**Testing Note: If you are running the application from a remote SSH terminal, then you must run “export DISPLAY=:0” to redirect the QT GUI output to a display device connected to the Raspberry Pi. DISPLAY=:0 is the default output device detected by the RaspberryPi on startup.**

**Test Case 1**: Connection Establishment

**Objective**: Verify the successful connection establishment between Publisher (server) and Subscriber (client).

**Acceptance Criteria (AC):**

* Publisher and Subscriber are operational. - **Passed**
* DDS middleware is successfully initialized on both sides. -**Passed**
* Both Publisher and Subscriber join the DDS domain without errors. **Passed**

**Expected Result:**

* Both Publisher and Subscriber applications run without any errors.
* DDS middleware successfully establishes domain participant connections.

**Testing Result** : Below two diagram are showing Publisher and Subscriber Running without error and both joined over DDS, its showed subscriber matched comment on subscriber window.

**Here Publisher Running without error.**

A screenshot of a computer program

AI-generated content may be incorrect.

**Here publisher will run with main program -> ./smart\_system**

A screen shot of a computer

AI-generated content may be incorrect.

**Here when publisher running without error:**

A screen shot of a computer

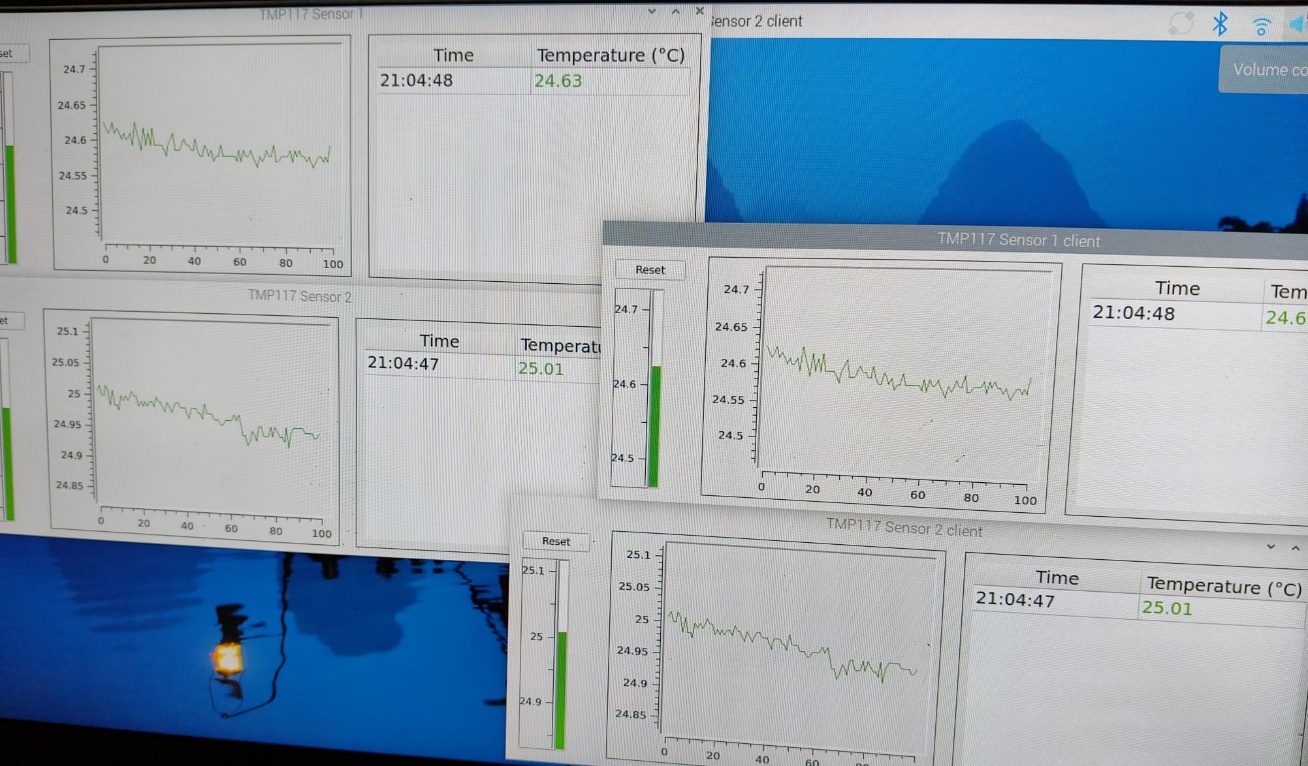
AI-generated content may be incorrect.

**Subscriber running without error and matched with publisher:**

A screenshot of a computer screen

AI-generated content may be incorrect.

After successful connection It will display the temperature in both terminal Publisher and Subscriber check below:



**Test case 1 Result -Passed**

**Test Case 2**: Server-to-Client Display Transmission

**Objective**: Validate that the server-side display data reaches the client side accurately.

**Acceptance Criteria (AC):**

* The Publisher (server) display data is successfully transmitted to the Subscriber
* The Subscriber displays exactly match the server-side displays.

**Expected Result:**

* Client-side display reflects server-side display data accurately and promptly.
* Test Case Result: Test Case 3: Event-Driven Data Synchronization

**AC Result - Passed**

A screenshot of a computer

AI-generated content may be incorrect.

Publisher and Subscriber Program Running on in same Machine

A screenshot of a computer

AI-generated content may be incorrect. Subscriber is Running on Another machine

**Testcase – 3**

**Objective:** Ensure any server-side display changes reflect immediately on the client side through DDS.

**Acceptance Criteria (AC):**

* Publisher detects event-driven changes (like temperature threshold).
* Publisher immediately publishes updated data.
* Subscriber accurately and immediately reflects the updated data without acknowledgment.

Testing Result : Passed

A group of papers with graphs

AI-generated content may be incorrect.

**AC Test Result : Passed** when Publisher and subscriber are running on same machine and result shows both temperature sensor crossing threshold limit.

A close-up of a graph

AI-generated content may be incorrect.

**Testing Result** : Passed , here subscriber running in another machine: both temperature windows showing changed value as per publisher change proving that publisher and subscriber synchronizing properly.