

**FOLLOWING ‘18**

**FOLLOWING’18**

!!!Are you all ready to squeeze your brains to develop the most fascinating machine which is going to rule the era!!!

Then get ready to dirty your hands in building a robot for following’18 - the annual flagship event of RMI. Like every year this year also we have come up with a new exciting problem statement and attractive prizes. Learn a lot of new stuff and enhance your skills by grabbing this wonderful opportunity.

So what are you still waiting for??

Get going ahead and check out the problem statement !!!

Happy robotics!!!

**PROBLEM STATEMENT**

Okabe Rintarou, the mad scientist, has a time machine. He eventually finds out that his childhood friend Mayuri, dies as a consequence of his time travel. He travels to different timelines to save her. But every timeline has a more disastrous consequence than the previous. So he decides to undo all his time travels and return to the initial timeline

**ARENA LAYOUT:**

**5**

**4**

**3**

**2**

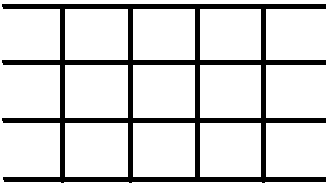
**1**

**T3**

**T4**

**T1**

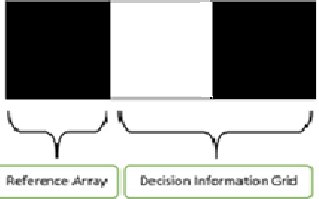
**T2**

****

There are 4 parallel lines, with equal perpendicular transversals, each represent a distinct timeline. The robot will start at any one of the lines. The goal will be to make a robot which travels to all the distinct timelines and travels it for one stretch(each number represents a stretch) and returns to the timeline which it started at for the last stretch.

Bonus:

The **1X3 matrix** should be read and the ending timeline should be calculated accordingly

**Sample matrix**



**Timeline 2 (T-2)**

**Timeline 3 (T-3)**

|  |  |  |
| --- | --- | --- |
| **Matrix colour 1** | **Matrix colour 2** | **Decision** |
| White | White | Timeline - 1(T-1) |
| White | Black | Timeline - 2(T-2) |
| Black | White | Timeline – 3(T-3) |
| Black | Black | Timeline - 4(T-4) |

The bot should finish in a timeline which would be read through a matrix before entering the lines. i.e. travel a stretch in all the lines and finish in the stretch corresponding to the matrix. Also, you should somehow “show” what your bot reads from the 3x3 matrix code. It can be shown/displayed in anyway, it’s left to your imagination!

The reference array will always be black.

**ARENA DESCRIPTION:**

1. Arena consists **of parallel black lines with black perpendicular transversals** on a white background**.**

Width of the black lines will be **3.00±0.2cm**

The spacing between between 2 parallel lines will be **70cm**

2. The arena layout shown here is just a sample. There may be any number of timelines and columns.

3. The code matrix would be a **2 cm X 6 cm** grid which comprises of **three** smaller squares of dimensions **2 cm x 2 cm**

4. The bot should not traverse more than one stretch in a timeline except for the starting timeline (or) the timeline given by the matrix.

5. The bot should choose the proper perpendicular transversal

(considering rule 3) and should finally reach the starting timeline or the timeline given by the matrix (for bonus points).

**RULES:**

1. A team can have a maximum of **three** members.

2. Maximum bot size must be **50cmx50cmx50cm**.

3. Each team can have a maximum of **two final runs** (best of two) and one trial run (not considered for evaluation).

4. The code is allowed to be changed only after the first trial. Any changes in the code made in between any of the two final runs will deem the team invalid of the 50 points offered by “1”(refer to points awarded section)

5. No two points on the bot can have a potential difference greater than **12V**.

6. The bot must be powered only by **on board batteries**.

7. Only ‘Embedded-C’ coding is allowed. Choice of microcontroller is left to the participant.

8. Maximum time for solving the problem and reaching the final timeline (either starting or the one given by matrix) is **10 minutes**.

10. The starting procedure of the bot should be simple and should not involve giving the bot any manual force or impulse in any direction.

11. Participants are allowed to adjust sensors (Gain, Position etc.), change speed settings and make repairs after the first run.

12. The decision of the judges is final.

**POINTS AWARDED:**

1. **+50 points** for traversing the grid and finding out the number of timelines as well as the starting timeline (i.e without hardcoding the number of timelines and starting timeline at the beginning).
2. **+20 points** for traversing a single stretch and moving to the next timeline via the perpendicular transversal.
3. **+20 points** for reaching the initial timeline.
4. **+20 Points** for traversing only one stretch in each timeline except the starting one (or) the one given by matrix
5. **+10 points** for reading and displaying the matrix
6. **+30 points** for reaching the timeline given by matrix
7. **Top 2 fastest** teams will gain **+20 points**.
8. **Top 2 teams** with compact/best design will get **+20 points**.

1. Usage of **self-fabricated boards/sensors** (should be working) will add **30 points** to your score. Partial points also will be awarded based on the complexity of the board being designed. The magnitude of the points will be decided by the organisers after scrutiny.

Try to get your combination right to gain maximum points.

**NOTE:**

1. The matrix and the arena given is only a sample. The real matrix may have any of the 4 sets of colours(WW/WB/BW/BB)

W- white B- black

1. Code must be explained by all members of the team separately after completion of the course.

\*Note:

Further changes will be intimidated and it is the responsibility of the participant to stay updated with the problem statement

**CONTACT:**