DOCUMENTAZIONE DI PROGETTO

*3* *assignment*

*IoT*

Mattia Flamigni - Mattia.flamigni4@studio.unibo.it

Filippo Badioli - [filippo.badioli@studio.unibo.it](mailto:filippo.badioli@studio.unibo.ot)

**WATER LEVEL MONITORING(ESP):**

Il sottosistema in questione è incaricato di monitorare costantemente il livello dell'acqua ad intervalli di tempo definiti dalla frequenza F, trasmettendo i dati al server del servizio di monitoraggio (river monitoring service).

La frequenza F è rappresentata attraverso una variabile globale. All'interno del sistema, tale variabile è inizialmente configurata nel file config.h con un valore predefinito di 500 millisecondi. In seguito, la classe MqttPub assume la responsabilità di aggiornare la variabile con il valore ricevuto tramite Mqtt sul topic relativo alla frequenza, garantendo così una flessibilità nella gestione della frequenza di campionamento del livello dell'acqua.

**RIVER MONITORING SERVICE**

il seguente sottosistema è stato implementato in Java e utilizza la connessione HTTP, Mqtt e seriale.

Il sottosistema ha 3 principali responsabilità:

determinare lo stato del livello del fiume:

Gli stati che può assumere il sistema sono stati definiti nella classe WaterLevelState utilizzando la struttura dati enum.

La classe WaterLevelSensor ha il compito di aggiornare lo stato attuale del fiume e di determinare la frequenza di lettura del livello del fiume.

Determinare il grado di apertura delle valvole:

Il grado di apertura delle valvole viene invece stabilito dalla classe ValveController a seconda dello stato ricevuto.

Ricezione elaborazione e invio dei dati

La classe MqttSubscriver si occupa di ricevere il valore del livello del fiume, elaborare il dato utilizzando le classi sopra citate e inviare ad Arduino il grado di apertura delle valvole tramite la porta seriale

Le classi con iniziale HTTP hanno il compito di leggere e inviare alla dashboard i valori utili per il suo funzionamento tramite connessione HTTP

**WATER CHANNEL CONTROLLER**

il seguente sottosistema lavora in due modalità diverse:

modalità automatica:

viene letto il valore inviato sulla seriale dal server e utilizzato per impostare il grado di apertura della valvola

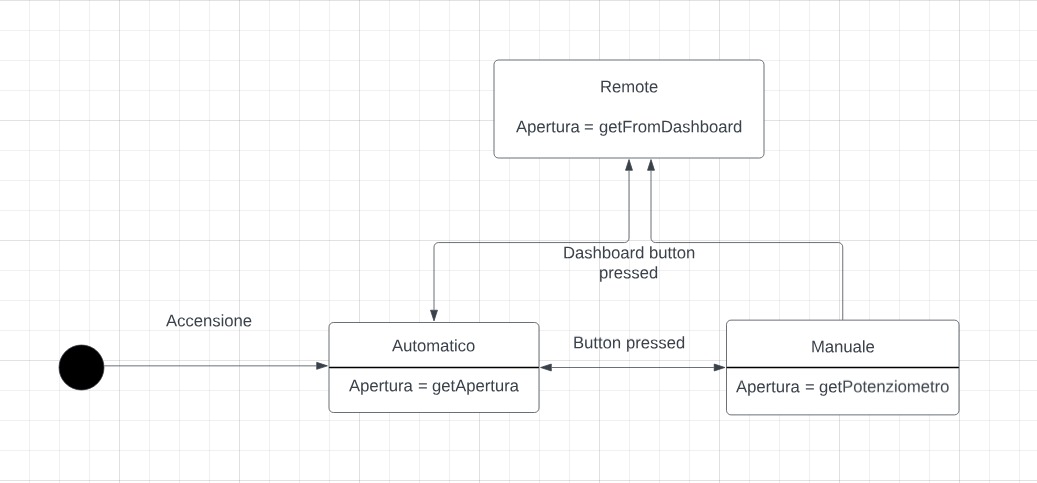
modalità manuale:

attraverso un potenziometro è possibile determinare il valore di apertura delle valvole indipendentemente dal valore inviato sulla seriale

il passaggio di stato è possibile utilizzando un apposito bottone.

Il display installato ha il compito di visualizzare il grado di apertura delle valvole corrente e la modalità

Il sistema è stato implementato utilizzando un singolo task e la F.S.M. con i due stati descritti



N.B ci sarebbe una modalità aggiuntiva (remote) per il settaggio delle valvole da remoto. Tuttavia, a causa di errori di progettazione che hanno portato a uno spaghetti code nel server, numerose prove d’esame nell’appello corrente e la quasi sessione completa non siamo riusciti a implementare quest’ultima funzionalità

**RIVER MONITORING DASHBOARD**

Il seguente sottosistema è stato realizzato in java, grazie a librerie di swing, java.io, java.net e org.jfree.

La dashboard consiste in una GUI, la quale mostra in tempo reale un grafico con il livello d’acqua degli ultimi minuti.

Mostra inoltre, lo stato attuale del Water Channel Controller e il livello di apertura dei gate.

Questi dati vengono presi grazie ad un client HTTP che riceve periodicamente messaggi dal Water Monitoring Service.  
Il grafico è realizzato mediante le librerie di org.jfree.chart.

N.B. come nel Water Channel Controller ci sarebbe dovuta essere una modalità per controllarlo da remoto ma non siamo riusciti ad implementarla.