COURIER SERVICE

1. Introduction

In our cities, we often see a few men rushing here and there in vehicles, wearing different colored clothes, and carrying huge backpacks. Yes, we are talking about couriers. A courier is like a genie: you make a wish, and there he appears with a parcel at your doorstep, fulfilling it. These men work day and night to make sure our parcels reach us on time and without any damage.

However, in many cases couriers tend to get lost finding their way through the city and often use unfamiliar roads to reach their destination. This can lead to delayed deliveries and unnecessary complications due to human error. Replacing the couriers with Autonomous Courier Robots will not only reduce the manual transportation tasks, but also increase the efficiency of delivering parcels on time along with reduced labor costs for the courier companies.

Keeping in mind the importance of this subject, “Courier Services” is one of the themes in the e-Yantra Robotics Competition Plus (eYRC+ 2015). In Stage 1, teams will learn the basic concepts in Image Processing and complete tasks to qualify for Stage 2. In Stage 2, selected teams will create a robot that goes around in an arena representing a city picking up parcels from various companies and delivering them to the respective customers. The team that performs the task best as per the rules set will be declared the **WINNER.**

2. Theme Description

The arena of this theme is a **6*6 Grid** made up of **36 cells**, representing the map of a city which is divided into two zones: **Warehouse Zone and City Zone** as shown in Figure 1.

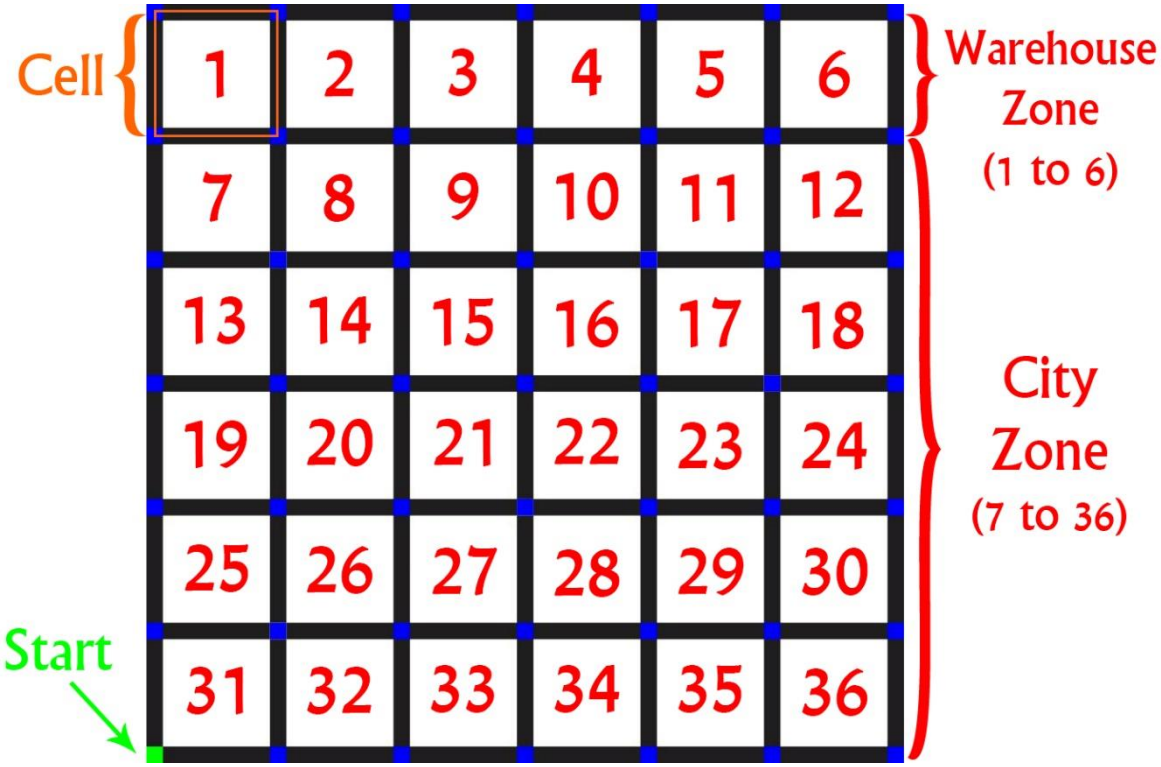

Figure 1: Zones and Cell Numbering

A. Warehouse Zone:

- Cell numbers **1 to 6** represent six Warehouses of different Companies **(Cell 1 = Company 1, Cell 2 = Company 2, … , Cell 6 = Company 6),** where the packages to be delivered are stored.

B. City Zone:

- Cell numbers **7 to 36** represent the **City** area, where the packages need to be delivered by the robot.

C. **Markers and Black Lines:**

- The **Green Marker** at the bottom-left corner represents the **Start.** This is where the robot should be placed at the beginning of the task.
- The **Blue Markers** present between two or more black lines represent **Junctions.**
- The black lines present between Junctions are referred to as **Links.**

Input Images:

There are two input images given to the teams 24 hours prior to the video submission (Task 5):

- **Test Image:** This Image is used to extract information like **Under Construction Road Sites, Functional Companies, Packages, Pick-Up Junctions, Delivery Junctions and Location of Traffic Signals.** All these terms are explained below. This image is given as an input image to the Raspberry Pi. Teams need to use image processing techniques learnt in Stage 1 of the competition to extract all the required information for completing the task (using OpenCV and Python). The extracted information will help them plan the traversal of the robot.
- **Arena Configuration Image:** This image is NOT used as an input to the Raspberry Pi. It will only be used by the teams to setup the arena before the run. It contains information like placement of Road Blocks and State of Traffic Signals (discussed in the Arena section).

The Test Image can be transferred to the Raspberry Pi using any USB Storage Device (Pen Drives and External Hard Disks being the most common options).

A sample Test Image is shown in Figure 2.

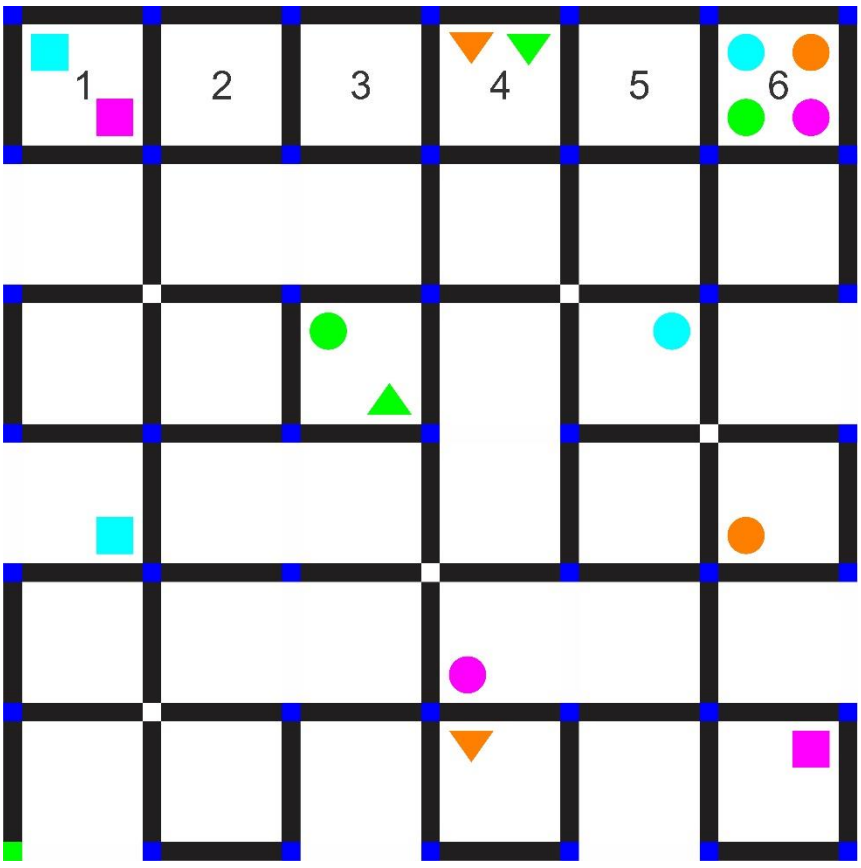

Figure 2: A Sample Test Image

A. Under Construction Road Sites:

- Under Construction Road Sites are the paths that are inaccessible to the robot and are represented by **Missing Links**. Links are said to be missing if there is no Link present between two Junctions. The robot has to avoid the Missing Links while planning its traversal from one point to another.

B. Functional Companies:

- For the purpose of the competition, we assume that in any Test Image only three companies out of six are functional, i.e., only three companies will have Packages that need to be delivered.
- For example, in Figure 2 Warehouses of only Company 1, Company 4 and Company 6 have Packages that need to be picked-up and delivered by the robot.

C. Packages:

- Packages of each Functional Company are differentiated by a different **shape** as shown in Figure 2:
  - **Square (S)**
  - **Triangular (T)**
  - **Circular (C)**
- For example, in Figure 2, Company 1 deals with **S**, Company 4 with **T** and Company 6 with **C.**
  Note: One Company deals with ONLY ONE shaped Package.
- In any given Test Image, each Functional Company can have a **minimum of 1 and maximum of 4 packages.**
- Packages can be of the following **colors** as shown in Figure 2:
  - **Sky Blue (SB)**
  - **Pink (P)**
  - **Orange (O)**
  - **Green (G)**
- These colors are used to make each Package unique, i.e., we should easily be able to differentiate one Package from another of the same Company.
  Note: A Company cannot have two Packages of the same color.

D. Pick-Up Junctions:

- Each Company Warehouse has a Pick-Up Junction (PUJ), which is located in the Warehouse Zone of the Arena, from where the Packages of that Company can be picked up as shown in Figure 3.

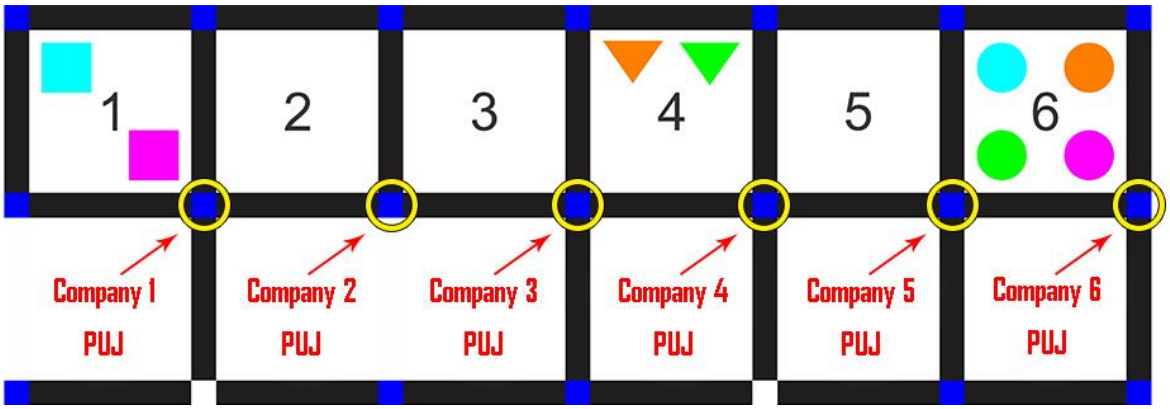

Figure 3: Company Pick-Up Junctions

- **Note: For this theme, teams do not have to create any mechanical structure for picking up Packages as the robot does not actually pickup any Package. It only needs to travel to the PUJ and indicate the pickup of a Package. The method used to indicate this pickup is discussed in the Theme Rules section below.**
- Pickup and Delivery can happen in any order -- the ONLY constraint being that at any given point of time, the robot can carry a maximum of 4 Packages.

  Note: Teams are free to design any algorithm that would provide the most efficient service.

E. Delivery Junctions:

- Images of Packages are present in the City Zone to indicate where the Packages have to be delivered. An example is illustrated in Figure 4.
- A Package should be delivered at the Junction that is nearest to the image of that Package in the City Zone referred to as a **Delivery Junction**.
- With reference to the example in Figure 2, Delivery Junctions for the packages in the City Zone are marked in Figure 4 using appropriate colored circles around the Delivery Junctions.

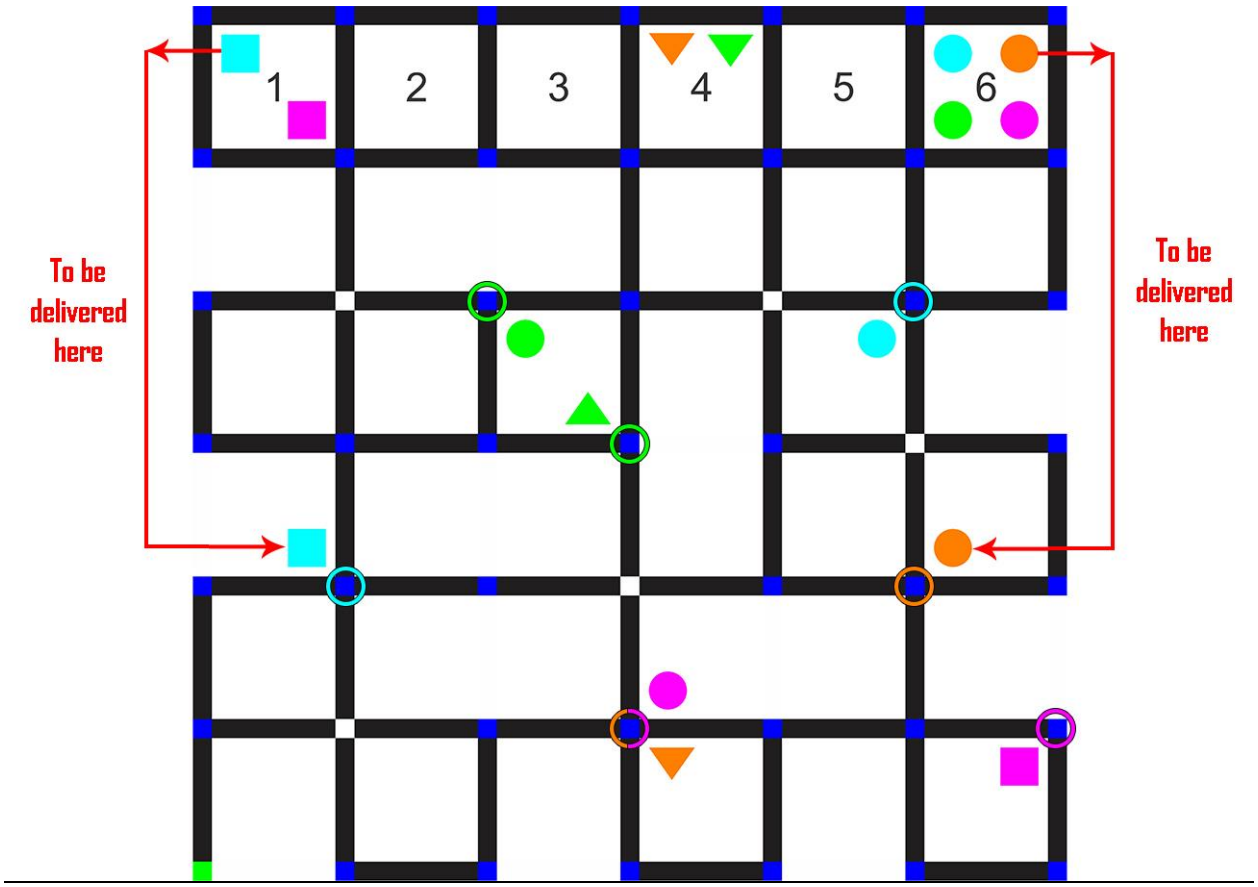

Figure 4: Delivery Junctions

- **Note: For this theme, teams do not have to create any mechanical structure for dropping the Packages as the robot does not actually drop any Package. It only needs to travel to the Delivery Junction and indicate the dropping of a Package. The method used to indicate this drop is discussed in the Theme Rules section below.**
- The robot can drop multiple Packages at a Delivery Junction. An example is illustrated in Figure 4 (Pink Circular Package and Orange Triangular Package).

F. Location of Traffic Signals:

- Some of the Junctions in the City Zone are **White** instead of Blue as indicated by Yellow Circles in Figure 5. These White Junctions indicate the **presence** of a **Traffic Signal.**

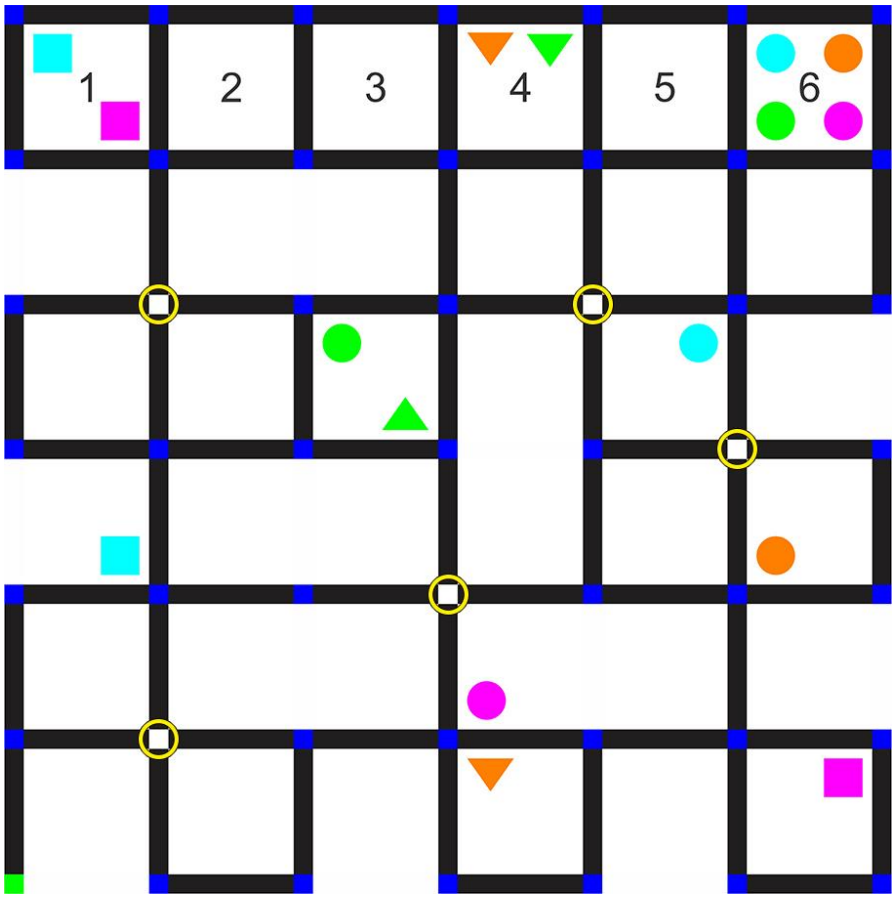

Figure 5: Location of Traffic Signals

- The robot can extract the location of a Traffic Signal from the **Test Image** using image processing.
- Each Traffic Signal has a State associated with it, indicated by RED or BLUE:
  - If the State is RED, the robot needs to halt over that Junction for a duration of 30 seconds (by calling the function " time.sleep (30)" and only then it can start traversing again.
  - If the State is BLUE, the robot can continue traversing without halting.
- **Note: On encountering a Red Traffic Signal, Teams are free to consider re-routing and choosing an alternate path if they feel they can save time using that alternate path.**
- The State of Traffic Signals is given in the Arena Configuration Image which is used to setup the arena (discussed in the Arena Section below).

- **Note**: We use both the images to configure the Traffic Signals:
  - **Test Image**: for finding the locations of Traffic Signals – for use with your code
  - **Arena Configuration Image**: to determine the State of the Traffic Signals and to set up the Arena used for traversal – for use by the robot to sense the State using the USB Camera at run time.
- **Teams hardcoding the State of the Traffic Signal will be disqualified at e-Yantra's discretion.**

Teams are free to mount the USB Camera and Raspberry Pi as per their design. The USB Camera is used for Line Following and extracting the States of the Traffic Signals.

After extracting all the required information from the Test Image:

- The robot plans its path and starts traversing the arena according to the algorithm used by the teams.
- It first goes to the Warehouse Zone to pick up the Packages.
- e-Yantra is providing a code snippet – **"Output_Format.py"** on the Resources Tab of the portal. **It contains the format in which the teams need to indicate the pickup and delivery of Packages. This code snippet needs to be integrated in your code for displaying the output on the screen of the laptop or computer used to remotely login into Raspberry Pi.** A Read_Me.pdf will be provided with this code that will explain how this code snippet needs to be used and integrated into your code.
- Through this code snippet we create an output window called **"Packages on the Robot"** and divide it into 4 sections. As mentioned earlier, the robot can pick up a maximum of 4 Packages at any given time. The 4 sections of the output window correspond to the 4 Packages that the robot can pick up.
- Initially, all 4 sections on the output window are **White** as shown in Figure 6**.** This indicates that the Robot is not carrying any Package.

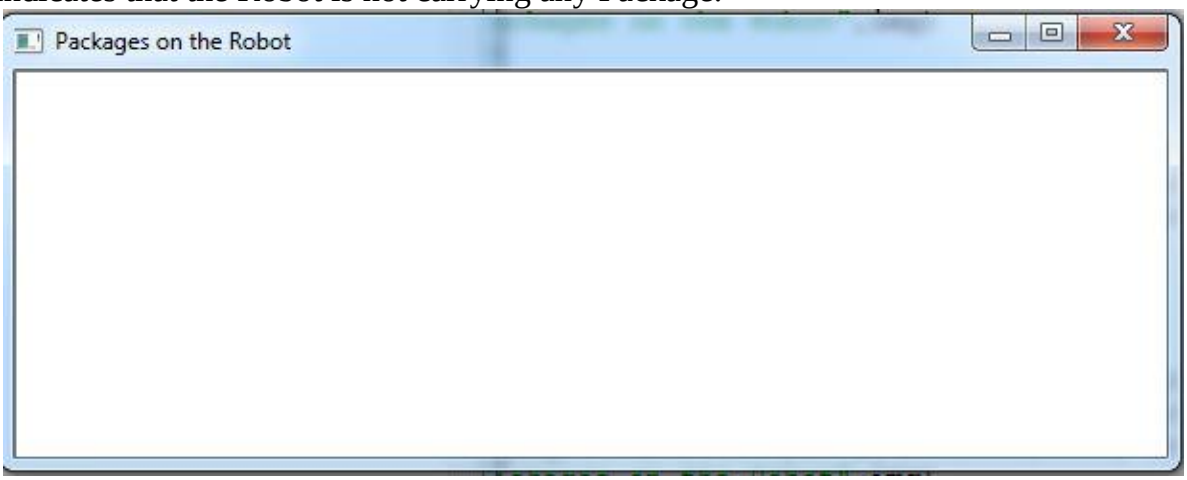

Figure 6: Output Window when the robot is not carrying any Package

- Once the robot reaches a Pick-Up Junction (PUJ), it should stop and call the appropriate pickup indication functions (mentioned in the Read_Me.pdf given along with Output_Format.py). The pickup is counted when the color of the Package appears and disappears thrice on one of the sections of the output window with an interval of 1 second between each appearance. After the third appearance the color of the Package freezes for that section of the output window. The robot can start moving again only after the color of the section corresponding to the picked up Package is frozen. For example, let's consider that 4 Packages are being picked up by the robot in the order: Sky Blue Circular, Green Triangular, Pink Square and Orange Circular. The indication on the output window with pickup of every Package is as shown in Figure 7.

Figure 7: Indication on screen when robot picks up Packages

- After reaching the corresponding Delivery Junction, the robot should stop and call the appropriate delivery indication functions (mentioned in the Read_Me.pdf given along with Output_Format.py). The delivery is counted when the color of the Package disappears and appears thrice with an interval of 1 second between each disappearance. After the third disappearance the color of that section becomes White again. The robot can start moving again only after the section corresponding to the delivered Package becomes White. For example, let's consider that 4 Packages are being delivered by the robot in the order: Green Triangular, Orange Circular, Pink Square and Sky Blue Circular. The indication on the output window with delivery of every Package is as shown in Figure 8.

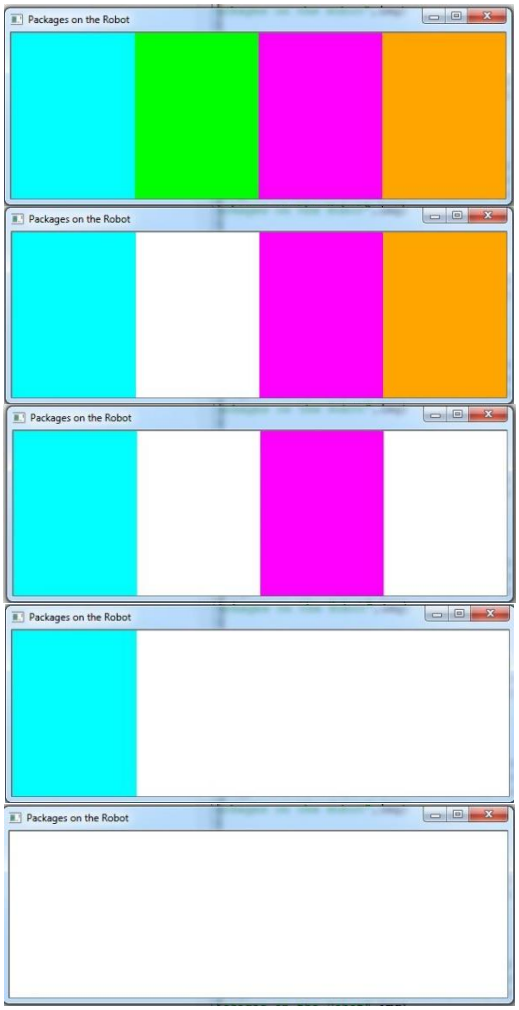

Figure 8: Indication on screen when robot delivers Packages

- The Task ends when the robot has picked up and delivered all the required Packages from all the Companies to their respective Delivery Junctions. To indicate the end of the Task, the robot should stop and call the appropriate task end indication functions (mentioned in the Read_Me.pdf given along with Output_Format.py). The task is

considered to be over when all 4 colors appear and disappear one after another on all 4 sections of the output window with an interval of 1 second between each color appearance.

- Click on this link to see a demo video indicating the pickup and delivery of 4 Packages, and end of the task: https://youtu.be/_APdCx_7h6g.
  Note: The text displayed in the video is only for explanation purpose. It will not actually be displayed on the output screen.

3. Arena

Preparing the arena:

Preparing the arena consists of two steps:

1. **Printing the arena design on flex sheet.**
2. **Setting up the Arena:**
    a. Placing **Road Blocks** on Under Construction Road Sites.
    b. Placing the Traffic Signals.

1. Printing the arena design on flex sheet:

- Arena design is shown in Figure 9.
- A .PDF file containing the arena design has been given to the teams. Each team prints the arena design on flex sheet according to the directions given along with the .PDF file.
- Teams are not allowed to make any changes to the arena design. Any team making any unauthorized modifications will be disqualified from the competition.

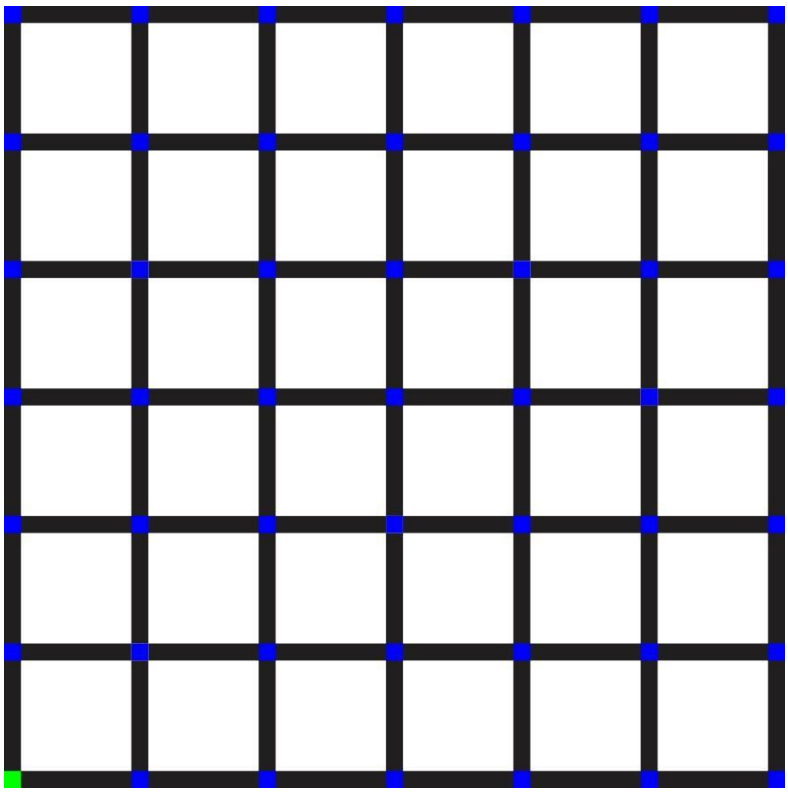

Figure 9: Arena Design

2. Setting up the Arena:

Teams will be given an Arena Configuration Image, containing information regarding placement of Road Blocks and State of Traffic Signals. In Figure 10 the Arena Configuration Image is given that corresponds to the Sample Test Image in Figure 2.

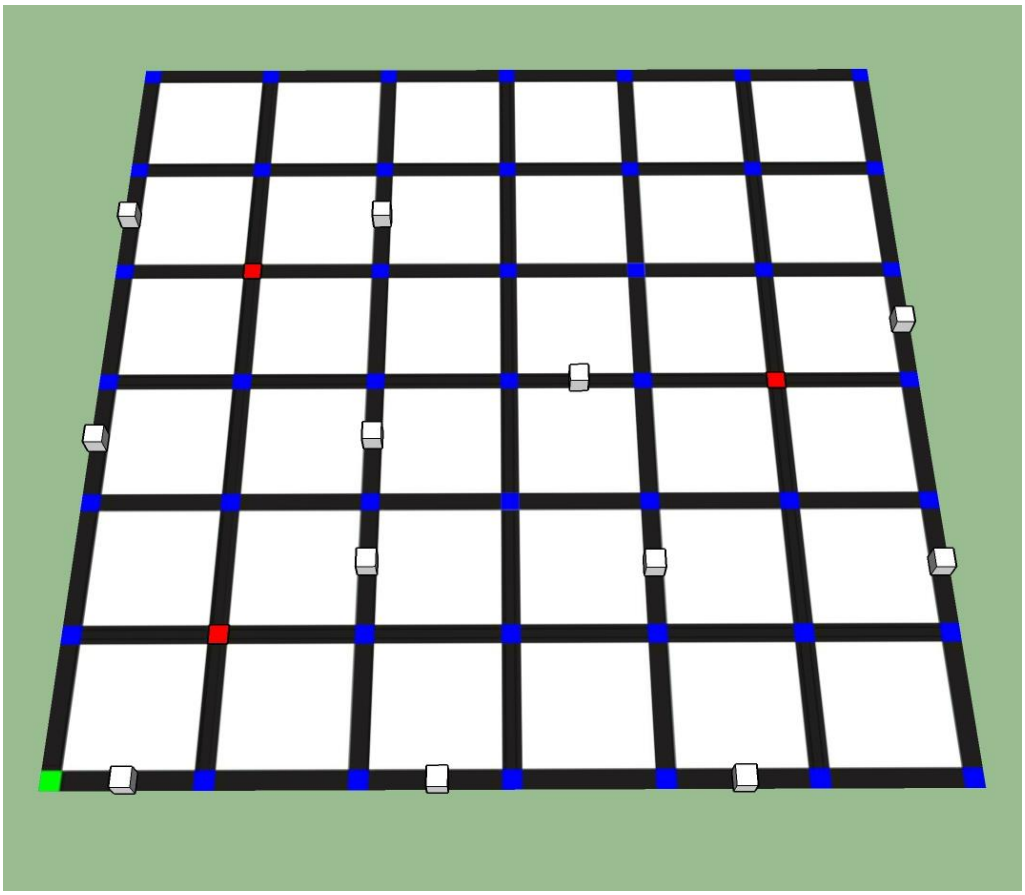

Figure 10: An Example Arena Configuration Image

a. Placing Road Blocks on Under Construction Road Sites:

- The 20 thermocol blocks prepared in Task 3 are to be used as **Road Blocks** to indicate Under Construction Road Sites.
- The placement of Road Blocks is done based on the Missing Links in the Test Image (An example is illustrated in Sample Test Image shown in Figure 2).
- Road Blocks are placed at the center of the Missing Links to visually indicate an Under Construction Road Site. Using Figure 2 when Road Blocks are placed, the Arena looks like Figure 10.

b. Placing the Signals:

- Red color paper has been provided in the robotic kit.
- Teams cut 5cm*5cm Squares from the Red Color Paper Sample given in the Robotic Kit and paste them over the Junctions where the Traffic Signals are RED. Paste the squares using **Transparent Tape** on the Junctions.
- Nothing needs to be pasted for a Blue Traffic signal as the Blue Markers are already printed on the arena.

c. Hardware Specifications

4.1 Use of Raspberry Pi:

- All participating teams must use **only** the Raspberry Pi B+ sent to them in the kit. **Only one** Raspberry Pi B+ is allowed per team.
- The Raspberry Pi robot programmed by the teams should be **completely autonomous**.

4.2 Components:

- Along with the Raspberry Pi, teams have been given an Edimax Wifi Dongle, a 8GB memory card, an Intex USB Camera, a Zoook 5000 mAH Power Bank, a Chassis and other accessories in the Robotic Kit.
- The teams are allowed to use only the components provided to them in the Robotic Kit. However, they are allowed to create any type of mechanical mount for mounting the camera.

4.3 Power Supply:

- The Raspberry Pi can be powered through the Power Bank that is shipped in the kit.
- The team may use alternative power sources during practice. However, during the video demonstration and final run, the team will not be allowed to use any other power source for powering the Raspberry Pi.

Note: No other expansion and/or microcontroller-based board shall be attached to the Raspberry Pi.

d. Software Specifications

- e-Yantra has provided all teams with an e-Yantra's Version of the Raspbian image that can be downloaded using the link provided in the "Installing Operating System Images on Raspberry Pi" tutorial that will be made available to the teams. This image has all the softwares and drivers pre-installed.
- The teams must use OpenCV and Python to write their code.
- You are allowed to use only inbuilt Python Libraries. Use of any other external 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.

e. Theme Rules

- The maximum time allotted to complete the task is 12 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.
- Test Image will be transferred to the Raspberry Pi prior to the start of the run. The teams will be allowed to change only the command used to import the Test Image under the supervision of an e-Yantra official. They will not be allowed to make any other changes to their code.
- Robot should be kept at the Start Point (Green Junction).
- An e-Yantra official will run the code on the Raspberry Pi, using remote login, when told to do so by the reviewer. This is the start of a run. The timer will start at the same time.
- Once the run starts, human intervention is NOT allowed.
- After reaching any Pick-Up Junction (PUJ), the robot should stop and indicate pick up of Package/s by flashing the color of the picked up Package/s thrice using the code provided by e-Yantra.
- A maximum of 4 Packages can be picked up by the Robot at any given instance.
- The Robot plans the path for deliveries and moves to the appropriate Delivery Junction to deliver the respective Packages. Delivery of Package is indicated by flashing the color of the delivered package thrice using the code provided by e-Yantra.
- After delivering all the requested Packages, the robot should print **"End of Task"** on the Python Shell and flash all four colors using the code provided by e-Yantra, to indicate the end of task.
- Robot is not allowed to traverse through the cell, it always has to follow the black line for traversal.
- A run ends and the timer is stopped when:
    - The robot stops and end of the Task is indicated 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.

- 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:
While traversing the arena if the robot strays off the black line, an e-Yantra official who is monitoring the task will place the robot on the previous node (node previously traversed by the robot prior to straying off the black line) in such a way that both the wheels of robot are parallel to the node and camera is facing the black line. 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.

NOTE: A maximum of two **Repositions** will be allowed in each run.

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.
- The 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:

Total Score = (720-T) + (CP x 50) + (CD x 50) - (IP x 30) - (ID x 30) + B – P x 20

Where:

- **T** is the total time in seconds to complete the task.
- **CP (Correct Pick-up):** When the Pick-up of a Package is correctly indicated on the output window.
- **CD (Correct Delivery):** When the Delivery of a Package is correctly indicated on the output window.
- **IP (Incorrect Pick-up):** When the Pick-up of a Package is incorrectly indicated on the output window or the pickup is not attempted.
- **ID (Incorrect Delivery):** When the Delivery of a Package is incorrectly indicated on the output window or delivered at the wrong Delivery Junction.
- **B (Bonus):** A bonus of 50 Points will be awarded only if:
  - If all the Packages are Picked-Up and Delivered Correctly
  - If no repositions are used in the run
- **P (Penalty):** Penalty will be incremented in the following cases:
  - When the robot crosses a Red Traffic Signal without halting for 30 seconds.
  - When the robot touches a Road Block.
  - When the robot is repositioned during the run.
  - When the robot traverses through the cell, instead of following the black line.

ALL THE BEST !!!