CS684: Embedded System Course

Rulebook: [1] Theme Description

- Figure 1 shows the arena design for this theme.
- The arena is an abstraction of a disaster-affected area made up of a grid with the **START** position marked.
- The grid is made up of **16 squares** called **Plots**.

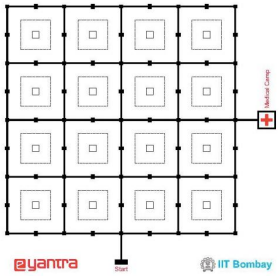

Figure 1: Arena design

---

1. Arena Components

- Each Plot has the following terms associated with it:
  - **Four Mid-Point Markers** on every path around the Plot. Figure 2a highlights the Mid-Point markers for a Plot.
  - The **Clearing Zone**, which is shown by the dotted square of **26cm x 26cm**, highlighted in the green box in Figure 2b.
  - A **6cm x 6cm Inner Square** that is highlighted in Figure 2c.
  - **Four Nodes** present at the corners of every Plot. This is shown in Figure 2d.

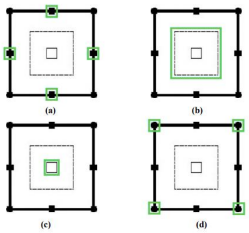

Figure 2: Terms associated with Arena

- Rectangular chart paper patches are used to represent **injured survivors** and **debris**.
- Three colored patches:
  - **Red** - represent **Survivors with Severe Injuries**, those require **urgent assistance**
  - **Green** - represent **Survivors with Minor Injuries**, those require **not so urgent assistance**
  - **White** - represent pieces of **Debris** strewn on the roads which cannot be moved, thus causing a roadblock

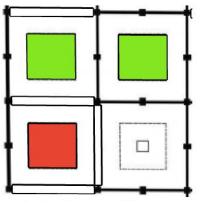

Figure 3: Injured Survivors and Debris on Arena

---

2. Theme Run Sequence

- The robot will start from the **START** position of the arena.
- It must traverse around the arena avoiding debris and check for the presence of a Survivors in each plot.
- If Survivor is present in the plot, robot detects the type of emergency (Minor or Severe) using color.
  - Communicates the presence and type of Survivor to Desktop/Laptop.
- During scanning, robot will get requests from the server on regular interval of 45 seconds.
  - Robot satisfies the requirement by traversing to the appropriate plot and ringing a buzzer for 1 second. **Note:** This indication can be done only from one of the Mid-Point Markers associated with that Plot.
- Once robot scans entire grid, it should stop at medical camp to mark an end of the run.

---

3. Communication Sequence

- Robot (Firebird V) will communicate with ESP 32 module over UART using serial communication.
- ESP32 will use Bluetooth Low Energy (BLE) to communicate with the internet-connected laptop. Note that ESP32 won't be connected to the Wifi and hence the internet.
- Laptop will communicate with the Thingsboard server.

4. Theme Requirements

Input Operations

- Thingsboard server will send RPC requests to the laptop on a regular interval of **45 seconds**. We will provide you with a script to mock this behaviour in development.
- Laptop receives the RPC requests using MQTT Protocol in JSON format.
- The requests may contain single or multiple requests (actions to perform). Robot can decide to satisfy the requests or ignore them.
- There can be different types of requests as given below:
  - Fetch RED Survivor in 10s: Robot has to travesrse to the nearest RED Survivor plot and ring the buzzer within 10 second to satisfy the requirement.
  - Identify Survivor at plot 4 in 20s: Robot has to traverse to plot 4, identify the Survior and ring the Buzzer within 20 seconds.
  - No Request: No action needs to be performed

Output Operations

- Once a request is performed by the robot, the response should be sent by the robot to the laptop over BLE and from laptop to the Thingsboard server. (The exact request and response formats are present under resource section (`MockingRPC`) **[Updated April 6]**).
- The data should be sent as **telemetry** and the protocol to use while sending the data back to Thingsboard server will be CoAP. **[Updated April 6]**
- Teams should prepare an interface (desktop app or web app) to show the requests received, indicate operations being performed on the arena and result of the operations. This is an open ended task so use your creativity.

CS684: Embedded System Course

Rulebook: [2] Arena

---

1. Arena Configuration

- For ease of reference:
  - The Plots in the arena are numbered from **1** to **16** as illustrated in red in Figure 1a.
  - The Nodes in the arena are number from **1** through **25** as illustrated in Figure 1b.

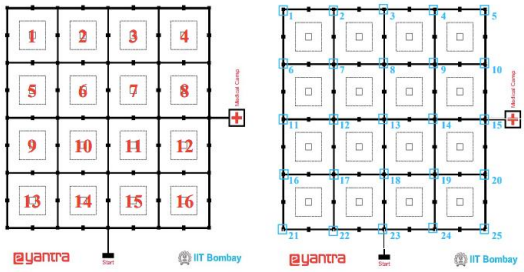

Figure 1: Arena design

---

2. Preparing the Debris and Injured Survivors

- Material required:
  - Red, Green and White color chart paper
- Preparing Debris and Injured Survivors:
  - Teams will prepare three kinds of chart paper patches. Characteristics of these are given in Table 1.

Type of Patch	Length (cm)	Width (cm)	Color	Count
Debris	40	6	**White**	10
Survivors with Major Injuries	26	26	**Red**	10
Survivors with Minor Injuries	26	26	**Green**	10

Table 1: Characteristics of Debris and Injured Survivors

- After preparing these patches, teams must paste them on the arena according to the final arena setup given below.

---

3. Placing the Debris and Injured Survivors on Arena

- Placement of Debris:
  - White patches of size 40cm x 6cm are used to indicate the Debris.
  - These patches need to be sticked on the Path between two nodes and covering Mid-point Marker.
- Placement of Injured Survivors:
  - Green and Red patches of size 26cm x 26cm are used to indicate minor and severly injured survivors.
  - These colored patches need to be sticked in the clearing zone.

Note: You must use transparent sellotape to stick these patches.

---

4. Final Arena Setup

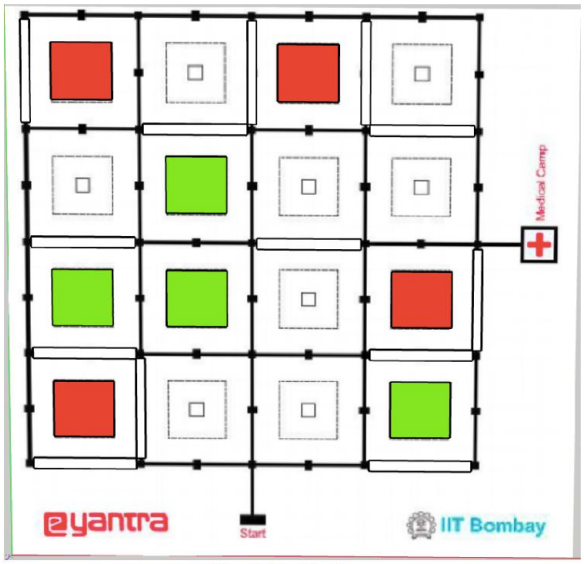

Figure 2: Arena Final Setup

Note: The position of Debris and Survivors is random. During final demonstration, position will be different. Your algorithm should be generic.

CS684: Embedded System Course

Rulebook: [3] Hardware and Software Specifications

---

1. Hardware Specifications

- **Use of Firebird V and ESP32:**
    - All teams must use Firebird V robot and ESP32 development board provided by e-Yantra.
    - 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 devices such as a camera while the robot is performing the task. The robot are only allowed to communicate using the Wireless Protocol mentioned.
- **Use of additional components:**
    - Firebird V and ESP32 communicate using UART. No other microcontroller-based board shall be attached to the Firebird V robot except ESP32.
    - Teams are not allowed to connect external actuators or structural hardware to the Firebird V robot.
    - Teams are not allowed to use any additional sensor apart from ones provided with the Robot.
- **Power Supply:**
    - The robot can be charged through battery or auxiliary power supply.
    - 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 battery powered robot.

---

2. Software Specifications

- Teams can use Eclipse, for writting code for AVR microcontroller. Teams are free to use any other open source Integrated Development Environment (IDE) for programming AVR microcontroller.
- Use of any non-open source libraries is not allowed and will result in disqualification.
- Teams can use any language and open source libraries for creating an interface (desktop app or web app).

CS684: Embedded System Course

Rulebook: [4] Theme Rules

Final Run

- The maximum time allotted to complete the task is 10 minutes.
- A maximum of two repositions (explained below) is allowed for each team.
- Once the robot is switched on, human intervention is NOT allowed.
- Robot is not allowed to traverse through the Plot, it always has to follow the black line for traversal.
- For the final demonstration, the Arena Configuration will NOT be given to any of the teams.The robot must navigate through a randomly setup arena and autonomously detect the White Debris, identify the Survivors without any prior knowledge of the arena configuration.
  Note: You MUST have a generic solution that can handle any setup, in real-time.
- Any of the 16 Plots may contain survivor.
- Debris may be placed on path of any of the Plots such that every Plot in the arena will have at least one path for the robot to reach to the Survivor.
- A run ends and the timer is stopped when:
  - The Robot stops at medical camp or
  - If the maximum time limit for completing the task is reached or
  - If the team needs repositioning but has used all repositioning options.

Repositioning the Robot

Suppose while traversing the arena, the robot strays off the black line, team member can place the robot on the previous node (node already traversed by the robot) by dragging (not lifting) the robot back to the line in such a way that both the wheels of robot are parallel to the node and castor wheel is on 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.

CS684: Embedded System Course

Rulebook: [5] Judging and Scoring System

$$Score = (600 - T) + (30 * CDS) + (\sum_{i=1}^{n}(GT_i - TT_i)) * 25 + (10 * NR) + AB - (P * 30)$$

Parameters:

- *T* : Time taken by robot to complete the Task
- *CDS* : Number of Survivors detected correctly
- GT_i : Given time for the ith request
- TT_i : Time taken by robot to complete the ith request
- *NR* : Number of requests satisfied by the robot
- *AB* : Bonus of 100 is provided based on the creativity shown in designing desktop/web app
- *P* : Penalty is incurred each time robot dashes against Debris and for each Reposition.
