

Descrizione del Progetto

Il progetto si concentra sulla realizzazione di un sistema in grado di misurare la temperatura di una stanza, elaborarla e regolare di conseguenza il livello di apertura di una finestra, il tutto monitorato da una dashboard. Il sistema è composto da quattro sottosistemi che comunicano tra loro sincronizzando i dati.

Control Unit

Il sottosistema principale è la Control Unit, realizzata in linguaggio Java utilizzando il framework Vert.x. Questa si compone di quattro **verticles** che lavorano contemporaneamente e di un oggetto che rappresenta la Control Unit, il quale serve per contenere e leggere i dati.

I servizi eseguiti comprendono:

- **MqttAgent**: responsabile della comunicazione con l'ESP tramite il protocollo MQTT. Utilizza due topic: uno per ricevere la temperatura e uno per inviare la frequenza. Quando riceve una temperatura, la comunica alla Control Unit, che la salva in un data store, la elabora e aggiorna lo stato della macchina a stati finiti. Se cambia stato, invia la nuova frequenza all'ESP.
- **AlarmTimer**: imposta un timer con Vert.x che si riavvia ogni volta che si entra nello stato TOO_HOT. Alla scadenza, verifica se la condizione persiste per gestire lo stato di allarme.
- **SerialService**: esegue periodicamente due operazioni principali:
 - **Ricezione**: monitora un buffer periodicamente e, se rileva un messaggio, ne estrae i valori e li comunica alla Control Unit.
 - **Invio**: verifica se è presente un messaggio nella coda di invio. Se la coda contiene dati, il servizio li trasmette sulla seriale.
La Control Unit utilizza il pattern **Observer** per notificare il SerialService ogni volta che un nuovo messaggio deve essere inviato, riempiendo così la coda di invio.

Comunicazione con la Dashboard

Vert.x mette a disposizione un server HTTP che riceve richieste GET e POST dalla dashboard. Grazie all'uso di **Ajax**, la dashboard aggiorna i dati dinamicamente senza ricaricare la pagina. Tramite l'interfaccia grafica è possibile:

- Selezionare il numero di record di temperatura da richiedere alla Control Unit, con calcolo di media, massimo e minimo.
 - Visualizzare un grafico temperatura-tempo.
 - Monitorare e modificare i valori correnti del sistema (temperatura, apertura finestra e modalità di funzionamento), sempre sincronizzati con Arduino.
-

Window Controller (Arduino)

Il sottosistema Window Controller è basato su **Arduino** e simula un sistema operativo tramite uno **scheduler** che esegue periodicamente diverse task:

- **ReceivingTask**: riceve dati dalla Control Unit.
 - **SendingTask**: invia dati alla Control Unit.
 - **ModalityTask**: gestisce la modalità MANUAL/AUTOMATIC.
 - **WindowControlTask**: regola l'apertura della finestra in base allo stato del sistema. Lo stato del sistema viene inoltre mostrato su un display LCD, informando l'operatore sulla temperatura attuale, il livello di apertura della finestra e la modalità attiva.
-

Temperature Monitoring (ESP32)

Il sottosistema che raccoglie la temperatura della stanza è realizzato su un **ESP32**, che utilizza **FreeRTOS** per gestire più task contemporaneamente. Questo garantisce un'operazione efficiente delle attività di rete e di acquisizione dati.

Il codice è strutturato in quattro **task principali**:

- **termoTask**: acquisisce la temperatura dal sensore.
- **wifiTask**: gestisce la connessione WiFi.
- **mqttTask**: mantiene la connessione al broker MQTT.
- **sendTask**: invia i dati al broker MQTT e gestisce lo stato dei LED.

L'esecuzione parallela viene ottimizzata assegnando le task ai core del processore:

- **termoTask** gira sul core 0.
- **wifiTask**, **mqttTask** e **sendTask** girano sul core 1.

Per evitare sprechi di tempo CPU, è stato utilizzato **xTaskDelayUntil** al posto di **delay()**, bloccando la task a livello di scheduler.

Ottimizzazione della Comunicazione

Per ridurre il traffico sui canali di comunicazione:

- I messaggi tra i dispositivi vengono inviati solo quando i valori vengono modificati.
- Arduino invia dati sulla seriale solo quando cambia stato o modifica il valore della finestra.
- La Control Unit esegue periodicamente un controllo su una coda per verificare la presenza di messaggi da inviare.

Questa gestione garantisce una comunicazione efficiente, evitando congestioni e sprechi di risorse.