Task 3A - Socket Communication with the Bot

Objective:

The objective of this task is to establish socket communication with the bot over a Wi-Fi link.

Given:

Before starting working on this task, you have to get well versed with the following tutorials provided on the **Quick Bytes** page of the website: https://e-yantra.org/products/eyfi-mega.

1. eYFi Wireless Serial Terminal for ATmega 2560

   Description of how to implement and test it is provided in the section *eYFi-Mega Wireless Serial Terminal Application* of Software Manual of the eYFi-Mega board.

2. Communication - UART between ATmega 2560 and ESP32

   Description of how to implement and test it is provided in the *README.md* document of the zip file of the project.

3. Communication - Socket Echo to Python over Wi-Fi

   Description of how to implement and test it is provided in the *README.md* document of the zip file of the project.

Note: Before you start working on Task 3A, you need to prepare two applications:

- **robot-server-esp32.c:** This is a ESP32 Robot Server application whose purpose is to:
  - Start Wireless Access Point
  - Receive commands from Python Client over socket
  - Fetch sensor data from ATmega 2560 via UART
  - Serve data of VL53L0X sensor to the Python Client via socket
- **robot-sensors-atmega2560.c:** This is a ATmega 2560 application whose purpose is to:
  - Receive commands from ESP32 via UART
  - Get readings of VL53L0X sensor in millimeters
  - Serve these readings to ESP32 when asked for

You may refer the above three Tutorials to write your codes for this task.

Arena Preparation:

In order to start the task, first arrange the walls in the following manner, starting from START (0,0) and ending at (0,9) as shown in the Figure 1.

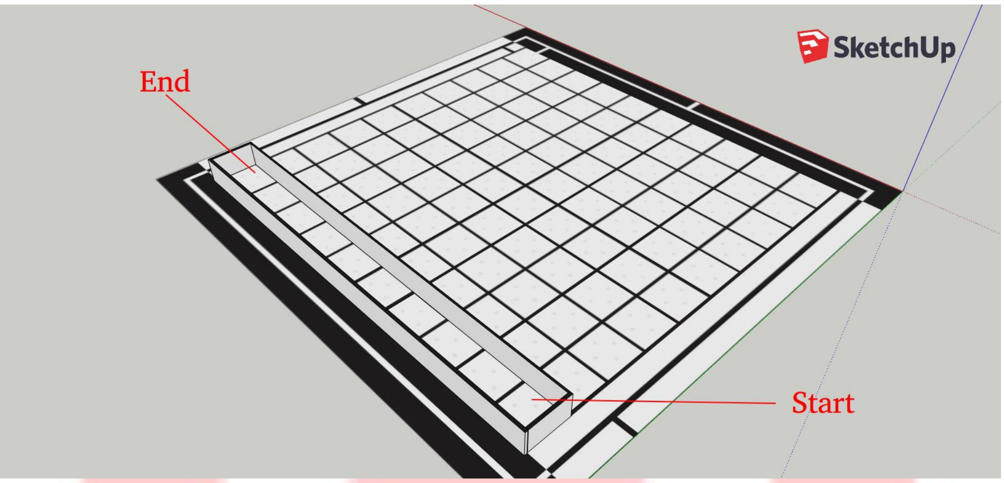


Figure 1: Wall placement for Task 3A

Problem Statement:

The task is to first, establish socket connection between Robot Server (ESP32) and Python Client over Wi-Fi and then make the robot perform certain tasks based on the commands sent to it by the Python Client.

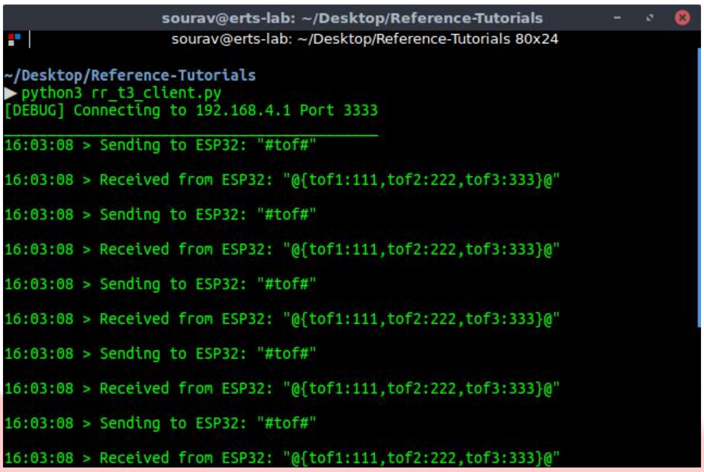
Following steps need to be completed in order to successfully complete Task 3A:

- First keep the robot at (0,0) and make it face towards the end wall. Turn on the robot.
- Connect your laptop to the Wireless Access Point created by the ESP32 Robot Server.
- Once you are connected to the robot, run the Python Client script **rr_t3_client.py** using the command: "**python3 rr_t3_client.py**". You must run the file inside the conda environment created in Task 0.
- The Python Client once connected to the ESP32 Robot Server via socket will start sending: **#tof#** contiuously until **Ctrl + C** is pressed.
- On receiving this, the ESP32 Robot Server application should take all three VL53L0X sensor readings in **millimeters** from ATmega 2560 application and send it to Python Client in the following format:

  @{tof1:<tof1_reading>,tof2:<tof2_reading>,tof3:<tof3_reading>}@

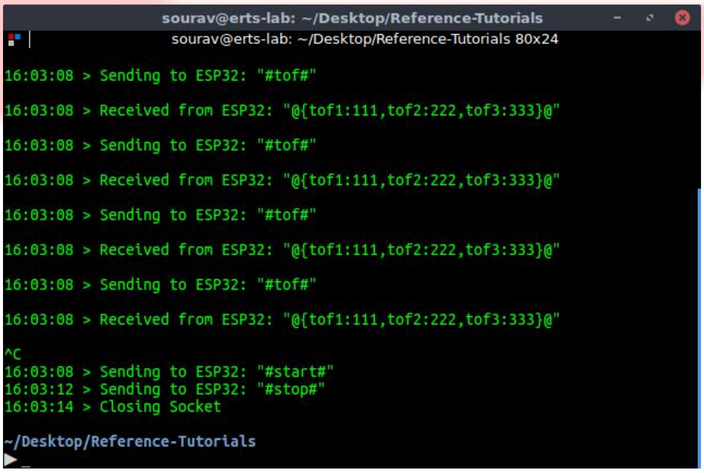
  For example,

  @{tof1:111,tof2:222,tof3:333}@


sourav@erts-lab: ~/Desktop/Reference-Tutorials
sourav@erts-lab: ~/Desktop/Reference-Tutorials 80x24

```
~/Desktop/Reference-Tutorials
▶ python3 rr_t3_client.py
[DEBUG] Connecting to 192.168.4.1 Port 3333

16:03:08 > Sending to ESP32: "#tof#"

16:03:08 > Received from ESP32: "@{tof1:111,tof2:222,tof3:333}@"

16:03:08 > Sending to ESP32: "#tof#"

16:03:08 > Received from ESP32: "@{tof1:111,tof2:222,tof3:333}@"

16:03:08 > Sending to ESP32: "#tof#"

16:03:08 > Received from ESP32: "@{tof1:111,tof2:222,tof3:333}@"

16:03:08 > Sending to ESP32: "#tof#"

16:03:08 > Received from ESP32: "@{tof1:111,tof2:222,tof3:333}@"

16:03:08 > Sending to ESP32: "#tof#"

16:03:08 > Received from ESP32: "@{tof1:111,tof2:222,tof3:333}@"
```

- Team has to turn the robot around 360 degree slowly by hand at the START position only and send all three sensors data continuously to the Python Client.
- Once done, the robot should be placed again at the START position facing towards the end wall.
- Then on the Python Terminal, press **Ctrl + C**. As soon as you do this, the Python Client will send: **#start#** only once to the ESP32 Robot Server application.
- On receiving this, the robot should start traversing forward whilst **avoiding the walls** placed adjacent to it. **Note:** The duty cycle of running the motors should be **60%** only, so that it does not dash into the end wall.
- Exact after **4 seconds**, the Python Client will send: **#stop#** once to the ESP32 Robot Server application and closes its socket connection.
- On receiving this, the robot should stop traversing at the position it is.


sourav@erts-lab: ~/Desktop/Reference-Tutorials
sourav@erts-lab: ~/Desktop/Reference-Tutorials 80x24

```
16:03:08 > Sending to ESP32: "#tof#"

16:03:08 > Received from ESP32: "@{tof1:111,tof2:222,tof3:333}@"

16:03:08 > Sending to ESP32: "#tof#"

16:03:08 > Received from ESP32: "@{tof1:111,tof2:222,tof3:333}@"

16:03:08 > Sending to ESP32: "#tof#"

16:03:08 > Received from ESP32: "@{tof1:111,tof2:222,tof3:333}@"

16:03:08 > Sending to ESP32: "#tof#"

16:03:08 > Received from ESP32: "@{tof1:111,tof2:222,tof3:333}@"

^C
16:03:08 > Sending to ESP32: "#start#"
16:03:12 > Sending to ESP32: "#stop#"
16:03:14 > Closing Socket

~/Desktop/Reference-Tutorials
▶ _
```

- Once the Python client closes its socket connection, it will generate the encoded text file named: **task_3a_output.txt**. This file contains all the data received from the ESP32 Server application on the robot by the Python Client.

Video:

- Once you are done practicing and are ready with complete run of Task 3A as mentioned in above steps, you have to record a demonstration video.
- Following should be clearly visible in the video:
  - Output of the Python terminal running the file **rr_t3_client.py**
  - Robot turned 360 degrees by hand at START position and sending readings of all three sensors to Python Client
  - Robot traversing forward from centre of START cell for 4 seconds and stopping
- Make sure you use camera of resolution **more than 5 MP** to record the video. You can take the screen capture of the Python terminal output on the laptop and the robot traversing on the arena from the camera separately and later merge them.
- While recording the video, one team member may explain the entire implementation of Task 3A. Maximum time limit of video: 1 minute.
- Upload the video on YouTube as **Unlisted** and provide the link on the portal under the **Task 3** page.
- Instructions for submitting the video link are mentioned on the portal.

For this Task 3A, you have to submit the following:

A. complete project folder of ATmega 2560 application created in eY-IDE with the source file named: **robot-sensors-atmega2560.c** and its compiled binary files inside the *build* folder. The name of the project folder will be same as the source file: **robot-sensors-atmega2560**.

B. the source file of ESP32 Robot Server application created in eY-IDE named:

robot-server-esp32.c

Note: You are not supposed to provide any compiled binary files of ESP32 Robot Server application, only the source file needs to be submitted.

C. the encoded text file generated: **task_3a_output.txt**

After completing all the steps successfully, Task 3 will be considered complete.

NOTE: Hard coding in any of the steps described in the Problem Statement will result in penalty.