PUZZLE SOLVER ROBOT

1. Introduction

Problems, which exist all around us, are in fact the very fuel for our intellect. As long as problems exist, there will be engineers to solve them. However, solving these problems is not always an easy job and many a times, we need to devote a significant amount of time formulating the best practical solution to these problems.

Engineering, in a way, can be described as the meticulous art of solving problems in our daily life. As the popular saying goes, "Necessity is the mother of all invention", however, Engineering is the tool, which when wielded, facilitates that invention.

e-Yantra Robotics Competition Plus 2015 introduces the "Puzzle Solver Robot" theme as a means to test the algorithm development skills of the participants. Such skills are mandatory to solve problems in a number of industrial as well as commercial applications like Warehouse Management, Air and Road Traffic Control, Shipyard Cargo Management, Automated Home Delivery, etc.

In this theme, teams have to solve a mathematical puzzle involving a variety of topics like Digital Image Processing, Motion Planning and Simple Arithmetic. The arena for this theme consists of two divisions – The first is a grid with each **Cell** in the grid containing a digit from 0-9; the second is a grid with a maximum of four numbers in the range 0-20 placed arbitrarily in any **Cell** of the grid. The robot has to choose the numbers in D1 such that they add up to each number in D2.

The challenge is to complete this task in the shortest time possible. The robot that performs the task best in accordance with the rules set for this task will be declared the **WINNER** of the competition.

2. Theme description

1. Prior to the start of theme execution, an input image of the arena is given to the team. An example is shown in Figure 1 that is used in this rulebook to explain the Theme. The arena represents a simplified abstraction of a puzzle with the following:

   - There are two divisions: Division 1 (**D1**), and Division 2 (**D2**).
   - D1 is a grid having 12 **Cells** and D2 is a grid having 24 **Cells** as shown in Figure 1.
   - Each **Cell** in D1 contains a one digit number (i.e. number from 0 to 9) while any one or two digit numbers only upto 20 (i.e. numbers from 0 to 20) are present in **some** of the **Cells** in D2.

2. The team analyzes the given image on computer by using image processing which generates the information regarding the **numbers and their positions in D1 and D2** and then communicates this information to the robot.

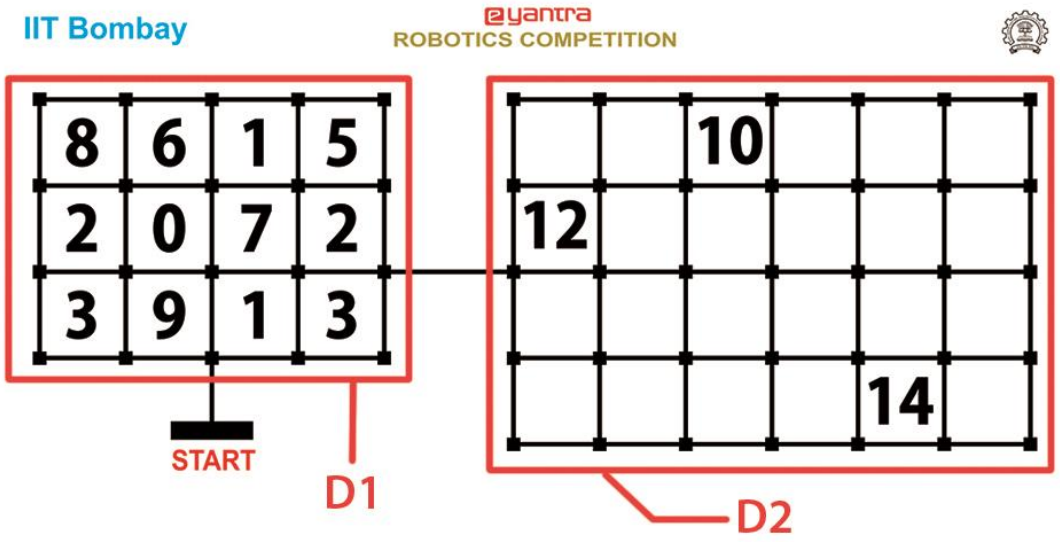

Figure 1: Input Image of Arena

3. The goal of this Theme is to choose numbers in D1 such that they add up to each number in D2.
4. The robot starts from START position and does the following:
   i. Traverses the grid in D1 and picks up the number.
   Pick up is considered when the number is displayed on GLCD (Graphics Liquid Crystal Display). While picking, the robot should be on the cell boundary of the corresponding number. The same is explained in detail in Section 6: Theme rules.
   ii. Traverses the grid in D2 and deposits the number.
   Deposition is considered when the message "Deposited" is displayed on GLCD. While depositing also the robot should be on the cell boundary of the number on which the picked number is deposited.
   iii. Numbers deposited on a number in D2 must add up to that number.

iv. The robot buzzes the buzzer for 1000 ms indicating completion of the solution for that number in D2.
v. Steps i – iv are repeated till all the numbers in D2 are solved.

5. **FINISH** line is **NOT** marked on the arena; the robot stops when it solves the puzzle and sounds a continuous buzzer as an indication to show that it has finished the task.

Note: You have already completed "Image Processing" and "Algorithm Development" in Stage 1. You may integrate these to complete the Theme.

3. Arena

1. The arena represents a simplified abstraction of a puzzle. It is divided into two divisions D1 and D2 with a black line connecting them. D1 and D2 consist of grids having nodes at the intersections. D1 consists of 12 Cells, whereas D2 consists of 24 Cells.
2. Each Cell in D1 contains a number while numbers are present in some of the Cells in D2. All the numbers which are used in arena are given in form of patches.
3. Arena design is shown in Figure 2. It is divided into two sections,
   - Arena section (top) and
   - Patch section (bottom).
4. A pdf file containing the arena design is given to the teams. Each team prints the arena design on flex sheet according to the directions given along with the file. **Note: You must print the entire arena design having both the Arena and Patch sections.**
5. Teams are not authorized to make any changes in the arena design. Any team making unauthorized modifications will be disqualified from the competition.

Details of Arena design (Refer to Figure 2):

- Dimension of arena is 243.84cm x 152.78cm.
- The arena consists of grids in D1 and D2, made of black lines of thickness 1 cm. Square nodes of dimension 3cm x 3cm are provided at the intersection of two lines.
- The dimension of each Cell in D1 and D2 is 30cm x 30cm.

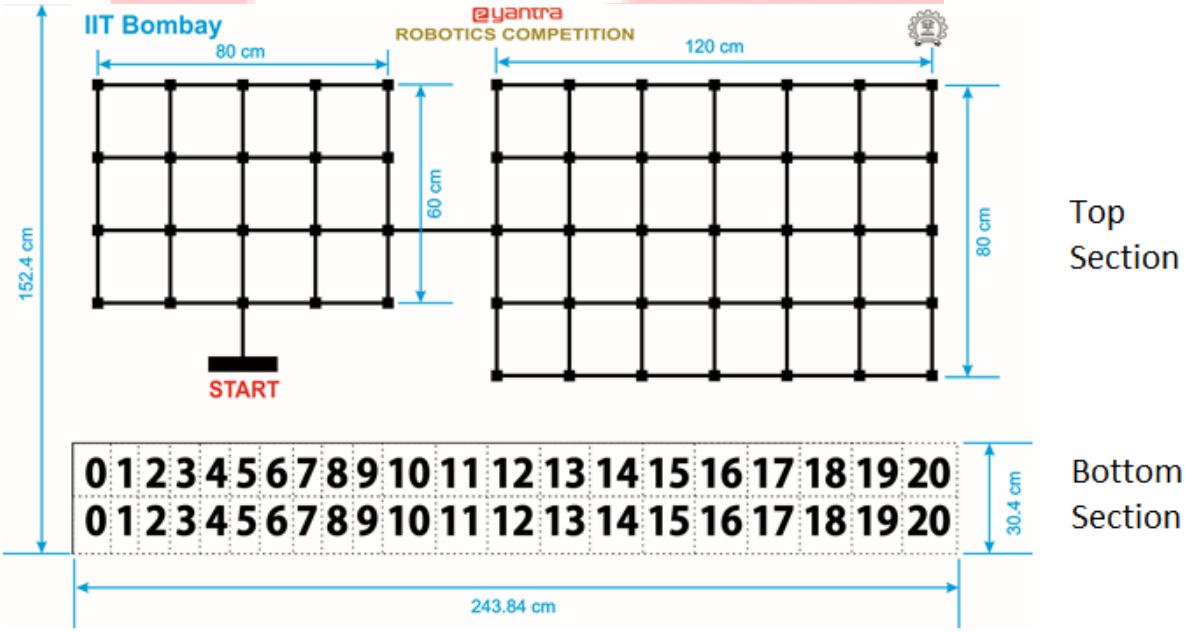

Figure 2: Arena Design; Top – Arena section, Bottom – Patch section

3.1 Preparing and Placing the Patches:

- Once the arena design is printed, the team must cut along the indicated line to separate the Arena section and the Patch section.
- The dimension of Patch section is 30.5cm x 243.84cm with the numbers printed on it as shown in Figure 3.
- Cut along the dotted lines to get 42 individual numbered patches.

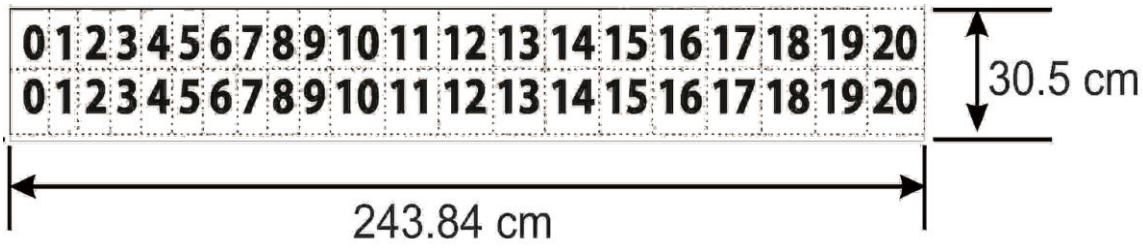

Figure 3: Patch Section

- Paste the number patches on to the arena as per the **input image** using **transparent cello tape only**.
- Figure 4 shows pasting the number patch using cello tape.

Figure 4: Pasting Number Patch Using Cello tape

- For our example, after pasting all the patches in the appropriate locations, the arena looks like Figure 5.

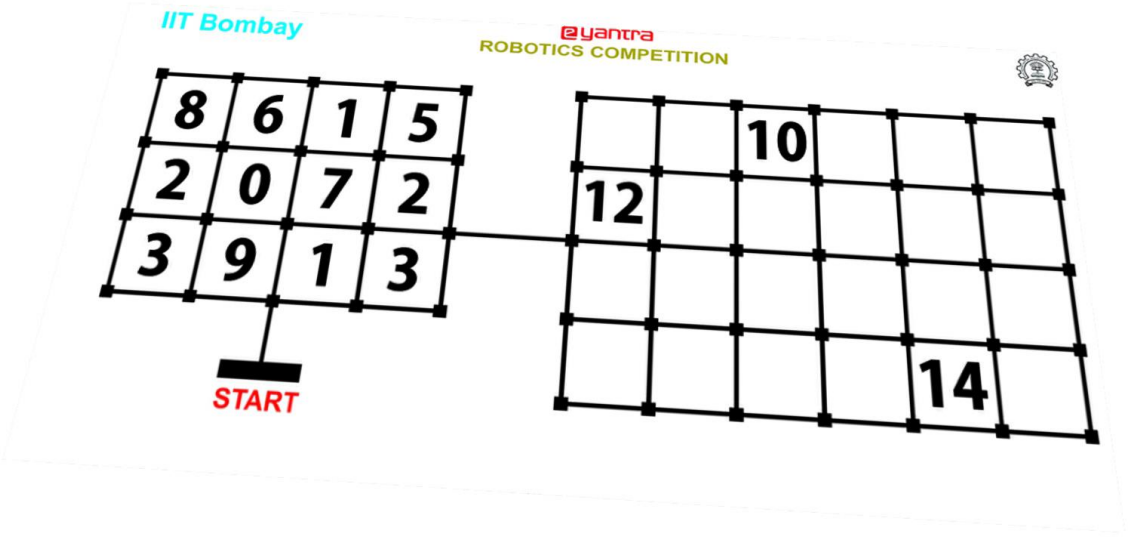

Figure 5: Example - Placement of Patches Using Input Image

Note: You must use removable tape of good quality because during the course of the competition, you will constantly be removing the numbered patches and pasting them again at different locations on the arena. Bad quality tape will damage the arena.

Now, we are ready with the arena. Please maintain the arena in a good condition. If the arena is found damaged or in a condition not good enough to properly evaluate the team, e-Yantra has the right to disqualify the team. **The final decision is at the discretion of the reviewer.**

Note: The arena shown in Figure 5 is specific to the example considered. During the competition the numbers and their positions will be different (given as an input image) and hence the placement of patches will vary accordingly.

WARNING: Please be careful while handling the flex sheet – avoid folding it like a bed-sheet since the resultant folds will cause problems while the robot moves. One way of "flattening" flex if it has been compromised is to hang it for a few hours in the sun -- it tends to straighten out. Never attempt ironing it or applying heat of any kind -- it may be a fire hazard. Best is to store the sheet rolled up.

4. Hardware Specifications:

4.1 Use of Firebird V:

- All participating teams must use **only** the Firebird V robot sent to them in the kit. **Only one** robot given in the kit is allowed per team.
- Team shall not dismantle the robot
- The robot should be **completely autonomous**. The team is not allowed to use any wireless remote or any other communication protocol or devices such as a camera while the robot is performing the task.

4.2 Use of additional components not provided in the kit:

- No other microcontroller-based board shall be attached to the Firebird-V robot.
- Teams may connect external actuators along with their driver circuits to the Firebird V robot only on the condition that the actuators must be controlled through the Firebird V robot.
- The team is not allowed to use any other sensors apart from those provided in the kit.
- The teams must use a laptop/computer capable of running OpenCV and Python.

4.3 Power Supply:

- The robot can be charged through battery or auxiliary power supply. These are shipped with the robot.
- The team cannot use any other power source for powering the robot.
- The team can use auxiliary power during practice but the final demonstration should only be made using only the battery powered robot.

5. Software Specifications:

- e-Yantra has provided all teams with ATMEL STUDIO 6, a free software for programming AVR microcontroller. Participating teams are free to use any other open source Integrated Development Environment (IDE) for programming AVR microcontroller.
- The teams must use OpenCV and Python to write their code.
- Use of any non-open source libraries is not allowed and will result in disqualification.
- As per e-Yantra policy, all your code and documents are open-source and maybe published on the e-Yantra website.

6. Theme Rules

- The maximum time allotted to complete the task is 10 minutes. A maximum of **two runs** will be given to a team (the better score from the two runs will be considered as the team's score). A maximum of **two repositions** (explained below) will be allowed in each run.
- The team should switch **ON** the robot when told to do so by reviewer. This is the start of a **run**. The timer will start at the same time.
- Robot should be kept at the **START** line with the castor wheel of the robot positioned on the line.
- Once the robot is switched on, human intervention is NOT allowed.
    - The following are the steps of the task:
        i. The input image is given before the start of the run.
        ii. Robot is placed on the START position and turned on.
        iii. The team must use OpenCV and Python to detect the numbers and their positions on the input image and display the same on python IDLE console.
        iv. The teams must use the USB to serial cable to communicate the information from the computer to the robot.
        v. When the communication is completed, remove the USB to serial cable.
        vi. Press the **Boot key** to start traversal of the robot.
        vii. Note that the robot waits at the START position till the Boot Key is pressed.
        viii. The robot must traverse the grid in D1 and pick up a number and deposit it in D2. Details are explained below.
        ix. For each number in D2, the robot buzzes the buzzer for 1000 ms. when the numbers deposited adds to the number in D2.
        x. Repeat steps viii and ix for each number in D2. After completing all the numbers in D2, sound the continuous buzzer to indicate the end of the task.

- **Displaying detected numbers and their positions :**
    - As in Task 1 in Stage 1 of this competition, for D1, display the entire array of numbers in the cells starting from 0 to 11 and for D2, display only the Cell number and the number contained in that corresponding Cell.
    - For example, given the test image in Figure 1 as input, the output on the Python IDLE console will look like:

```
D1 = [8,6,1,5,2,0,7,2,3,9,1,3]
D2 = [ [2,10], [6,12], [22,14] ]
```

- **Picking up the number :**
    - Picking up a number is represented by displaying that number on the GLCD.
    - While picking, robot should be on the boundary of the cell. The position and direction of the robot can be on any of the four red dots given in Figure 6.

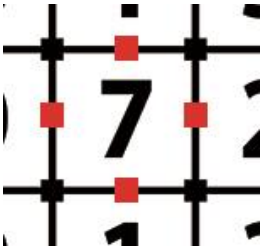

Figure 6: Position of the robot

- Two LEDs should be connected on either side (left and right) of the robot. To indicate the exact cell, robot must glow the LED on the corresponding side.

  Example:
  Suppose robot has to pick up the number 7.
  Robot traverses to the cell; if the position of the robot is as per Figure 7 then it should glow the right LED and display '7' on the GLCD.

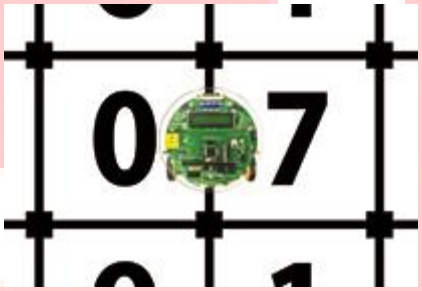

Figure 7: Example robot position

- **Depositing the number :**
  - Depositing a number is represented by turning off the LED at the appropriate Cell in D2 and displaying the message "Deposit" on the GLCD.
  - While depositing also the robot should be on the boundary of the cell.
    Note: When a LED is ON the robot should turn it off only when the number (on which the picked number is to be deposited) is on the same side as the lit up LED .
    Example: In continuation of the example above, suppose robot has to deposit the picked number 7 on number 10 in D2.
    Since right LED is ON, number 10 should be on Right Hand Side of the robot. This is illustrated in Figure 8.

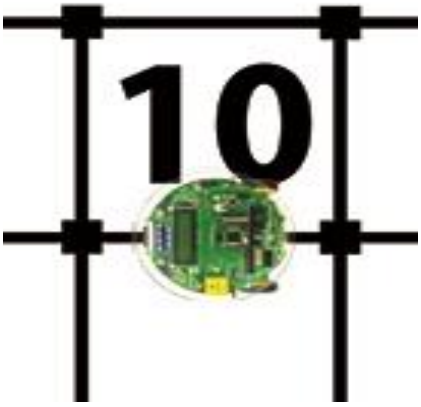

Figure 8: Valid Deposition Position

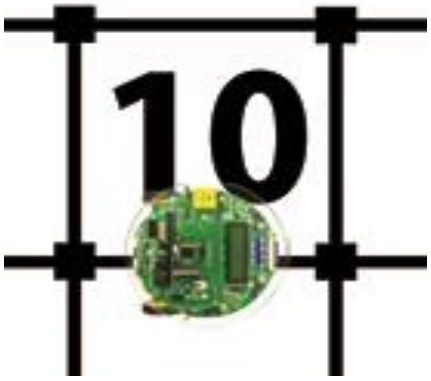

Figure 9: Invalid Deposition Position

- **Numbering in Divisions**:
  - In D1,
    - Numbers are present in **all** the 12 Cells.
    - Each cell may contain any one number from 0 to 9.
  - In D2,
    - Numbers are present in **some** of the 24 Cells.
    - A maximum of 4 Cells contain any one number from 0 to 20.

    Note: The numbers given in the input image are such that, at least one possible sum can be obtained for every number in D2 from the numbers given in D1.

- Buzzer sound for more than **5 seconds** will be considered as continuous buzzer.
- A run ends and the timer is stopped when:
  - The robot stops and sounds the continuous buzzer or
  - If the maximum time limit for completing the task is reached or
  - If the team needs repositioning but has used both repositioning options of that run.
- Second run will start once again whilst resetting the score, timer and arena. The score of both runs will be recorded and best of two runs will be considered as the team's score.
- Participants are not allowed to keep anything inside the arena other than the robot. The time measured by the reviewer will be final and will be used for scoring the teams.
- Time measured by any participant by any other means is not acceptable for scoring.
- Once the robot starts moving on the arena, participants are not allowed to touch the robot.
- The robot is not allowed to make any marks while traversing the arena. Any robot found damaging the arena will be immediately stopped; repositioning will be allowed as per the rules. **The final decision is at the discretion of the e-Yantra team.**

Repositioning of robot:

Suppose while traversing the arena robot strays off the black line (Refer to Figure 10), a member of e-Yantra team who will be monitoring the task will place the robot on the previous node (node already traversed by the robot) in such a way that both the wheels of robot are parallel to the node and castor wheel is on the black line (Refer to Figure 11). This is termed as a **Reposition**. Note that the timer used for measuring the task completion time in the competition will be continuously running during a Reposition and robot will not be switched off.

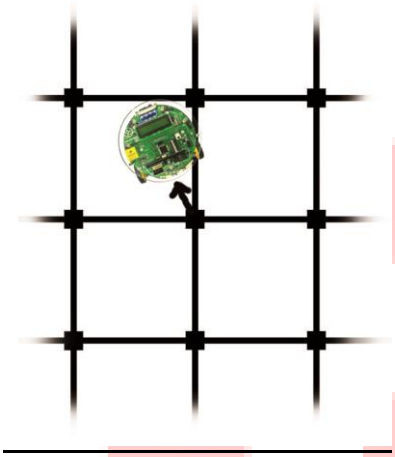

Figure 10: Robot strays off the black line

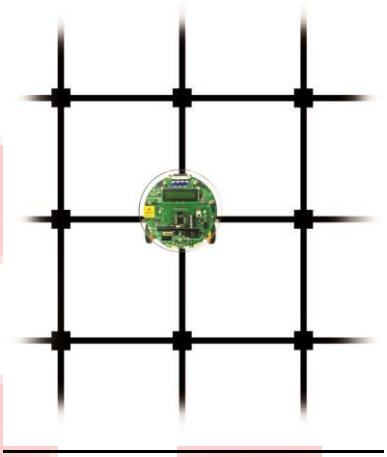

Figure 11: Placing of Robot after reposition

NOTE:

- **Example input images are given along with this rulebook on the portal for practice purpose.**
- **You will be given another input image having different numbers and positions just before the submission of Task 3: Video submission along with instructions to complete this task.**
- **After completion of all tasks, teams will be selected as finalists based on their cumulative scores across all the tasks. Complete rules and instructions for the finals at IIT Bombay will be sent to those teams that qualify for the finals.**
- **In case of any disputes/ discrepancies, e-Yantra's decision is final and binding. e-Yantra reserves the rights to change any or all of the above rules as we deem fit. Any change in rules will be highlighted on the website and notified to the participating teams.**

7. Judging and Scoring System:

- The competition time for a team starts from the moment the robot is switched ON. The timer will stop as soon as the robot finishes the task.
- Better score of the two runs for a team will be considered as the final score of the team.
- The team's total score is calculated by the following formula:

$$\textbf{Total Score} = (600 - T) + (CD \times 10) + \sum_{i=0}^{n} \left[100 - error_i \; X \; 10 \right] + B - P$$

Where:

$error_i = | Required\_no_i - Placed\_no_i |$

n can take value between 0 to 4, as there can be maximum of 4 numbers will be present in D2.

- **T** is the total time in seconds to complete the task.
- **CD** is the number of correctly detected and displayed numbers on python IDLE console.
- $Required\_no_i$ is the number at a cell in D2.
- $Placed\_no_i$ is the sum of the numbers deposited on the $Required\_no_i$.
- **P** is a penalty where 30 (thirty) points are deducted for each number which is displayed incorrectly on GLCD or the LED is not turned ON or OFF correctly
- **B** is a bonus of 100 points awarded,
  - Completes the task within 10 minutes,
  - Does not incur any penalty.
  - **error** is zero.

ALL THE BEST…!!!