

e-Yantra Robotics Competition (eYRC-2018)**Task 1.1: Ant Bot****Objective:**

- To give you prowess in coding using open source software and libraries via the domain of Image Processing.
- To create images with particular pattern based on dictionaries of Bit Sizes and Combinations.
- To be able to identify the usage of dictionaries mentioned in the above point, so as to be able to create the desired patterns.

Structure of Task Directories:

Please find the following folders within the folder that contains this “*Read Me Task 1.1.pdf*” file. The folders have been numbered:

- **1. Resources:** You will find a “*Reading_Material.pdf*” file in this folder. This document explains the different tutorials and their use as resource material before you dive into solving the **Task1.1**. The tutorials and resources are contained in the other documents in this folder namely-
 - *ArUco_library.pdf*
 - *Getting_started_with_ArUco.pdf*
 - *Introduction_to_OpenCV_python*

Apart from these resource tutorials, additional Image Processing resources will be provided to you under the **Resources** tab on the portal. You are required to first go through these resources and exercises provided to you before you attempt the **Task1.1**.

- **2. Code:** You will find 2 code files in this folder:
 - *Aruco_lib.py*: This code file contains supporting API created by e-Yantra Team to help you learn and interface with the ArUco library of OpenCV fast.
Note: Do NOT edit any line in this file.
 - *Task1.1.py*: This code file is the one are supposed to edit to create your algorithm to solve Task1.1. However, do **NOT** edit the already created skeleton code or change the name of the functions within this code file. The program has two functions:
 - **main()**: which calls the function to generate or create ArUco markers of specified Ids.
 - **aruco_gen()**: This function expects two parameters as arguments - ArUco ID and the number of pixels in resolution. Both these inputs are whole decimal numbers.

- **3. Task_Description:** You will find the document “*Task1.1.pdf*” in this folder. This document describes the problem statement, given assumptions, inputs and data along with the required outputs and their formats for **Task1.1**. Follow this document for understanding the problem statement and design your solution for the same to get the output in the format specified in it.

Submission Instructions:

- Save the code that you generated to solve the problem in a folder named “**Code**”. Also, save the “*Aruco_lib.py*” you used in conjunction with your algorithm code in this folder.
- Save the generated output images in a folder named “**Images**”
- Save both these folders within a folder named “**<TeamID>_Task1.1**” where if your team ID is 1001, then the folder name will be **1001_Task1.1**
- Compress the folder into a **.zip** file and upload it within a week as your submission.
Note: Task1.1 should be uploaded on the portal on or before **11:59 pm, 14th November 2018.**
- Ensure your zip folder is **less than 5MB** in size.

Note: Do **NOT** edit the skeletal of the code (function names, indentations, etc.) in “*Task1.1.py*” and “*Aruco_lib.py*”. The files submitted by you will be run through a test script for automatic grading. **Teams making any changes will be disqualified.**

Warning:

- **IMPORTANT:** The document you submit should be **YOUR WORK** in **YOUR WORDS**. To avoid any copyright violations, you must **NOT** copy phrases directly from manuals or web.
- The team should **NOT** mail or upload the document anywhere else, except on the e-Yantra portal.
- Teams failing to submit the document by the deadline will lose the marks for this task.
- e-Yantra **WILL NOT** entertain any request for an extension of the deadline for uploading the task.
- e-Yantra has complete discretion to disqualify a team if any foul play is suspected.