

Smart Bridge

La nostra implementazione di *Smart Bridge* utilizza un architettura *task-based* insieme ad una *synchronous Final State Machine* per definire i comportamenti dei *task*.

Nella *FSM* abbiamo definito tre stati diversi, i quali si prestano ad identificare il livello di allarme corrente dello *Smart Bridge* :

- *Normal state*: corrisponde allo stato di allarme neutrale ed è attivo quando il livello dell'acqua si trova entro i parametri di norma. In questo stato, la *smart lighting* è operativa, LCD è inattivo e la valvola rimane chiusa.
- *Prealarm state*: indica un livello dell'acqua alto e vicino alla soglia di allarme. La *smart lighting* e la valvola rimangono come nello stato normale ma LCD si accende mostrando il livello dell'acqua corrente.
- *Alarm state*: il livello dell'acqua ha raggiunto un valore troppo alto raggiungendo la soglia di allarme. La *smart lighting* viene disattivata, la valvola inizia a aprirsi e diventa possibile prenderne il controllo, infine LCD mostra sia il livello attuale dell'acqua e l'apertura della valvola.

Di seguito, è mostrato un diagramma rappresentante i vari stati della *FSM* e le loro iterazioni:

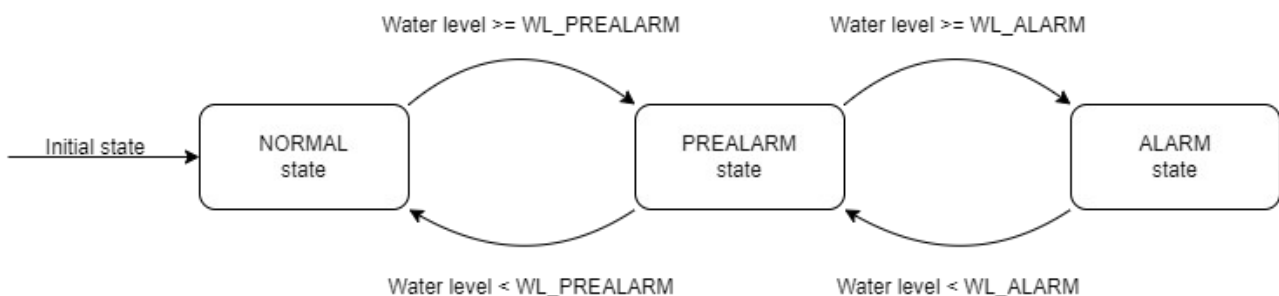


Figura 1: Diagramma riassuntivo del funzionamento della *FSM* implementata.

Per quanto riguarda i *task*, abbiamo deciso di svilupparli adottando il pattern MVC ovvero delineando in ciascuno di essi tre elementi fondamentali (Model, View e Control) permettendo così di poter seguire un'implementazione ordinata ed efficace.

Ognuno dei *task* si occupa di una determinata attività da eseguire periodicamente all'interno dello *scheduler*.

Questa è una lista dei *task* che sono stati predisposti per l'esecuzione dello *Smart Bridge*:

- *LedPowerTask*: gestisce l'accensione e spegnimento dei led in base allo stato di allarme corrente.
- *LightCheckTask*: aggiorna la misurazione dell'intensità della luce.
- *SonarCheckTask*: aggiorna la misurazione del livello dell'acqua.
- *PirCheckTask*: controlla se è stato rilevato del movimento dal sensore.
- *ServoControlTask*: controlla la valvola per il passaggio dell'acqua.
- *LCDTask*: aggiorna i messaggi visualizzati dal LCD.
- *SerialCommunicationTask*: gestisce l'interazione col programma *java*.

Infine, alleghiamo qua di seguito lo schema della *breadboard* per il funzionamento del sistema:

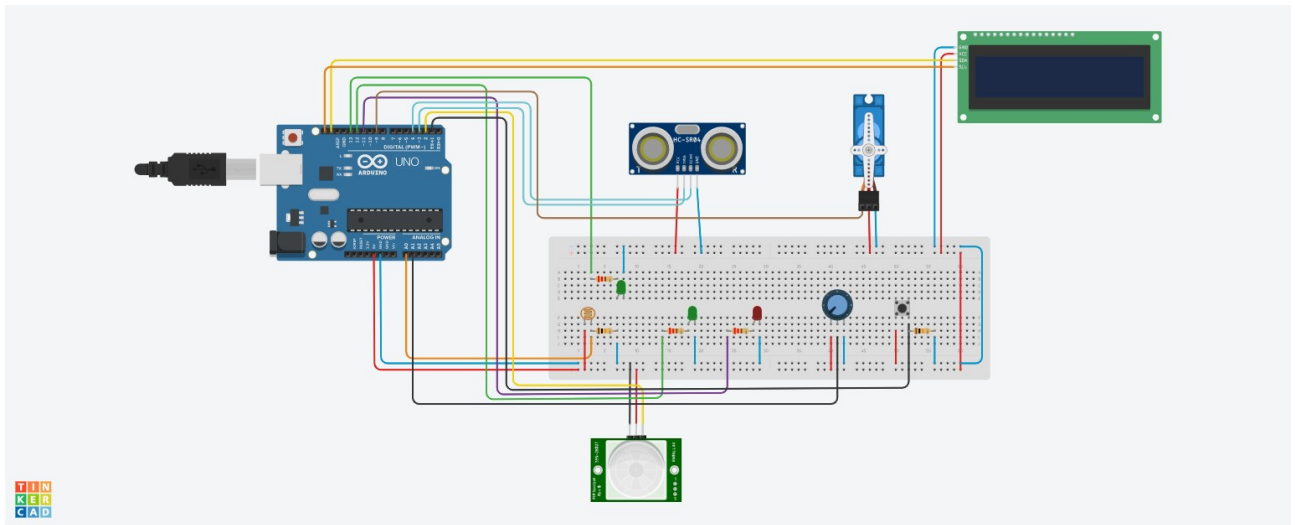


Figura 2: Schema della breadboard dello Smart Bridge.