

# Table of Contents

## Contents

<b>1</b>	<b>Progress Report</b>	<b>2</b>
1.1	1 <sup>st</sup> Iteration . . . . .	2
1.2	2 <sup>nd</sup> Iteration . . . . .	2
1.3	3 <sup>rd</sup> Iteration . . . . .	2
<b>2</b>	<b>Mechanical Design of MazeBot</b>	<b>3</b>
2.1	First Iteration . . . . .	3
<b>3</b>	<b>Software Design of MazeBot</b>	<b>4</b>
3.1	Variables Used for Representation of Directions . . . . .	4
3.2	Variables Used for Moving Mechanism - Going Forward . . . . .	4
3.3	void goFwdCell() . . . . .	5
3.4	Function Description . . . . .	5
3.5	Flow Chart . . . . .	5
3.6	Function Parameters . . . . .	5
<b>4</b>	<b>Appendix</b>	<b>5</b>

# **1 Progress Report**

## **1.1 1<sup>st</sup> Iteration**

Should be achieved by:

## **1.2 2<sup>nd</sup> Iteration**

Should be achieved by:

## **1.3 3<sup>rd</sup> Iteration**

Should be achieved by:

## 2 Mechanical Design of MazeBot

### 2.1 First Iteration

#### Goals:

- Accurate movement

#### Mechanical Drawing

XOURNAL LOL

#### Observations & Measurements

##### **Structural Integrity of the Drive System**

In order for more accurate movement, we added a high gear ratio. However, since we added the high gear ratio, we are unable to find space to properly secure the left and right drive wheels. When testing, we found one wheel to slipped forward and the other to slipped back when turning which defeats the accuracy of the encoder. Because of this, we are unable to meet our goal of accurate movement.

##### **Robot is too large.**

Since the brick is upright, it is top heavy. We needed two rods in the back and one metal ball in the front in order to balance the robot. The additions of the two rods and one metal ball remove the spacial advantage of having the robot's brick be upright. Even though the dimensions of the robot are within the size of one square, it leaves very little room for error. As such, the robot begins to run into walls after the 3<sup>rd</sup> turn.

##### **Conclusion**

In conclusion, we have decided for our next iteration to have the robot's brick be flat on the ground in order to ensure that we have enough room to properly secure our drive system. By properly securing the drive system we hope to not have to readjust every time we enter a cell.

### 3 Software Design of MazeBot

Our main goal with the software of the mazebot was to create program solved the problem simply and was easy to build upon. Furthermore, we wanted our software to have very few constants that we would need to tested for. For example, in order to move forward one cell, we would need to give the following function the degrees to move each of our drive motors:

```
setMotorTarget(leftMotor, degrees, 75);
```

The degrees needed to move one cell forward could be achieved by constantly testing different values of degrees to achieve the movement to the new cell. However, we chose to calculate the exact degrees that the robot's drive motors would need to move in order to move exactly one cell forward. This approach in contrast to the former has two advantages:

1. It allows us to isolate any problems with moving accurately to a mechanical problem.
2. We would not have an accumulation of error because of us testing incorrectly.

Therefore, we chose to mathematically calculate the degrees that we needed to move the motors rather than testing.

A sketch of the derivation of how many degrees to move forward is shown below:  
Therefore:

```
degrees = (SIZE_OF_ONE_CELL / CIRCUMFERENCE_OF_WHEEL) * DRIVE_GEAR_RATIO * ONE_ROTATION
```

A similar derivation exists for turning the robot 90 degrees:

#### 3.1 Variables Used for Representation of Directions

- The four integer values that represent each direction were declared as a variable

```
#define NORTH 0
#define EAST 1
#define SOUTH 2
#define WEST 3
```

- Structure name cell was declared and it has five parameters: North wall, South wall, East wall, West wall, and Visited (used to tell if the robot has visited the cell or not)

```
typedef struct{
int NWall;
int SWall;
int EWall;
int WWall;
char Visited;
}cell;
```

- 

#### 3.2 Variables Used for Moving Mechanism - Going Forward

Because of the uncertainties caused by each motor, the robot was not moving as straight as it could be. Therefore, one of the motor had to slow down in order to make the robot move exactly straight as wanted.

### **3.3 void goFwdCell()**

- **Example of our code**

```
setMotorTarget(leftMotor, degrees, 75);
```

Briefly explain the algorithm of the maze bot

### **3.4 Function Description**

Explain what each function does

### **3.5 Flow Chart**

Flow chart of the algorithms (I think there is a online thing where u can make flow chart)

### **3.6 Function Parameters**

Yea

## **4 Appendix**

Source of Code with brief comments