

Assignment 03 – Smart River

Giacomo Foschi

giacomo.foschi3@studio.unibo.it

Martino Narducci

martino.narducci@studio.unibo.it

Gioele Santi

gioele.santi2@studio.unibo.it

Architettura

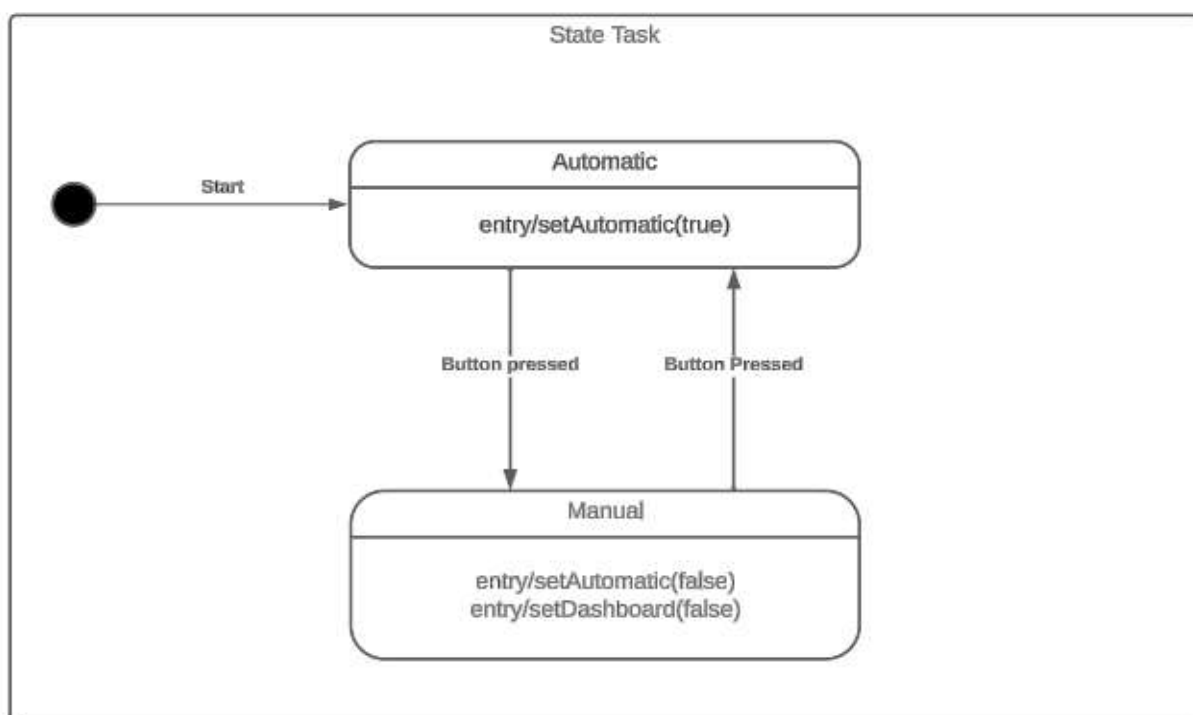
La soluzione proposta per l'Assignment 03 – Smart River Monitoring si basa su quattro ambienti principali: Arduino, Java, modulo Esp, e una dashboard. Il sistema è progettato per operare in due stati principali: "automatico" e "manuale". La transizione tra questi stati avviene in risposta alle operazioni effettuate sulla dashboard o su Arduino.

1.1 Arduino

Arduino, nel sottosistema Water Channel Controller, agisce come il cuore operativo del sistema Smart River Monitoring. Con una Finite State Machine (FSM), Arduino gestisce gli stati del sistema, controllando la valvola tramite un servo motore in risposta alle istruzioni del River Monitoring Service. La FSM permette una transizione fluida tra modalità automatica e manuale, consentendo agli operatori di intervenire direttamente. La comunicazione bidirezionale con il River Monitoring Service tramite una linea seriale assicura una risposta tempestiva alle variazioni nel livello dell'acqua, garantendo il corretto funzionamento del sistema.

1.1.1 State Task

Il sottosistema del Water Channel Controller su Arduino è stato implementato utilizzando una macchina a stati finiti (FSM). Questo approccio permette una gestione efficiente degli stati del sistema, garantendo una transizione controllata tra gli stati automatico e manuale. La FSM svolge un ruolo fondamentale nel decidere il comportamento del sistema in base alle informazioni ricevute dal River Monitoring Service.



1.1.2 Valve Task

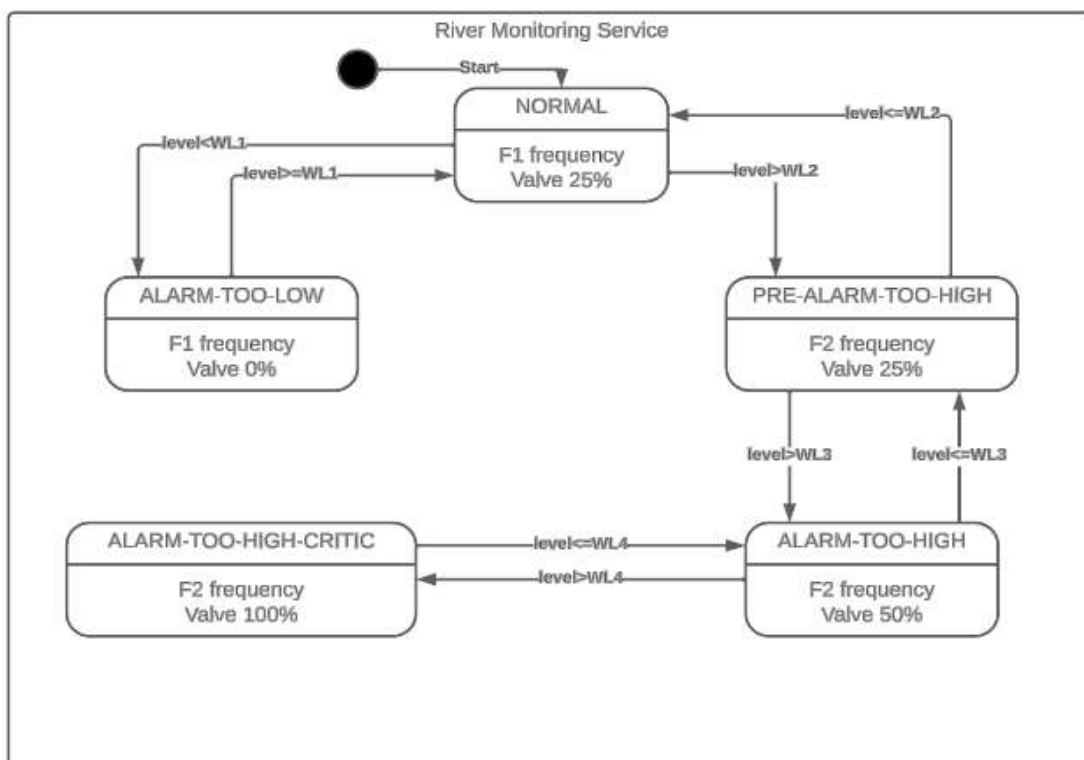
La componente Valve Task, gestita dall'Arduino, controlla il servo motore che regola l'apertura della valvola. La posizione del servo è determinata dal livello di apertura desiderato, che può variare da 0% a 100%. La possibilità di transizione verso la modalità manuale permette agli operatori di controllare manualmente l'apertura della valvola utilizzando un potenziometro.

1.1.3 Serial Task

La componente Serial Task si occupa di gestire le interazioni con il back-end sulla linea seriale. Legge i messaggi in ingresso e fornisce al controller lo stato corrispondente che verrà utilizzato dal sistema in modo adeguato. Inoltre, prepara i messaggi da inviare all'unità centrale e li spedisce sulla linea seriale con una formattazione adeguata.

1.2 Java

Il River Monitoring Service, implementato in Java e eseguito su un PC server, è il cuore del sistema. Utilizza una FSM per gestire il comportamento globale del sistema in risposta al livello dell'acqua rilevato dal Water Level Monitoring subsystem. La politica del sistema, definita in base a intervalli di livelli d'acqua, determina le azioni da intraprendere



1.3 Esp32

Il Water Level Monitoring subsystem, eseguito su un modulo ESP32 o ESP8266, utilizza un sonar per monitorare continuamente il livello dell'acqua nel fiume. I dati vengono inviati al River Monitoring Service tramite MQTT con una frequenza determinata dallo stato del sistema.

1.4 Dashboard

La River Monitoring Dashboard, eseguita su un PC, è l'interfaccia utente principale per visualizzare lo stato del sistema. Presenta un grafico del trend del livello dell'acqua e indica lo stato attuale del sistema (NORMAL, ALARM-TOO-LOW, PRE-ALARM-TOO-HIGH, ALARM-TOO-HIGH, ALARM-TOO-HIGH-CRITIC). La dashboard è scritta in python e comunica con il back-end tramite richiesta http, utilizzando il formato Json.