ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Project Report

Secondo elaborato per il corso di Sistemi embedded e Internet-Of-Things

Bevilacqua Anny

Rannick Nguemo Olivia

Sommario

Il progetto è stato realizzato come elaborato per il corso di Sistemi embedded e Internet-Of-Things, in riferimento all'anno accademico 2022-2023.

Il corso di Sistemi embedded e Internet-Of-Things mira a dare agli studenti una conoscenza di base per lo sviluppo e la realizzazione di tecnologie basate su microcontrollori e sistemi operativi embedded/realtime. In tale ambito si pone il presente progetto, che richiede di realizzare il sistema di automatizzazione di un ponte, sfruttando una gestione real time degli eventi e utilizzando microcontrollori gestiti tramite l'utilizzo di una scheda arduino. Tale sistema sarà in grado di gestire il flusso d'acqua tramite un'apposita valvola per ridurre il rischio di esondazione e rilevare eventuali movimenti avvenuti sul ponte, riconducendoli al passaggio di persone e accendendo di conseguenza il sistema di illuminazione per facilitarne la traversata.

L'elaborato si compone quindi di diverse parti. A lato software sono presenti un sistema per la gestione dell'illuminazione del ponte, uno per la gestione del sistema idrico comprendente valvola e monitoraggio dell'acqua e un'interfaccia grafica per visualizzare lo stato del sistema. Lato hardware è invece presente un circuito che permette di gestire sensori.

Componente hardware

In questo progetto, abbiamo usato alcuni sensori, e ovviamente il BREADBOARD e ARDUINO sono gli elementi centrali del circuito realizzato. Come sensori, abbiamo usato il PIR per riconoscere la presenza di qualcuno sul ponte. Questo è segnalato da una LED VERDE accesa(ledA). Il SONAR per misurare la distanza tra l'acqua e il ponte che è il componente principale per l'implementazione degli stati del ponte (ALARM, PRE-ALARM e NORMALE) segnalati rispettivamente da una blinking-led rossa(ledC) ed una led verde(ledB) accesa. Il SERVO MOTOR che rappresenta il nostro sistema di valvola in situazione di allarme. E poi il LIGHT SENSOR per avere il livello di luminosità sul ponte. In caso di allarme, per permettere l'apertura manuale delle valvole, abbiamo usato un PULSANTE prendere il controllo manualmente e il POTENTIOMETER per fare girare il servo motore. LCD usato per informare il livello dell'acqua e il degree_opening_valve in caso di allarme.

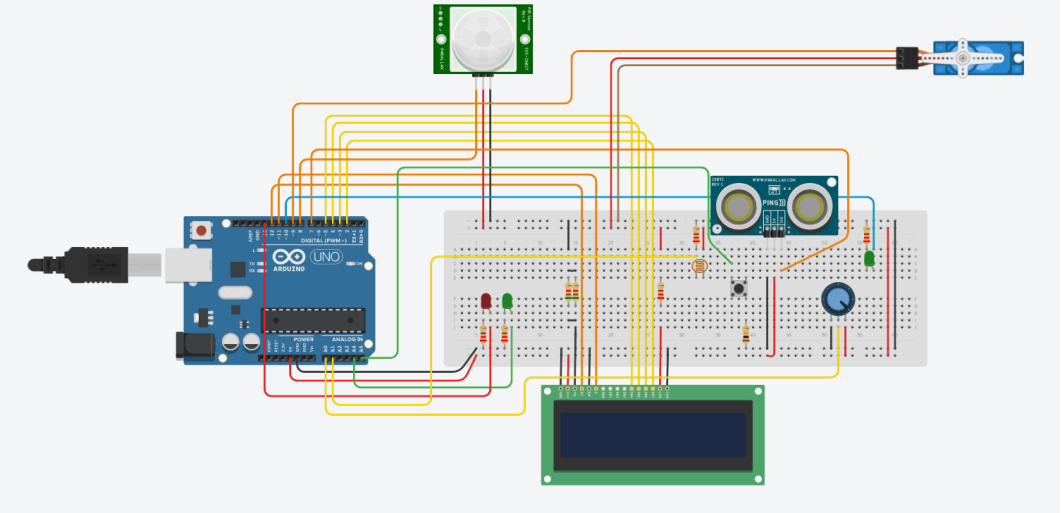


Fig. 1 - Schema del circuito nella versione con display LCD 1602 IIC

Componente software

Sistema di idrico

Il monitoraggio del livello dell'acqua e la conseguente gestione dello stato del sistema compongono la parte più articolata del programma. Il file di riferimento per questa parte del programma è WaterMonitor.h, assieme alla propria implementazione, che contiene le funzioni per la gestione dei cambiamenti di stato del sistema, nonché riferimenti alle varie sottoparti del sistema idrico.

In tale file sono infatti definiti, oltre a delle variabili per tenere traccia dello stato corrente e del livello dell'acqua, riferimenti ai componenti hardware del sistema, come i led per la segnalazione dello stato o il display LCD per visualizzare distanza dalla superficie dell'acqua e apertura della valvola di drenaggio.

Il calcolo della distanza della superficie dell'acqua, assieme alla funzione di blinking del led rosso durante lo stato di pre-alarm, sono gestite attraverso delle task ripetute periodicamente e che possono essere abilitate o meno a seconda delle necessità.

L'implementazione della valvola per il deflusso dell'acqua fa invece riferimento ai file BridgeValve.h e BridgeValve.cpp. In realtà, anche questi due file possiedono funzionalità limitate, delegando il grosso del lavoro a task e sottoclassi dedicate.

La regolazione automatica della valvola avviene mediante una task che, preso in input il livello attuale dell'acqua, esegue una proporzione tra la distanza dalla superficie e i gradi di apertura da comunicare al servomotore. La regolazione manuale, attivata successivamente alla pressione del pulsante, viene invece gestita tramite le task contenute in UserInputHandler.h.

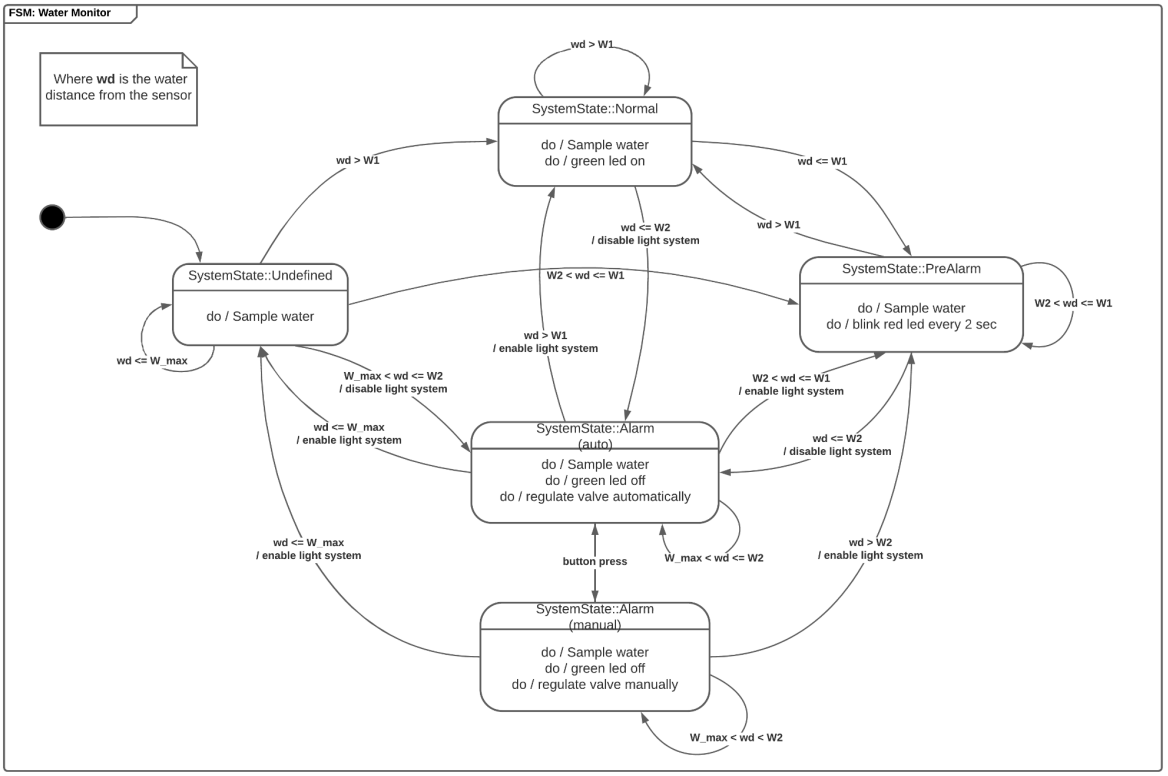


Fig. 2 - Diagramma degli stati per il sistema di monitoraggio dell'acqua

Sistema di illuminazione

Il sistema di illuminazione del ponte viene definito nel file Illumination.h e nel rispettivo file di implementazione. La classe è di per sé molto semplice, contenente i riferimenti alle luci e ai sensori interessati. Il controllo sul livello di illuminazione e la rilevazione dei movimenti vengono eseguiti all'interno di una task apposita, che definisce quindi se sia il caso o meno di accendere le luci sul ponte.

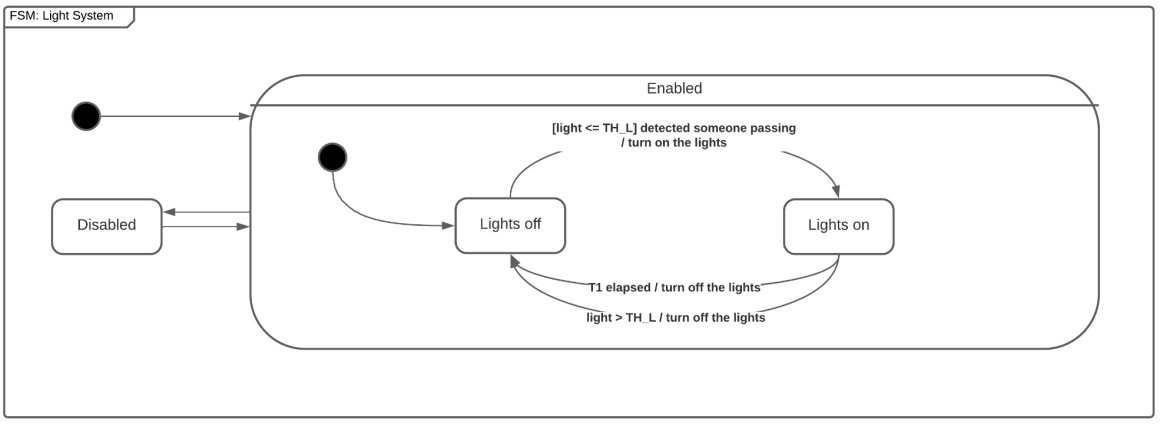


Fig. 3 - Diagramma degli stati per il sistema di illuminazione del ponte

Componente di interfaccia grafica

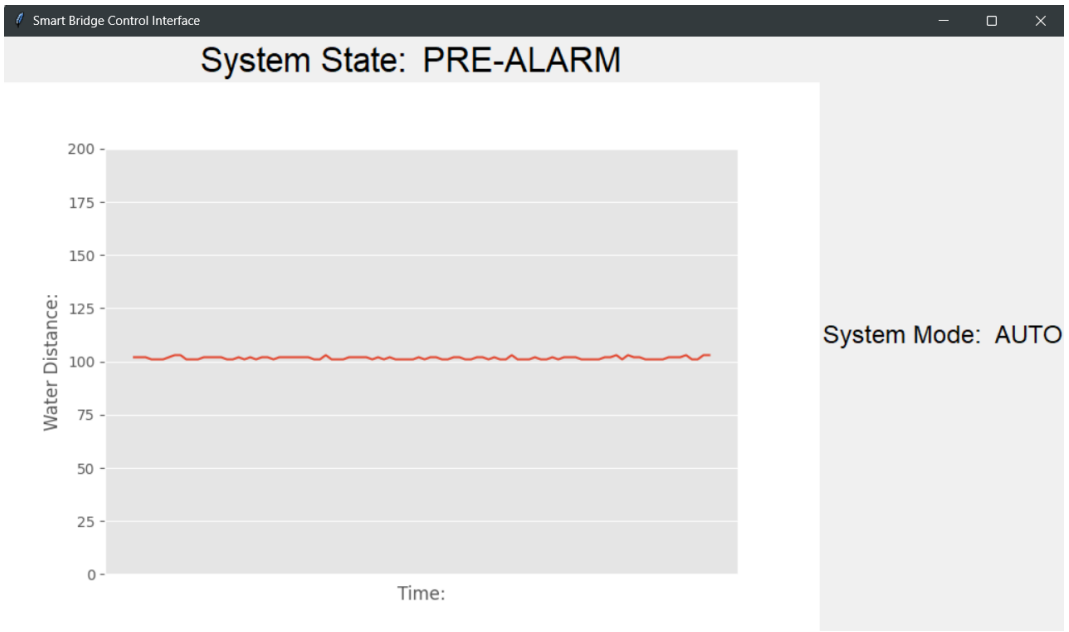
L'interfaccia grafica è molto semplice, riportante lo stato del sistema e un comodo grafico con l'andamento del livello dell'acqua nell'ultimo periodo. L'applicativo è stato realizzato in linguaggio python ed è suddiviso principalmente in due file principali, più uno di avvio.

Il file ArduinoBoard.py, contenente l'omonima classe, fornisce delle semplici funzioni per gestire il passaggio dei messaggi dalla scheda all'applicativo python.

La ricezione dei messaggi inviati da Arduino avviene in un thread separato, per assicurare un ascolto continuo, tramite il collegamento del programma al monitor seriale della scheda. Una volta letti, i messaggi vengono decodificati e interpretati, per poi essere gestiti a seconda delle informazioni contenute. In linea generale si è deciso di salvare il contenuto dei messaggi ricevuti su dei file di log, per facilitare un'ipotetica esportazione dei dati ad altri applicativi e tenere una traccia dello stato corrente del sistema in caso di un'interruzione improvvisa del servizio.

Il file SmartBridgeGui.py contiene invece il codice necessario alla realizzazione vera e propria dell'interfaccia grafica, dalla creazione della finestra, al plotting del grafico con l'andamento della distanza dalla superficie dell'acqua. Nonostante il sistema tenga traccia di tutte le misurazioni dall'avvio del sistema, per facilitare la visualizzazione dei dati raccolti, il grafico è stato impostato per mostrare solo le ultime 100 misurazioni della distanza dalla superficie dell'acqua.

Fig. 4 - Interfaccia grafica dell'applicazione



Conclusione e pensieri finali

La possibilità di utilizzare sensori per ottenere maggiori informazioni dal mondo esterno è sicuramente modo curioso per interagire con il programma e dare un senso a quelli che altrimenti sarebbero solo numeri sullo schermo. Durante la carriera universitaria spesso capita di concentrarsi unicamente sul lato software dei programmi, talvolta dimenticando ciò che si potrebbe realizzare con solo qualche semplice componente elettronico.

Nonostante alcune difficoltà tecniche incontrate durante lo sviluppo del progetto, possiamo affermare di aver raggiunto i requisiti fondamentali dell'elaborato, pur non essendo riuscite ad sviluppare il controllo della valvola da interfaccia grafica. Implementare uno SmartBridge in arduino è stato molto interessante soprattutto nella gestione delle diverse task nel progetto. Siamo certe che questa esperienza ci abbia mostrato come poterci interfacciare con un sistema real-time basato su dati in continuo cambiamento e possibili criticità nell'esecuzione.

Risorse:

Link al video:

https://www.canva.com/design/DAFUexuuCsw/YBSTXdUQsVkZLlF1waLG7A/watch?utm_content=DAFUexuuCsw&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton