic mail to reach any of the other Mazebot software developers.

9.0 Reference Information

9.1 Operating Details

The RS-232 cable must be connected to the robot at all times to supply the robot with electrical power as well as allow the robot to communicate with the computer program residing in the desktop.

The lighting conditions must be florescent at 50 Hz.

The maze must be laser printed in black on sheets of white paper, attached using tape.

9.2 Glossary

CVS - Concurrent Version Control System

GUI - Graphical User Interface

Make - The *make* utility automatically determines which pieces of a large program need to be recompiled, and issues commands to recompile them.

Pop-up window - A small window that pops up during the execution of the Mazebot software with a message indicating either a prompt or an error has occurred. An error message will succinctly describe the problem and suggest remedies to the problem.

SEP - Software Engineering and Project

SRS - Software Requirements Specification

System - The system includes the host computer, the serial link and robot. The system also includes the software on the host computer and in the robot. The system does not include the external environment, the user or the maze.

User - The person who operates the system by physically placing the robot on the maze and interacting with the graphical user interface.

UML - Unified Modeling Language

9.3 Reference List

[CASIO] CASIO Computer Co., LTD n.d., *V.P.A.M. fx-992s: User Manual*, MARCO Corporation (M) SDN BHD, Kuala Lumpur

[cvshome.org] CollabNet, Inc 2003, viewed 21 October, 2003. Available: CVS Manual: *Version Management with CVS*: <<http://www.cvshome.org/docs/manual/>>

[Dell] Dell 2002, *Dell Latitude Portable Computers – Dell - Installed Microsoft Windows 2000 Professional Setup Guide*, Dell, Kuala Lumpur

[Group1SEPG03a] Group 1 2003, *Software Requirements Specification (SRS): v3.0*, The University of Adelaide, Adelaide

[Group1SEPG03b] Group 1 2003, *Software Project Management Plan (SPMP): v3.0*, The University of Adelaide, Adelaide

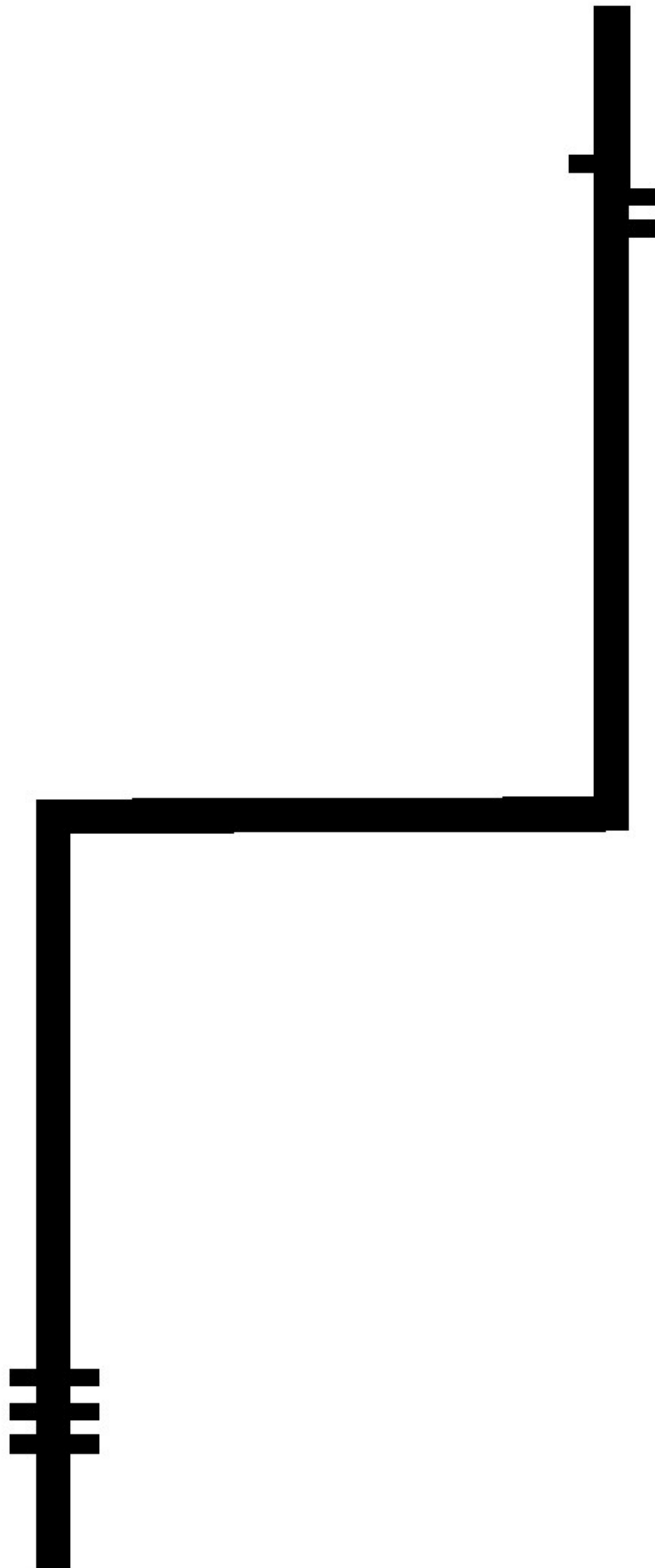
[Group1SEPG03c] Group 1 2003, *Software Design Description (SDD): v3.0*, The University of Adelaide, Adelaide

[Microsoft] Microsoft 2001, *Discovering Microsoft Office XP Standard and Professional*, Version 2002, Microsoft, Singapore

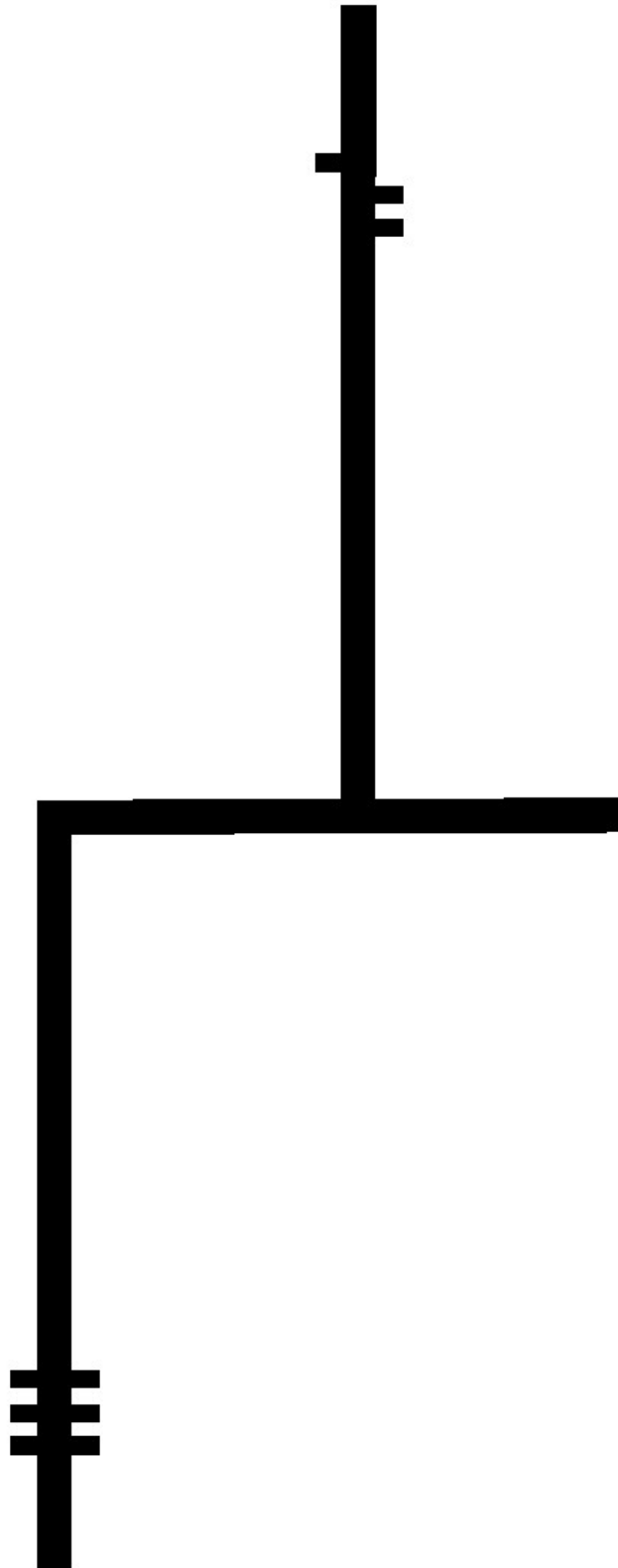
Appendix A – Calibration Line / Sample Maze 1



Appendix B – Sample Maze 2



Appendix C – Sample Maze 3



Appendix D – Sample Maze 4

