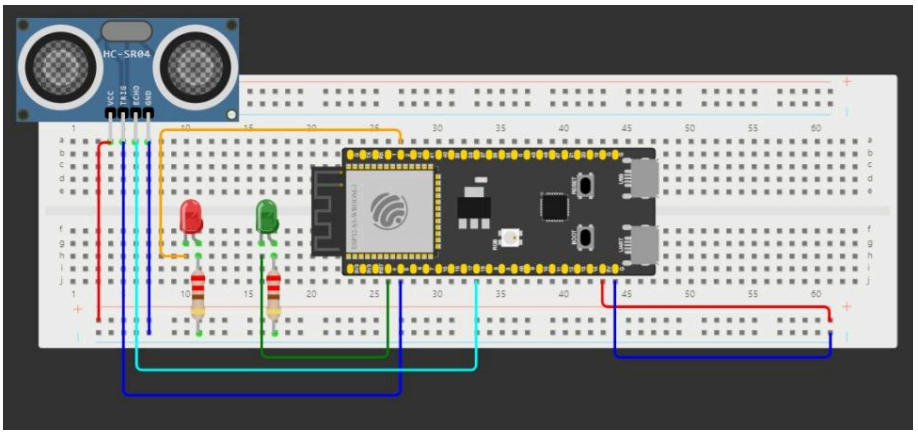
LINK VIDEO: https://www.youtube.com/watch?v=97EUmfyid2c
LINK WOKWI ESP32:
https://wokwi.com/projects/387815875102747649
LINK TINKERCAD ARDUINO:
https://www.tinkercad.com/things/4x54TmFUS5v-neat-kasi-jaiks/editel?sharecode=90uztEK7fy9jVdWofIqyn9OF4lvTqNn3guI0wWjpNKQ

Il codice è stato sviluppato in maniera modulare, definendo e implementando separatamente i 4 moduli fondamentali del sistema che interagiscono tra loro:

- **Water Level Monitoring subsystem**, si occupa di monitorare continuamente il livello dell'acqua del fiume. Necessita di essere connesso ad Internet in quanto comunica tramite MQTT con il River Monitoring Service subsystem. Per avere un riscontro diretto sulla connettività del modulo sono presenti due Led: uno rosso (acceso in caso di problemi di rete) e uno verde (acceso in caso di funzionamento corretto).

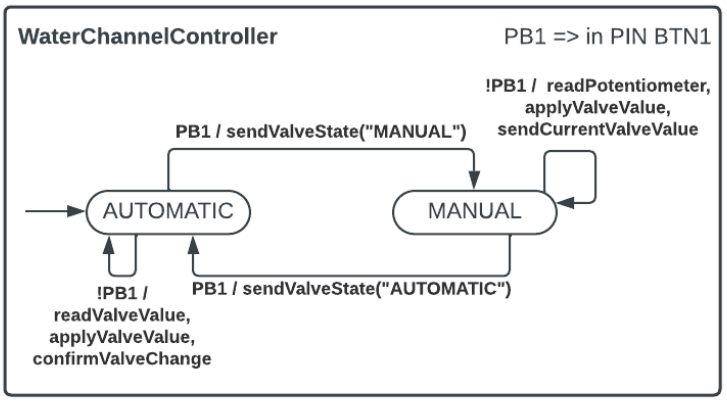


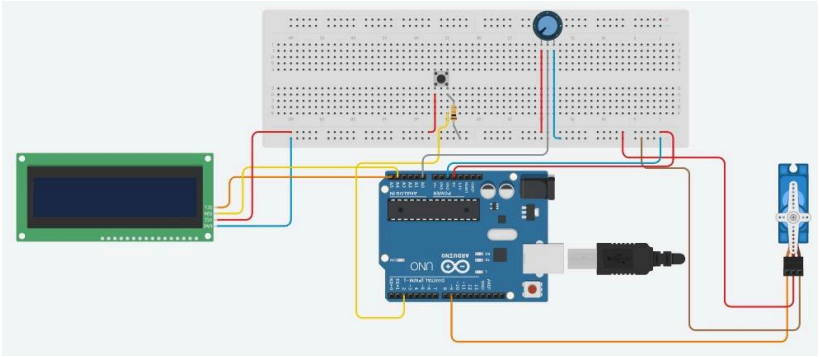
- **River Monitoring Service subsystem**, è il modulo principale che si occupa di gestire e controllare tutto il sistema, infatti comunica con il Water Level Monitoring subsystem per conoscere il livello dell'acqua corrente e stabilire come il sistema intero debba operare. Può quindi reagire a situazioni critiche modificando la frequenza di campionamento del livello dell'acqua e l'apertura della valvola di sfogo. Per poter modificare l'apertura della valvola è direttamente connesso al Water Channel Controller subsystem tramite collegamento Seriale. La modifica dell'apertura della valvola viene sempre confermata dal Water Channel Controller subsystem una volta che la applica. Inoltre, è stato implementato usando Flask come web framework e Python come linguaggio di programmazione. Questo modulo ha un proprio file JSON di configurazione per permettere di modificare i parametri relativi alle interconnessioni tra moduli senza dover modificare il codice (es. indirizzo broker MQTT, porta per la linea seriale, ecc…). Si occupa anche di interagire tramite HTTP con il River Monitoring Dashboard subsystem aggiornandolo sullo stato attuale del sistema quando questo cambia. Supporta la connessione simultanea di più istanze del River Monitoring Dashboard subsystem tramite il pattern Observer.

- **Water Channel Controller subsystem**, il modulo si occupa di gestire il funzionamento e la corretta apertura della valvola in base allo stato corrente del sistema. Comunica esclusivamente con il River Monitoring Service subsystem tramite PCDashboardComunicator, sfruttando il collegamento Seriale. Nello specifico mostra in un display LCD lo stato corrente del sistema e la percentuale di apertura

della valvola, dispone inoltre di un potenziometro (ManualModeController) e di un bottone (ManualModeActivator), quest'ultimo per alternare le due diverse modalità di funzionamento:

- Automatica: in questa modalità viene letto il livello corrente di apertura della valvola dal River Monitoring Service subsystem, viene aggiornata di conseguenza la valvola e il display LCD.
- Manuale: questa modalità prevede l'uso del ManualModeController per la lettura della percentuale di apertura della valvola. Come nella modalità precedente viene aggiornata la valvola ed il display LCD, inoltre viene comunicato al River Monitoring Service subsystem il nuovo stato ed apertura della valvola.





- **River Monitoring Dashboard subsystem**, questo modulo si occupa di mostrare graficamente lo stato dell'intero sistema, aggregando in un'unica interfaccia grafica le informazioni sul livello del fiume registrate dal Water Level Monitoring subsystem e quelle sullo stato della valvola, gestite invece dal Water Channel Controller subsystem. Il modulo visualizza il livello del fiume storicizzando i dati degli ultimi 2 minuti, inoltre mostra e permette di controllare l'apertura della valvola se il sistema è in modalità automatica. Comunica, tramite HTTP, con il River Monitoring Service subsystem, al quale segnala la propria presenza all'avvio e poi richiede le soglie del livello del fiume, corrispondenti ai vari stati del sistema. E' stato implementato in Python, utilizzando Flask come web framework e definendo gli opportuni endpoint per lo scambio dati.