Smart Bridge

Sistema di automatizzazione della gestione di un ponte di un fiume

Muccioli Federico
federico.muccioli5@studio.unibo.it
0000971342

Design SYNCHRONOUS FSM

Task

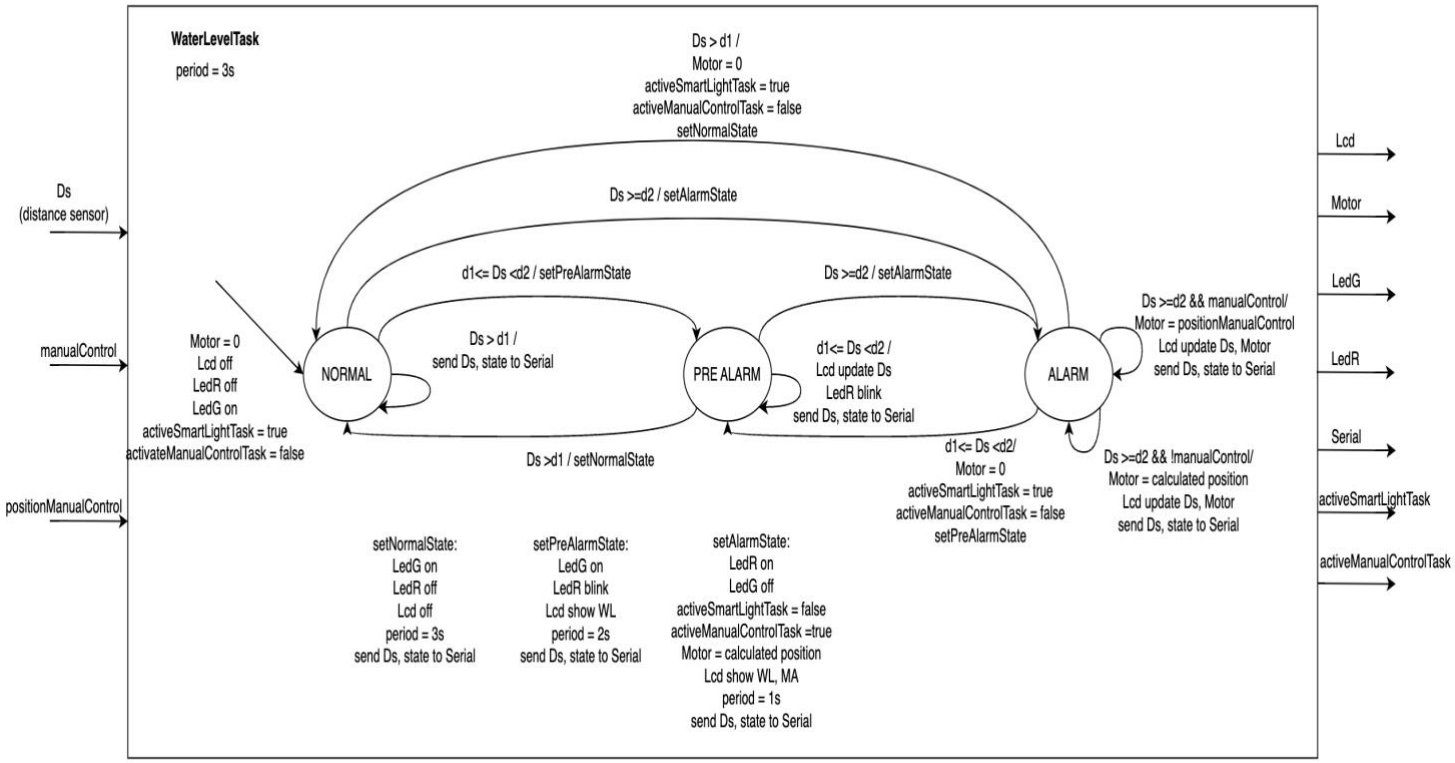
Il sistema consiste nell'automatizzazione della gestione di un ponte sul fiume.
Il sistema presenta principalmente due sottosistemi: lo **smart lighting**, che consiste nell'illuminazione del ponte al passare delle persone e il monitoraggio del **water level**, che rileva il livello del fiume e regola una valvola di flusso.
Per implementare quindi la macchina a stati finiti all'interno di Arduino è stato scelto di implementare due principali task: **smartLightingTask** e **waterLevelTask**.
Per il controllo della valvola di flusso è data la possibilità di interagire manualmente sul posto, tramite un bottone e un potenziometro o dalla centrale, tramite software. In questo caso è stato scelto di implementare un ulteriore task, **manualControlTask** che verifica e gestisce le richieste di **manual control**.
Il task relativo al monitoraggio dell'acqua è un task privilegiato in quanto in base allo stato assunto abilita o disabilita i restanti task. Esempio, il livello dell'acqua supera una certa soglia, il sistema entra in modalità d'allarme, viene disabilitato il sistema di illuminazione intelligente e viene attivato il sistema di controllo manuale, quindi verrà disabilitato smartLightingTask e abilitato manualControlTask. I due task sono dati come parametro alla classe del costruttore del waterLevelTask mascherati dall'interfaccia Active che permette l'abilitazione/disabilitazione ma non la visione degli altri metodi come tick.

Scheduler e periodicizzazione

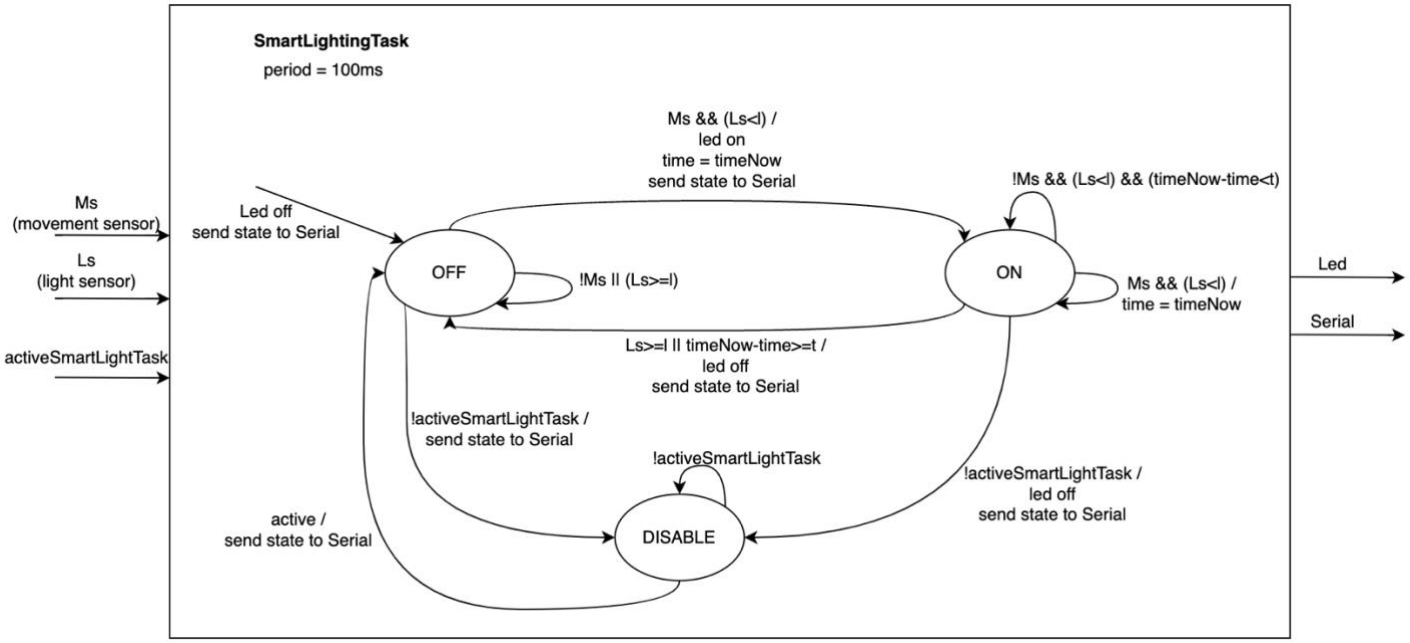
Per la gestione dei task è stato implementato uno scheduler che alterna un task all'altro in base al periodo e allo stato (attivo o disattivo).
Per il campionamento e la gestione del livello dell'acqua è stato scelto un periodo di 3 secondi in quanto il livello non varia bruscamente ma vogliamo comunque avere una misura aggiornata.
Superato un certo livello, in questo caso di 70cm, il periodo di campionamento diminuisce a 2 secondi in quanto siamo in una condizione di pre-allarme e vogliamo un sistema più reattivo.
Superati gli 80 cm entriamo in una condizione di allarme, il periodo varia a 1 secondo ed entra in azione la valvola.1
Per il sistema di illuminazione intelligente il periodo è invece di 100 millisecondi, in quanto vogliamo che l'illuminazione sia semi-istantanea alla rilevazione del movimento.
Infine, per il controllo manuale, il campionamento è di 200 millisecondi, valore che ci permette di cogliere qualsiasi evento e contrastare il bouncing del bottone che sfasa le misure.
Appuntati i valori dei periodi dei task possiamo scegliere il periodo dello scheduler ovvero 100 millisecondi (minimo comune multiplo) che permette l'esecuzione di ciascun task e di non far lavorare inutilmente il microcontrollore.

Diagrammi degli stati finiti

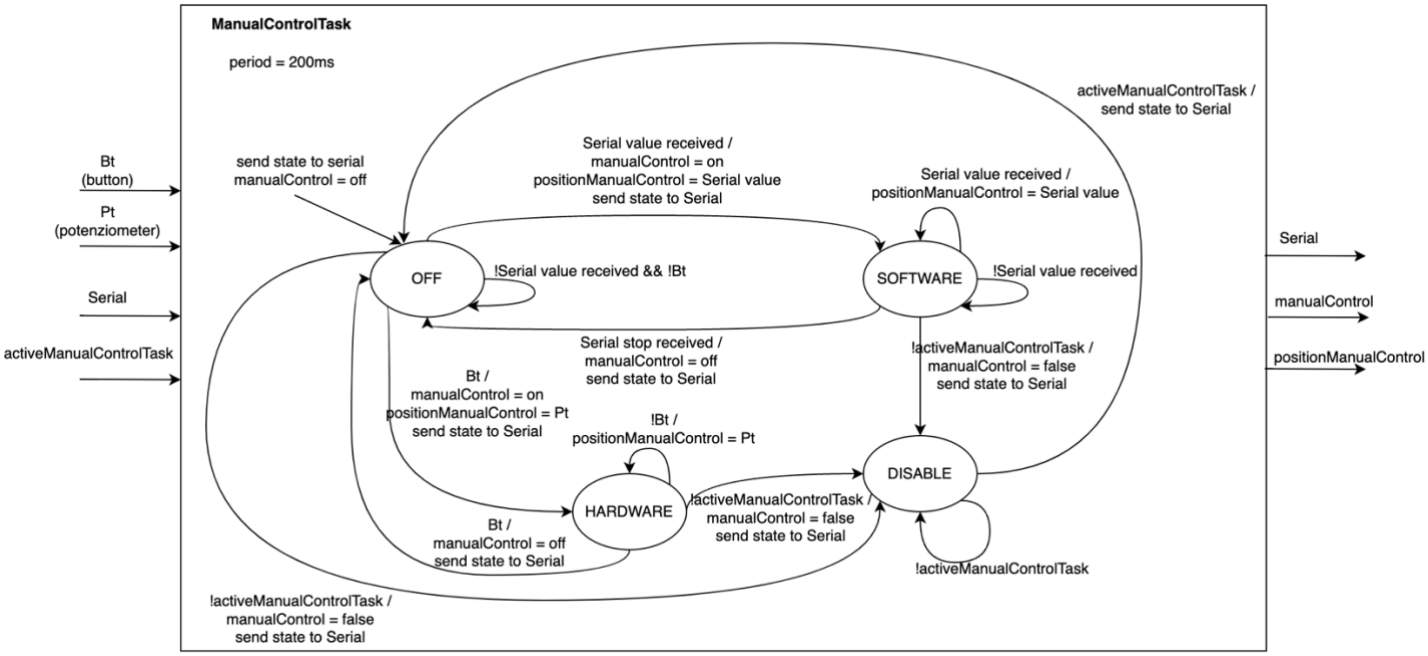
WaterLevelTask



SmartLightingTask



ManualControlTask



Scelte progettuali

Power saving

Proprio per non sforzare troppo l'Arduino e ricevere un guadagno nei consumi del microcontrollore è stato scelto di implementare la funzione di sleep. Infatti, il periodo di 100ms dello scheduler, è un lasso di tempo abbastanza ampio per addormentare Arduino, risvegliarlo in tempo e risparmiare energia. Per la funzione di sleep è stato scelto di implementare la funzione IDLE che permette comunque l'esecuzione dei timer necessari al funzionamento delle periferiche.

Comunicazione seriale

Il microcontrollore comunica costantemente attraverso la seriale mandando dati relativi allo stato dei sistemi, al campionamento del livello dell'acqua e ricevendo comandi del controllo manuale via software. È stato quindi necessario scegliere un protocollo di comunicazione. La comunicazione avviene attraverso stringhe con a capo un carattere di valore univoco che rappresenta il tipo di messaggio: un messaggio inoltrato da Arduino può essere "v78" dove 'v' indica che il messaggio è relativo al livello dell'acqua e a seguito abbiamo il valore del livello d'acqua. Il compito di tradurre i messaggi nel software Java del computer è stato dato all'interfaccia **SmartBridgeMessageInterface** che, attraverso la relativa implementazione, definisce metodi utili per ottenere informazioni decifrate. Per quanto riguarda Arduino non è stato necessario implementare un'interfaccia di decodifica in quanto i messaggi in input sono di un solo tipo e relativi al controllo manuale. In futuro potrebbe comunque essere comodo implementare una classe definita per questo scopo.
Per quanto riguarda invece la comunicazione seriale vera e propria sia a livello di microcontrollore che a livello dell'applicazione del computer sono state utilizzate le classi fornite a lezione, MsgService per Arduino e SerialCommChannel che utilizza la libreria jssc per java a lato pc.

Altre scelte progettuali

- Come clock per lo scheduler è stato utilizzato il timer1 con supporto della libreria TimerOne.
- Come timer per gestire il servo motore è stato invece utilizzato il timer2 attraverso la libreria ServoTimer2. Finché c'è un motore attaccato (tramite attach) la libreria fa un uso esclusivo del timer 2 che viene anche utilizzato per la pwm dei pin 3 e 11, per questo il motore è stato collegato al pin 11.
- La rilevazione della pressione del bottone è stata eseguita attraverso polling eliminando il fattore del bouncing settando un periodo di campionamento di 200ms ovvero lo stesso periodo del task che lo gestisce.

- Per Lcd è stato utilizzato il protocollo I2C tramite utilizzo della libreria LiquidCrystal_I2C che limita l'utilizzo dei pin.
- Il blink del led rosso è stato implementato all'interno del task che gestisce il livello dell'acqua, in quanto task di livello secondario. Poteva comunque essere definito come task principale ed essere abilitato e disabilitato dal waterLevelTask, come per gli altri task.
- Per la visione dell'andamento del livello dell'acqua a lato computer è stata sfruttata la libreria jfreechart che fornisce un ChartPanel che estende JPanel di Swing. Questo pannello utilizza una serie alla quale aggiungendo dati aggiorna il grafico in tempo reale.
- Al sonar non è stato applicato un tempo massimo in quanto è supposto che anche se il fiume in secca il segnale rimbalza sul fondo. È stato comunque utilizzato un settaggio che impone il livello minimo del fiume a 0 non ottenendo così valori negativi anche alla rilevazione di una distanza maggiore.

Schema elettrico