

# Robot Tour 2024 Documentation

Cash Coffman

February 29, 2024

## 1 Introduction

In this event we need to make the robot go through a unknown maze. We will need to calculate the distance of each stretch of the maze so we can calculate how long the bot needs to stay in a state

## 2 The Speed Formula and Calculating State time

The formula for calculating speed is given by:

$$speed = \frac{distance}{time} \quad (1)$$

- **speed** is how fast an object is moving.
- **distance** is how far the object has traveled.
- **time** is the amount of time it took for the object to travel the distance.

We can use this formula to derive the time the motors need to spin to go a set distance if we know the robots speed

$$1.0m/s = \frac{distance}{time} \quad (2)$$

From this we can conclude that:

$$time = \frac{speed}{distance} \quad (3)$$

### 3 Function Documentation for Robot Movement Commands

#### Backward Function

This function is responsible for moving the robot in the backward direction.

```
void backward(int speed, unsigned long duration)
```

**Parameters:**

- **speed** - The speed at which the motors should run (0-255).
- **duration** - The time in milliseconds the robot should move backward.

**Operation:** Sets the motors to rotate in the direction that moves the robot backward, then stops the movement after the specified duration.

#### Left Function

This function turns the robot to the left.

```
void left(int speed, unsigned long duration)
```

**Parameters:**

- **speed** - The speed at which the motors should run (0-255).
- **duration** - The time in milliseconds the robot should turn left.

**Operation:** Adjusts the motor speeds to turn the robot left, then stops the movement after the specified duration.

#### Right Function

This function turns the robot to the right.

```
void right(int speed, unsigned long duration)
```

**Parameters:**

- **speed** - The speed at which the motors should run (0-255).
- **duration** - The time in milliseconds the robot should turn right.

**Operation:** Adjusts the motor speeds to turn the robot right, then stops the movement after the specified duration.

## Stop Function

This function halts any movement by stopping both motors.

```
void stop()
```

**Operation:** Sets all motor control pins to LOW, effectively stopping the robot and ceasing any further movement. It also sets the speed to 0 by writing a 0 PWM signal to the enable pins for both motors.

## Forward Function

This function propels the robot forward.

```
void forward(int speed, unsigned long duration)
```

### Parameters:

- **speed** - The speed at which the motors should run (0-255).
- **duration** - The time in milliseconds the robot should move forward.

**Operation:** Sets the motors to rotate in a manner that moves the robot forward, applying the specified speed. The robot continues moving forward for the duration specified before automatically calling the ‘stop’ function to halt movement.

## Setup Function

This function initializes the robot’s motors and serial communication.

```
void setup()
```

**Operation:** Sets up the motor control pins as outputs and starts serial communication. This is where the robot’s initial configuration is defined, and it runs once at the start of the program.

**Note:** Each of these functions plays a critical role in the operational capabilities of the robot, allowing for precise control over its movements. By manipulating the parameters of these functions, the robot can be programmed to navigate through the maze efficiently.

## 4 Conclusion

So given the robot travels at SPEED; we can conclude that in order to calculate the time for each state, we will need to find the distance of each stretch of the maze and plug it into the formula.