CSCI 3302 Spring 2017 Robotic : Lab2.1
ChenHao, Cheng

Inspect the line following code provided by ArcBotics. What happens during the delay(100) statement and where would you want to introduce your odometry code?

While executing the delay(100) statement, it would stop 0.1 second and keep running code. We want to introduce the odometry code before the delay statement. The reason is that the Sparki start running before any other code of calculating the z angle.

Think about what would happen if calculating a position update would take longer than 100ms. Think about ways to measure the time the execution of a command takes and making sure every loop takes exactly 100ms. Hint: check out the millis library for Arduino.

If there's an error for the time calculation, then the data of calculating Sparki speed would incorrect. In the mean time, the Sparki position would more incorrect when we run the program. Using the millis() function, we can get the elapsed time at the beginning and the end to see the entire program.

Use the millis library to measure how long it takes the robot to move 30cm and calculate its speed from there. This will allow you to calculate its speed in m/s without actually measuring the wheel diameter.

```
firstTime = millis()
sparki.moveForward(30) // 0.3 meters
secondTime = millis()
sparki.println(secondTime - firstTime) = 10.77
30cm / 10.77s = 2.78 cm/s

2.78 cm/s = 0.0278 in meter/second
```

This method allows me not need to measure the wheel dismeter. I use millis to get two different times and subtract each other and then convert the proper unit.

Display the robots pose on its display. (Make sure this operation does not destroy your timing!). What do you expect the display to show when the robot arrives at the start line at the second (the third, the fourth) time? What actually happens?

When the Sparki arrives at the start line at the first(and second, third, fourth..) time. We would expect the display is 0 because the inner map is 360 degree. But in practice, we got 1% error every time when the Sparki arrives the start line.

Use the robots sensors to identify the start line. How could you exploit this information to reduce your error?

We could exploit the information and reduce my error by resetting the coordinates of the origin.