

## Programming Challenge

# Wait States

### Investigation Summary:

#### Part 1

Download and run the sample code below on your robot. Observe how far the robot travels. Then modify the amount of time that the robot travels by changing the value of line 5 from 500 to 1000. Download and run the program and observe the distance the robot travels. Repeat these steps for values of 2500 and 3250 as well. In each case, the variable that you are manipulating is the amount of time that the robot “waits” until it does something else.

### Sample Code

```

1 task main()
2 {
3     motor[leftMotor] = 127;
4     motor[rightMotor] = 127;
5     wait1Msec(500);
6 }
7

```

#### Part 2

Measure the average distance that the robot travels in 500 milliseconds, using three attempts:

Attempt 1:

Attempt 2:

Attempt 3:

Average Distance:

Use the Average distance you found to predict how far the robot will travel for 1000, 2500, and 3250 milliseconds. Fill in the chart with your predictions.

### Prediction Chart

Wait state in milliseconds	Wait state in seconds	Predicted distance traveled
500	1/2	
1000		
2500		
3250		

## Programming Challenge

### Wait States

Now find the average distance for the 1000, 2500, and 3250 trials. In each case, run the program three times.

Measure the average distance that the robot travels in 1000 milliseconds:

Attempt 1:

Attempt 2:

Attempt 3:

Average Distance:

Measure the average distance that the robot travels in 2500 milliseconds:

Attempt 1:

Attempt 2:

Attempt 3:

Average Distance:

Measure the average distance that the robot travels in 3250 milliseconds:

Attempt 1:

Attempt 2:

Attempt 3:

Average Distance:

Fill in the chart with your data and calculate the percentage of error with the provided calculation.

Wait state in milliseconds	Predicted distance traveled	Actual distance traveled (average)	% Error
500			
1000			
2500			
3250			

$$\% \text{ Error} = \frac{\text{predicted distance} - \text{actual}}{\text{predicted}} \times 100\%$$

## Programming Challenge

### Wait States

What is the correlation between distance and time, if any?

Which millisecond value gave you the largest percentage of error? The smallest? Speculate why in both cases.

What (if any) external factors that seem to influence the results?

What could you do to reduce error in the system?