Project Rover

Project Demonstration : During Tutorials April 7th and April 9th 2015
Written Test: During Lecture on April 7th 2015
Report Due Date: 11pm April 10th 2015

Introduction

In this challenge, your robot will start in the middle of an unknown cell location on a grid as shown in the figure below. The main objective is to first calculate the current position of the robot then find the lost treasure (a block) on the grid and transport it back to the starting cell.

Project Guidelines

1. The robot will be placed randomly in a cell. It has to calculate its starting position and orientation within the arena.
2. After the robot has finished step 1, a signal should be given to the instructor/TA to place the treasure block in the arena. The signal can be a tone or a text displayed on the computer screen.
3. The treasure will be placed randomly on the grid, position of the treasure should be autonomously found by the robot.
4. Your robot has to track the lines of the grid using the light sensor and search for the treasure block using the ultrasonic proximity sensor. Once achieved, a sound or text signal has to be indicated.
5. After the treasure is located, the robot has to move it back to the starting cell predefined at the start of the competition. Once achieved, a sound or text signal has to be indicated.
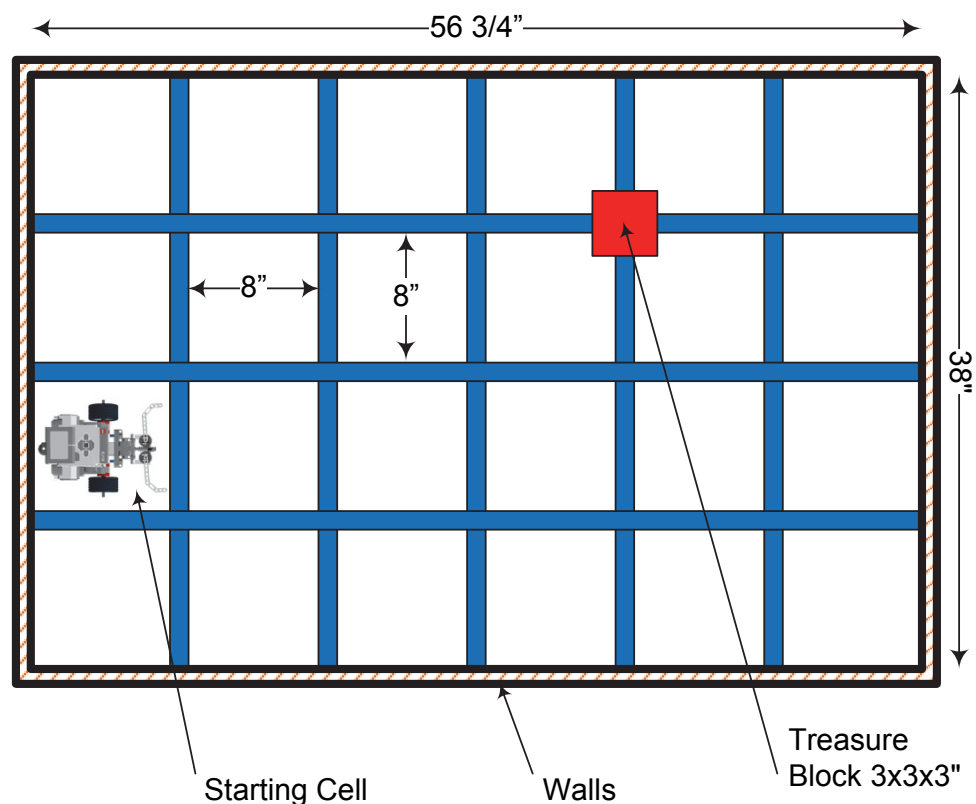

Figure 1

Project Constraints

Mechanical Constraints

➢ Robot should start in a center of a cell as shown in figure 1
➢ Robot should completely fit inside a cell without any parts overlapping or in contact with grid lines, a footprint of 8"x8" would fit but might cause problems when turning.
➢ Robot should only use the color sensor to navigate the maze while tracking grid lines
➢ Robot can only use the ultrasound and/or the touch sensor to find the treasure block
➢ Treasure block will be placed at an intersection of grid lines
➢ Robot can either hold, carry, or guide the block back to the center of the starting cell

Software Constraints

➢ Fundamental motion functions separate from the main loop (ex: MoveFwd1Cell, RotateCW90, RotateCCW90, etc.). These functions should use motor encoder values to control motion (i.e. timers are not allowed).
➢ Robot's orientation has to be calculated and monitored using a single integer "int RobotCompass" such that North=0, East=1, South=2 and West=3. Enumerators are strongly encouraged in this case.
➢ Robot's original position has to be calculated then monitored while robot is navigating the grid.
➢ Robot's position and orientation has to be saved in a two dimensional array of structures as:
➢ Each structure will hold
    a. A string holding the robot's name
    b. The robot's current orientation
    c. The robot's current position/intersection
    d. Other parameters if needed
➢ The size of the two dimensional array should be 3 rows and 5 columns referring to the number of intersections in the maze. When a robot changes its position on the grid, the data structure should be updated.
➢ The following elapsed time intervals should be displayed on the PC screen
    a. Elapsed time for robot to calculate its starting position and orientation
    b. Elapsed time for robot to find the treasure block and reach it
    c. Elapsed time for robot to move the treasure block back to the starting position