Use Case for Practical Work (v2025/26)

Context:

The COMCS.Lda organization is a company that sells a range of products manufactured in its industrial units and stored in a centralized warehouse. Given the sensitivity of the products, it will be necessary to ensure that the different stages of production, storage, and transport maintain constant levels of temperature and humidity. Small fluctuations can compromise product quality. Identifying the source of problems is fundamental to ensuring quality and customer satisfaction. To respond quickly to this need, you should develop a solution that ensures constant monitoring of the indicators mentioned.

Expected Outcomes:

1. **Command Centre**:
    a. Implementation of a command centre (using Node-RED) for a specific work cell/warehouse site.
    b. Installation of one ESP32 and one Raspberry Pi Pico device with DHT11 sensors in the cell, sending temperature and humidity data every second.
    c. Implementation using the MQTT protocol.
    d. Display the average temperature and humidity values in the cell.
    e. Display of alerts, e.g. when temperature rises above 20º or below 10% (use values which allow you to test this feature).
2. **Alert Server**:
    a. Use of an UDP server (can be a virtual machine) for managing alerts.
    b. Implementation of best effort / guaranteed delivery QoS policies on UDP communications.
        i. Present tests showing that your protocol works correctly
    c. Ability to respond to multiple clients.
    d. Receiving readings from clients (current value and the average), applying temperature and humidity differential calculations, i.e. calculating the difference in temperature and humidity obtained between the current and the average received. Validate if the values obtained are within the range of use of the equipment (temperature between 0 and 50ºC and humidity between 20 and 80%).
    e. Generating alerts (log/screen) when the differential exceeds the error rate (temperature~=2º and humidity~=5%) and/or the range of the equipment's.
    f. Sending alerts via MQTT to the command centre.
3. **Clients for monitoring using ESP32 and RaspberryPI devices**:
    a. The ESP32/RaspberryPI devices should be installed at strategic points in the work cell, in order to monitor the temperature and humidity every second.
    b. Data transmission via MQTT protocol to the command centre and via UDP to the Alert server.
    c. Use the smartdata model message format for temperature and humidity. https://github.com/smart-data-models/dataModel.Weather/blob/master/WeatherObserved/doc/spec.md
4. **Status Rolling Window using the ESP32/RaspberryPI**

a. In case of communications failure, a rolling window must be implemented using a file in the SPPIFS area to store temporarily the telemetry and sent to the UDP server and command centre when communication is re-established.
b. Error handling must be implemented for the communications.
c. In case of a restart of the ESP32/RaspberryPI device, the data in the file must be transmitted to the server and command centre.

Acceptance Criteria:

1. **Command Centre (Node-RED MQTT)**
   a. Command centre implemented using Node-RED.
   b. Average temperature and humidity reception and display, instant and historical graphs should be displayed.
   c. Alert indicator.
   d. Communication system using smartdata models
2. **Alert Server**
   a. UDP server (can be a Virtual machine).
   b. Implementation of QoS policies such as best effort / guaranteed delivery.
   c. Able to respond to multiple clients.
   d. Adequate tests for a and b.
   e. Receiving client readings and calculating the temperature and humidity differential.
   f. Sending alerts via MQTT to the command centre when the differential exceeds the error rate.
   g. Communication system using smartdata models

3. **ESP32/RaspberryPI**
   a. Communication system using smartdata models
   b. Rolling window area for last status
   c. Error handling for the communications.

Additional Valuation Criteria:

1. Implementation of redundant communication via BLE.
2. Implementation of additional QoS mechanisms.
3. Failure detection in system components.

Deliverables:

1. Code produced for the different components
2. Report
   a. Components
   b. Compilation
   c. Installation and Setup

The work should be completed in groups of two and demonstrated on a date to be designated. The report should be concise and focused (not more than 4 pages, 1 column). The demonstration can be supported by slides describing the architecture and other details of the work.

Materials:

1. Virtual Node-RED server.
2. UDP server.
3. ESP32 & Raspberry PI Pico
4. DHT11 x2

Note:

Students may submit their own work proposal to the course instructors before starting the work. This proposal must have a level of complexity equivalent to or greater than the one presented above.