

# RELAZIONE PROGETTO#3-Smart Dam

È stato realizzato un sistema embedded che implementa una versione semplificata di un sistema intelligente per il monitoraggio del livello idrometrico di un fiume e il controllo di una diga.

Il sistema proposto si compone di cinque sottosistemi che comunicano tra loro tramite scambio di messaggi sotto forma di stringhe JSON inviate, a seconda dei casi, attraverso seriale, bluetooth o HTTP.

## Remote Hydrometer (ESP)

Questo sottosistema ha la funzione di monitorare il livello idrometrico di un punto specifico del fiume, interagisce con il Dam Service via HTTP ed è organizzato nei seguenti due task:

- Hydrometer Task: task di periodo 500 ms che si occupa di rilevare il livello dell'acqua, di impostare sulla base di questo valore lo stato del sistema e di comunicarlo al Dam Service.
- Led Alarm Task: task di periodo 100 ms che definisce il comportamento del led in base agli stati stabiliti dal task descritto precedentemente.

Le interfacce Light.h, Sonar.h e TemperatureSensor.h forniscono delle API di alto livello che permettono di gestire i sensori.

Il comportamento dei due task è descritto più dettagliatamente nei diagrammi presenti nel file 'SmartDamDiagram.pdf'.

## Dam Service (Backend - pc)

Rappresenta il sistema di controllo e interagisce via seriale con il Dam Controller e via HTTP con il Remote Hydrometer, con la Dam Dashboard e con la Dam App. Inoltre, il Dam Service si occupa di memorizzare le ultime 20 rilevazioni, lo stato del sistema, la percentuale di apertura della diga e la modalità (manuale o automatica) di gestione di apertura della diga stessa.

Per la comunicazione HTTP è stato utilizzato Vert.x Web, che permette di creare un web server che gestisce richieste POST tramite la definizione di un apposito handler.

È possibile inviare al Dam Service i seguenti comandi:

Request	Response
{"command": "add_detection", "water_level": [water level], "state": "[state]"}	200 OK
{"command": "get_last_detection"}	{"water_level": [water level], "state": "[state]", "time": [time]}
{"command": "get_mode"}	{"mode": "[mode]"}
{"command": "get_dam_opening"}	{"dam_opening": [dam opening]}
{"command": "get_all_detections"}	{0:[detection], 1:[detection], ...}

Per la comunicazione via seriale è possibile inviare i seguenti messaggi:

Request	Response
{"command": "set_mode", "mode": " [mode] "}	ok
{"command": "set_dam_opening", "dam_opening": [dam opening]}	ok
{"command": "get_last_detection"}	{"water_level": [water level], "state": " [state] ", "time": [time]}

## Dam Controller (Arduino)

Rappresenta il sistema di controllo della diga che interagisce via seriale con il Dam Service e via bluetooth con la Dam App ed è organizzato nei seguenti due task:

- DamTask: task di periodo 1 s che si occupa di comunicare via seriale con il Dam Service e via bluetooth con la Dam App. Inoltre, quando necessario, apre la diga (simulata dal servomotore) calcolandone la percentuale di apertura.
- LedTask: task di periodo 200 ms che definisce il comportamento del led in base agli stati del sistema.

Le interfacce Light.h e ServoMotor.h forniscono delle API di alto livello che permettono di gestire i sensori.

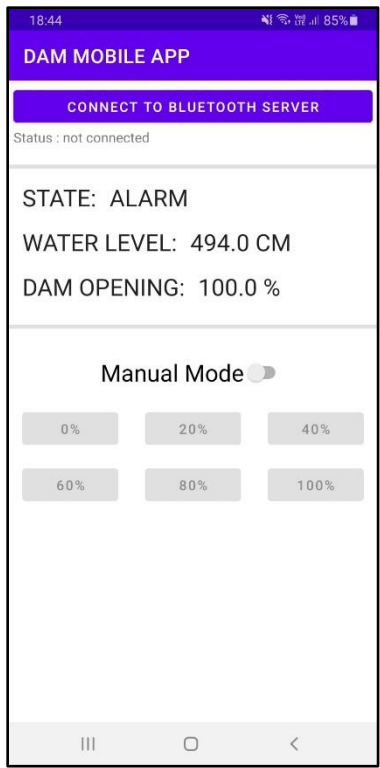
Il comportamento dei due task è descritto più dettagliatamente nei diagrammi presenti nel file 'SmartDamDiagram.pdf'.

## Dam App (Android - smartphone)

Mobile app che permette all'utente di controllare lo stato della diga e di intervenire in caso di allarme. L'applicazione comunica via bluetooth con il dam Controller e via HTTP, utilizzando la libreria Volley, con il Dam Service. I comandi che i sottosistemi possono scambiarsi sono quelli precedentemente descritti nella sezione Dam Service.

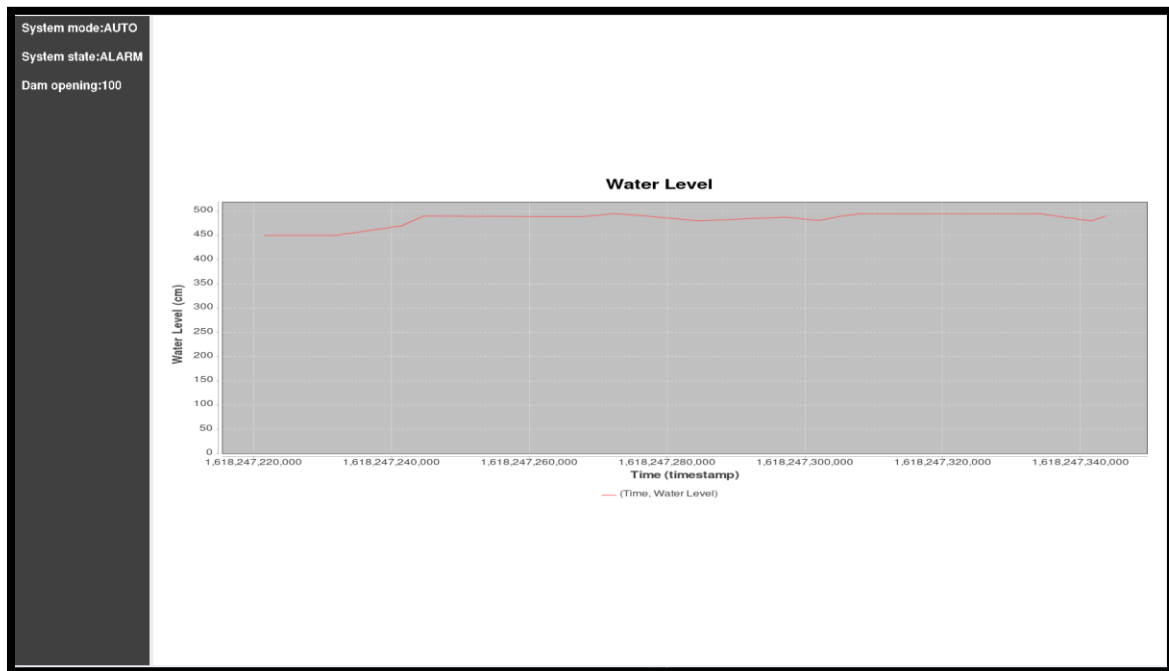
Nel caso in cui il sistema passa in stato di allarme, l'utente può scegliere di attivare la modalità manuale che permette di scegliere la percentuale di apertura della diga. Quando termina lo stato di allarme, il sistema torna alla modalità automatica.

Nella figura sotto riportata viene mostrata l'interfaccia della Dam App.



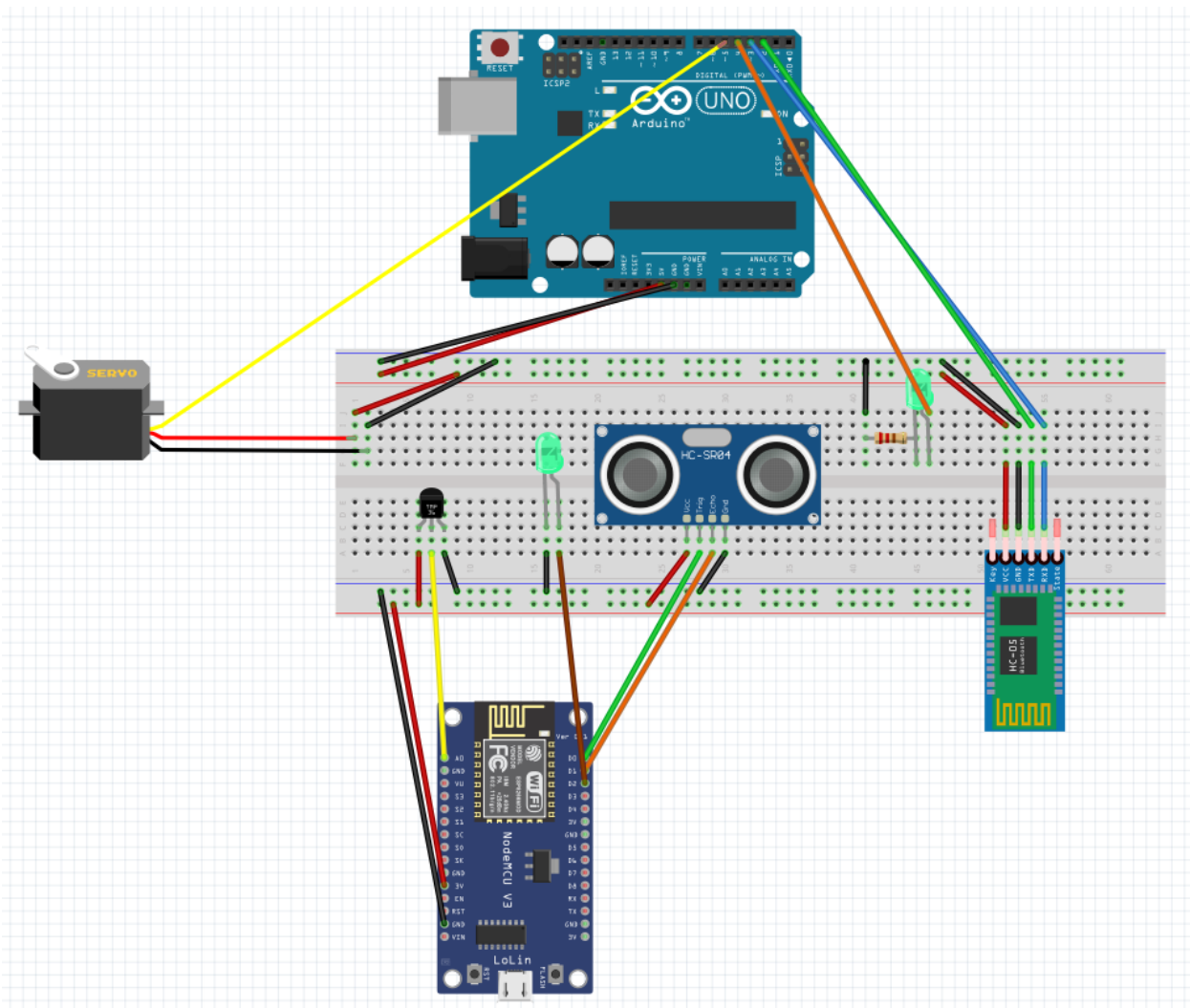
## Dam Dashboard (Frontend)

Rappresenta il frontend per la visualizzazione ed il monitoraggio dati.



Il sottosistema comunica via HTTP con il Dam Service, al quale richiede tutte le informazioni in formato JSON. In particolare, mostra la modalità di gestione della diga, lo stato del sistema e, in stato di allarme, la percentuale di apertura della diga. Inoltre, in stato di preallarme o allarme, mostra in un grafico le ultime 20 rilevazioni.

Di seguito lo schema Fritzing del circuito:



Progetto realizzato da:

Eddie Barzi

Sofia Belloni

Filippo Vissani