IEEE Ottawa Section

IEEE Ottawa Robotics Competition
Compétition de robotique d'Ottawa d'IEEE

King of the Hill Challenge

Last Revised: 20 March 2016

Table of Contents

ONLY TEAMS WITH EV3 KITS CAN PARTICIPATE IN THIS CHALLENGE

Disclaimer

It is your responsibility to read and understand this document on a regular basis because we may update it from time to time.

King of the Hill Challenge

The objective of this 1 vs. 1 robot challenge is for your robot to be the first robot to reach and climb the hill represented by a bridge at the end of the arena. Before reaching the hill, your robot will have to navigate through a maze with various obstacles. If at the end of the time limit no robot has reached the hill, the robot closest to the hill will be declared the winner. Since obstacles will change in position and in number, robot design and programs that exhibit general maze-solving abilities are encouraged.

Challenge Rules

1. At the start of the challenge, all teams will gather at the contest area and no further adjustments to the robot will be allowed.

2. Each team will be assigned either the right or the left side of the arena and all robots must start on their respective sides in a designated START area.

3. The front wheels of the robots must start behind the green START line in the designated START area.

4. All robots must have a 3-second delay at the beginning of their programs.

5. For each run, robots will have a time limit of 1 minute to complete the maze and reach the hill. After the start of the run, there must be no human interference.

6. Any type of physical contact by the robot is allowed in order to claim the hill, except for striking movements, which will result in disqualification.

7. The hill is captured at the end of the run when a robot reaches the top of the bridge.

8. Your robot must be controlled by an EV3 brick for this challenge and the gyro sensor is permitted to be used in this challenge. Motors, sensors, and other non-electrical parts continue to be permitted from any Lego Mindstorms kit.

9. Your robots must be smaller than 8 by 10 inches.

Judging and Scoring

1. The maze with all the obstacles will be modified on and during the day of the competition.

2. All teams will gather at the competition area and remain there for the remainder of the round.

3. Judges will time and score each run.

4. Teams will receive the following points for a run:
    a. First robot to claim the hill will be awarded **4 points**.
    b. If no robot has claimed the hill within the time limit, the robot closest to the hill will be awarded **2 points**.
    c. If a robot does not have a 3 second delay, **1 point will be deducted**.

5. The first half of the competition will be a round-robin style tournament. Each match will consist of three runs.

6. Teams will be ranked based on their round robin wins. Team will then compete in elimination playoffs.

7. Decisions of the judges are final.

King of the Hill Diagram

The dimensions of the entire arena are 4 by 6.5 feet, each of the maze walls is approximately 10 inches long and has a minimum height of 4¾ inches.

Your robots be smaller than 8 by 10 inches.

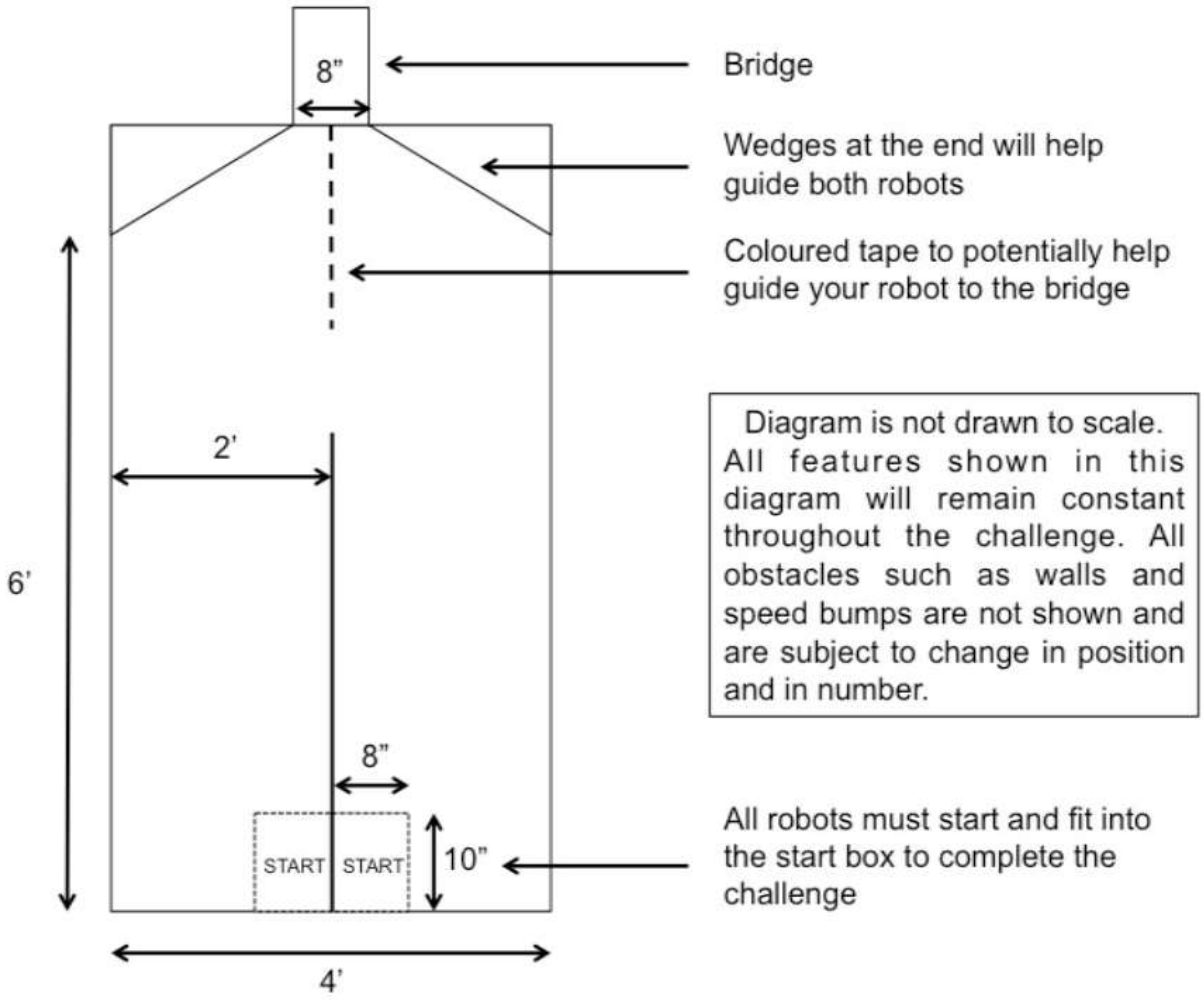
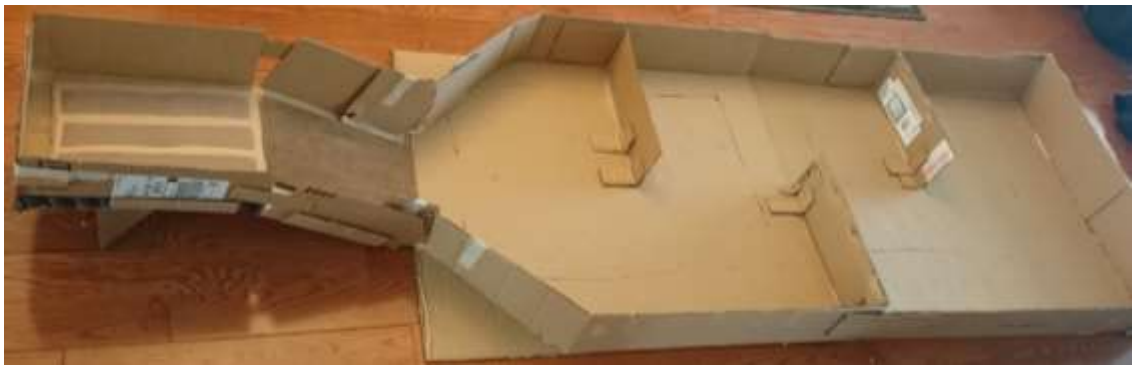


Figure 1: King of the Hill Arena Dimensions.

Also, we would like to thank Mr. Alan Stewart for providing materials in order to make the arena for this challenge.

Sample Homemade KOTH Arena

Some teams got creative last year in trying to create their own King of the Hill arena and used inexpensive cardboard.



Suggestions & Hints

1. There are wedges at the end of the arena guiding both robots to the hill. You may want to give special considerations to the design of your front bumper. Some ideas are:
    a. Shape of the front bumper (i.e. triangular, semicircular)
    b. "Cage" surrounding the robot

2. Think of ways to try "wall finding". King of the Hill is set up as a maze with obstacles which may be modified during the day of the competition, however the outer walls will always stay constant.

3. Watch YouTube videos on Roombas.

4. Think of how your robot is built. Does it use its mass effectively? Where are all of the sensors placed? Does your program use all of the sensors on it?