

# SMART-BRIDGE-IOT

Marco Antolini, Daniel Capannini, Roberto Sopranzetti

Mail: marco.antolini6@studio.unibo.it , [daniel.capannini@studio.unibo.it](mailto:daniel.capannini@studio.unibo.it), [roberto.sopranzetti@studio.unibo.it](mailto:roberto.sopranzetti@studio.unibo.it)

## Progettazione hardware:

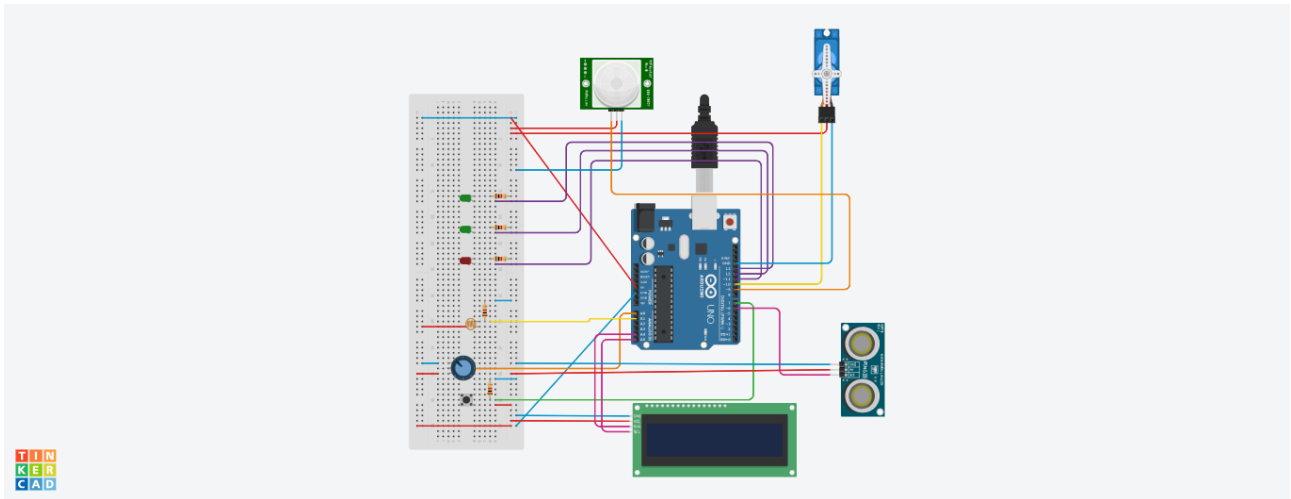


Figura 1.1: schema progetto hardware

## Progettazione software:

Essendo il LightSystem (parte che si occupa della gestione delle luci) e il WaterSystem (parte che si occupa delle gestione del livello dell'acqua) separate e quasi senza punti comuni abbiamo deciso di inserire uno scheduler che si occupa di scegliere quale dei due sistemi deve essere eseguito.

Scheduler

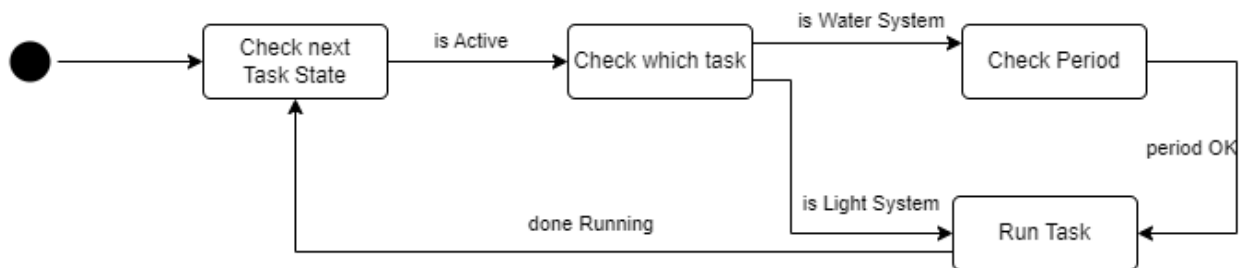


Figura 1.2: schema funzionamento scheduler

Come si vede in figura 1.2 lo scheduler controlla se il prossimo task è pronto per l'esecuzione e successivamente di quale task si tratta, nel caso del WaterSystem controlla inizialmente il periodo del sistema e una volta stabilito che è tutto ok esegue il task, mentre per il LightSystem non ha bisogno di

ulteriori controlli in quanto non è né periodico né limitato. Finita l'esecuzione del task lo scheduler ritorna a controllare lo stato del task successivo.

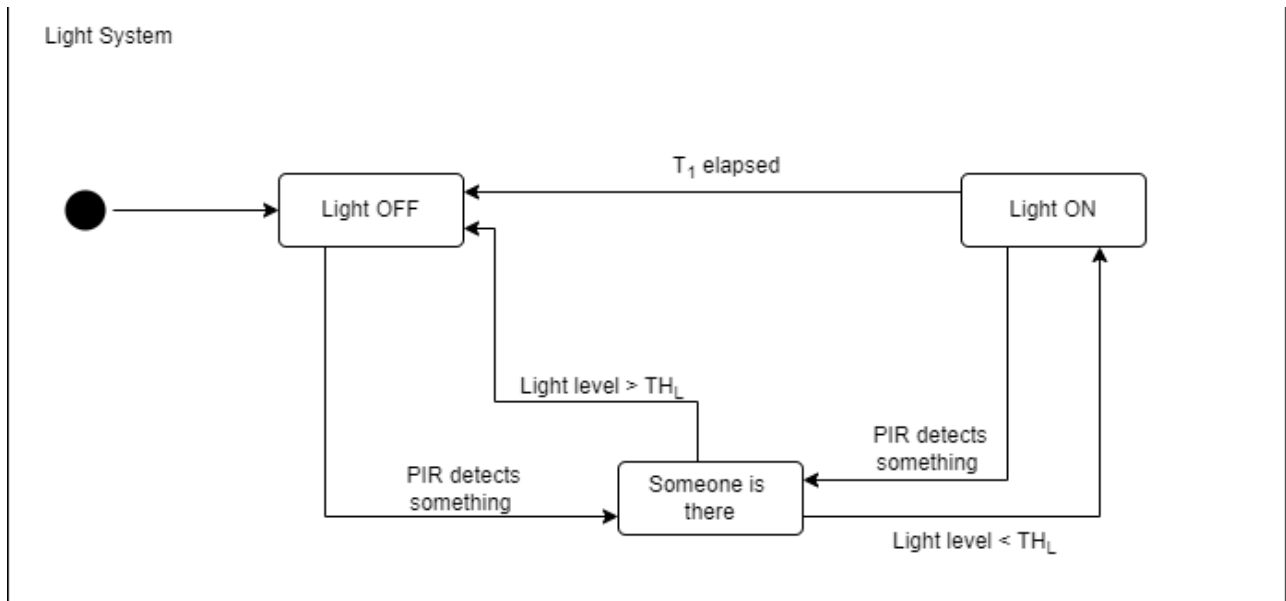


Figura 1.3: Schema funzionamento LightSystem

Il LightSystem che gestisce il funzionamento delle Smart Lights come spiegato sopra, parte dallo stato iniziale OFF e così rimane fino a quando non viene rilevato del movimento dal PIR, che significa che c'è qualcosa sul ponte. A questo punto se il livello di luce rilevato dal Light Sensor non supera quello del threshold allora la luce si accende, altrimenti rimane spenta. La luce rimarrà accesa fintanto che il livello di luce rilevata rimane sotto la soglia ed il PIR continua a rilevare del movimento, in caso contrario una volta trascorso il tempo  $T_1$  la luce si spegne e il sistema torna allo stato iniziale.

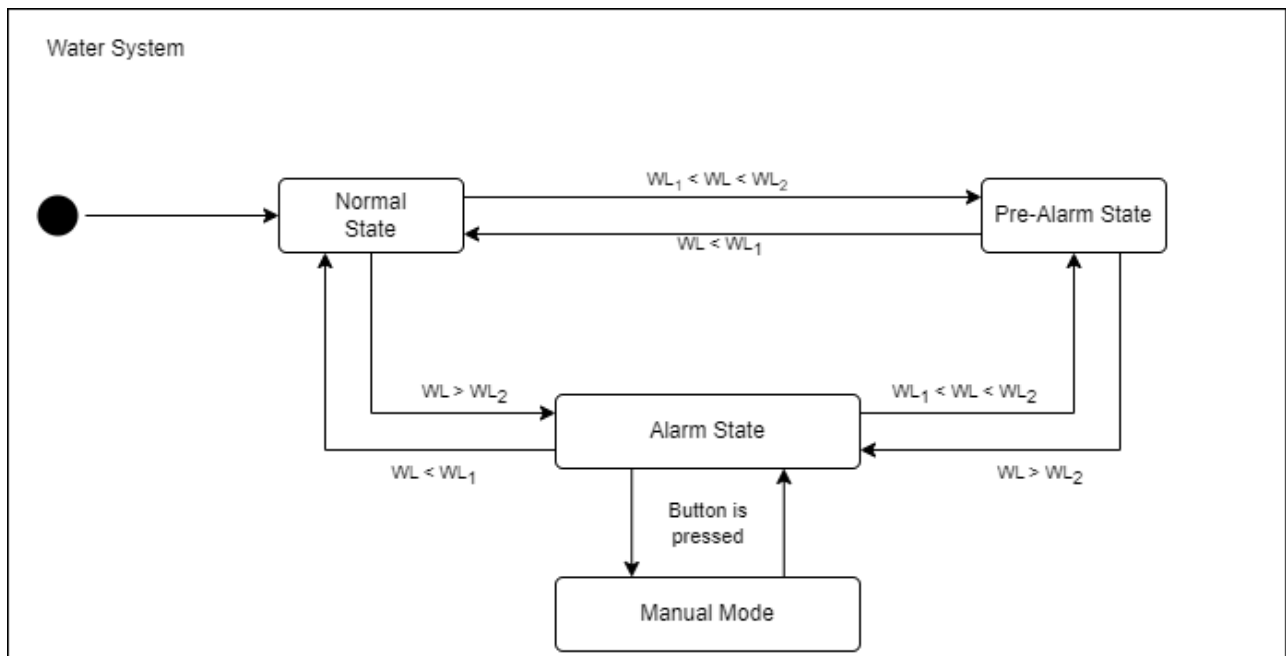


Figura 1.4: Schema funzionamento WaterSystem

Nel WaterSystem viene gestito lo stato del ponte in base al livello dell'acqua rilevato dal Sonar. Si parte dallo stato normale dove il Sonar esegue una rilevazione ogni 3 secondi. Se il WL (Water Level) supera il primo stadio, ma non il secondo il sistema entra in pre-allarme e la rilevazione viene eseguita ogni 2

secondi anziché 3. Non abbiamo inserito un controller per far lampeggiare il LED rosso in quanto il task viene già eseguito ogni 2 secondi essendo periodico, per cui ad ogni esecuzione del task viene switchato lo stato del LED. Se il WL supera anche il secondo stadio, allora il sistema entra in stato di allarme. Innanzitutto il LightSystem viene disattivato, poi per il WaterSystem il periodo del task viene azzerato e la rilevazione viene effettuata ogni secondo. A seconda del livello dell'acqua viene aperta di un certo valore la valvola ed è possibile in questo stato premere il Bottone per entrare in modalità manuale, dove l'apertura della valvola viene regolata dal potenziometro, mentre se viene premuto una seconda volta il bottone il sistema ritorna in modalità automatica. Se il sistema non è in stato normale sullo schermo LCD viene stampato lo stato attuale del sistema, insieme al WL e l'ampiezza dell'apertura della valvola. A seconda del WL è possibile passare da uno stato all'altro senza passare per stati intermedi.

#### Python:

in python viene stampato un grafico con i valori livello acqua/tempo e stampati se il sistema di illuminazione è acceso e il che stato si trova il sistema dell'acqua (allarme, pre-allarme, normale), i dati vengono letti dalla seriale, grazie al fatto che il programma su Arduino stampa una stringa contenente tutti questi valori che poi vengono decodificati e separati dal programma python.