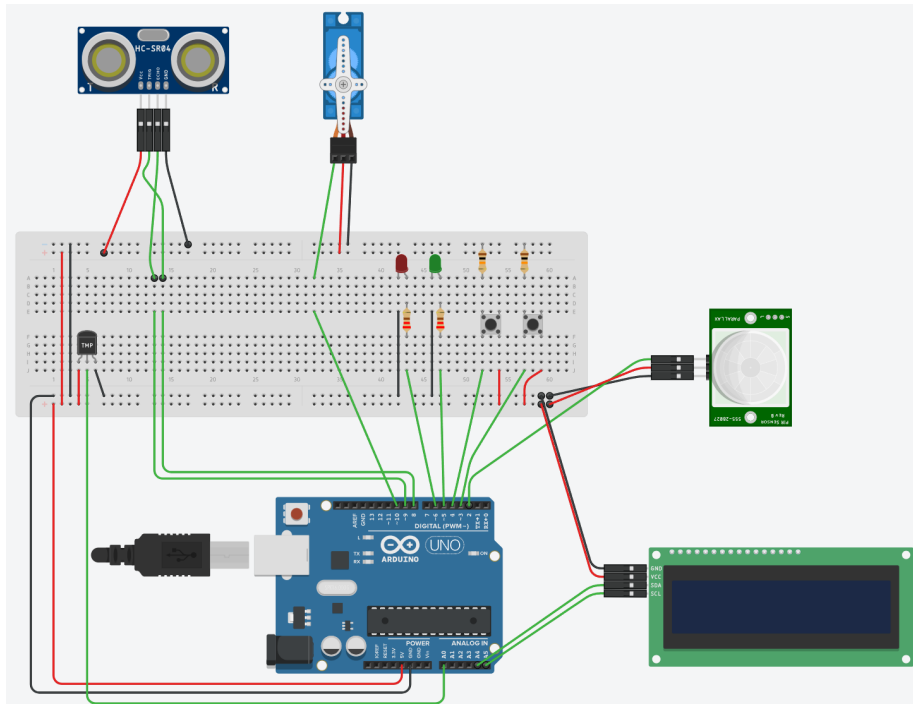


Assignment-02 Sistemi embedded & IoT

Luca Venturi

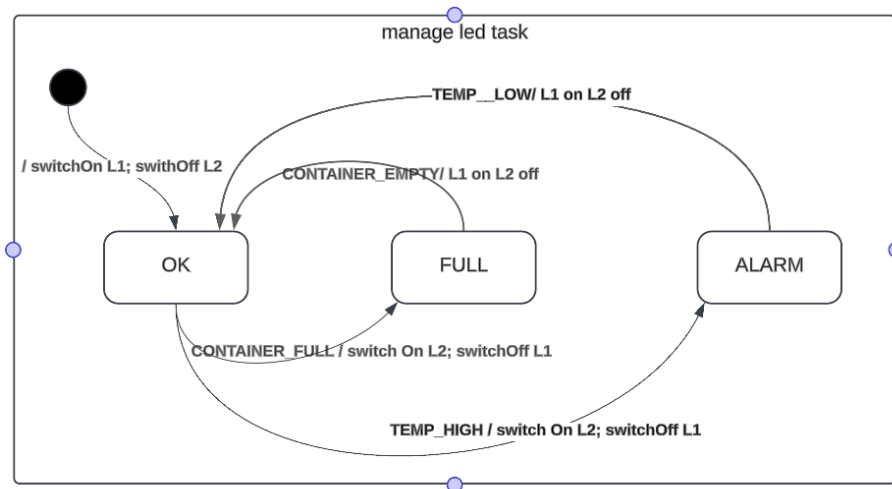
December 2024

1 Representation of the circuit using Tinkercad

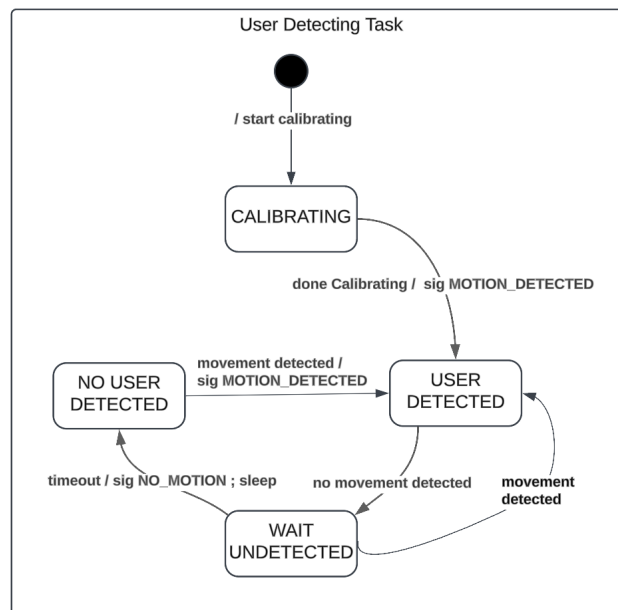


2 Task

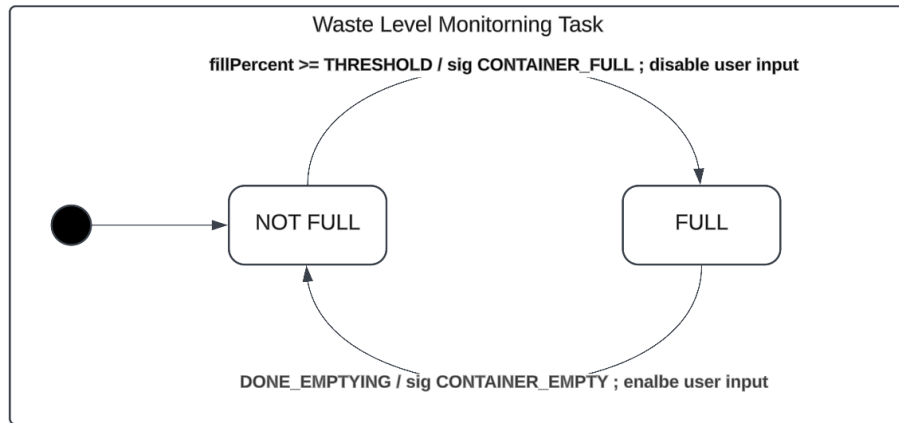
2.1 Controllo led - periodo 200ms



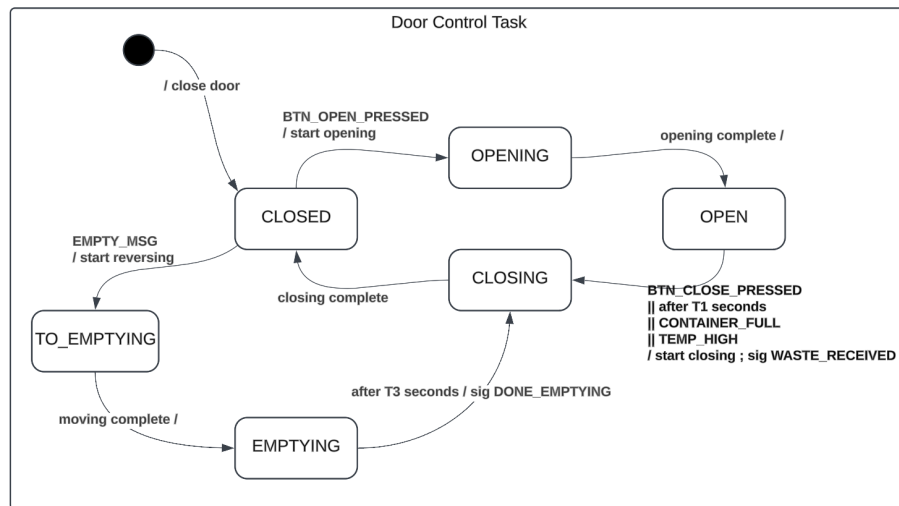
2.2 Rilevamento utente - periodo 200ms



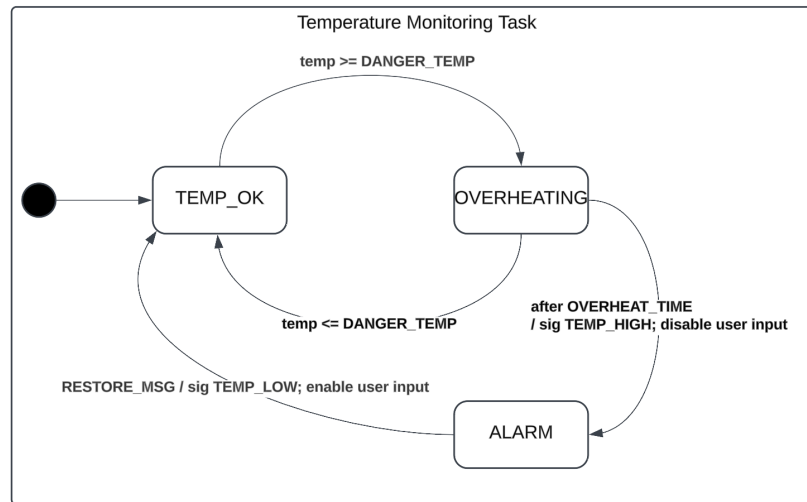
2.3 Monitoraggio livello Waste - periodo 200ms



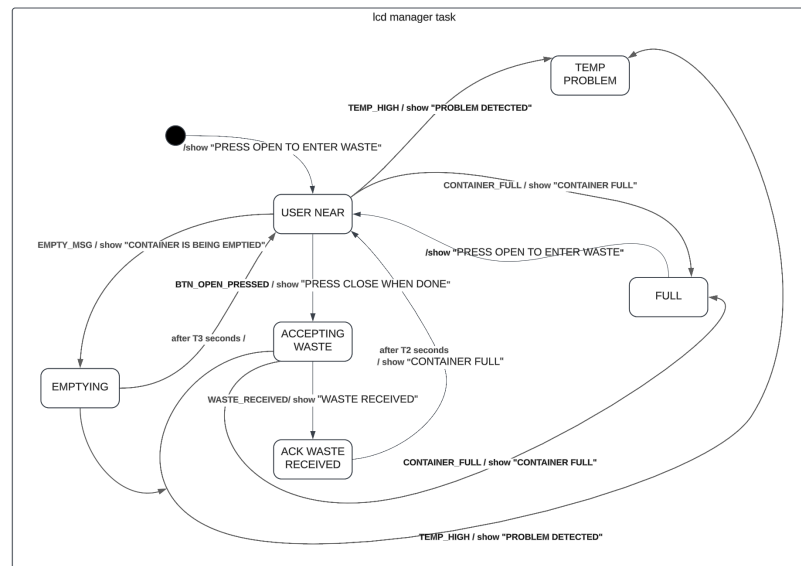
2.4 Controllo porta - periodo 200ms



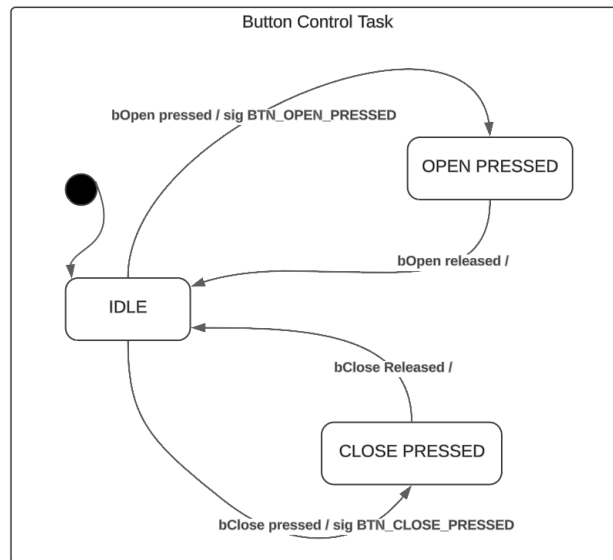
2.5 Monitoraggio temperatura - periodo 200ms



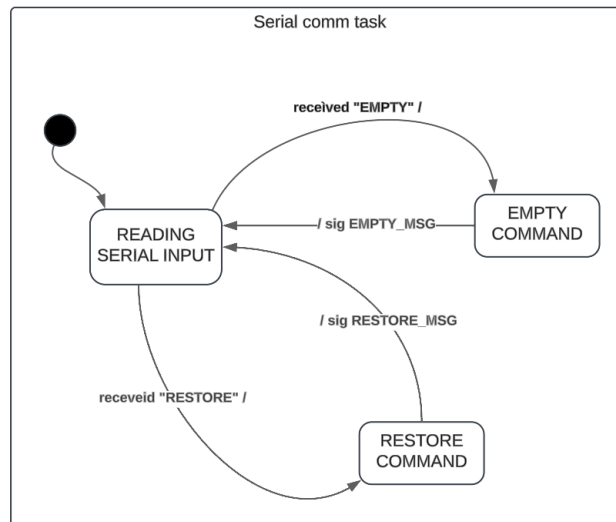
2.6 Output su lcd - periodo 200ms



2.7 Controllo sui bottoni - periodo 100ms



2.8 Comunicazione con la dashboard operatore - periodo 200ms



3 Osservazioni

3.1 Comunicazione tra task

Shared o Observer? Per far comunicare le task ero in dubbio se farle comunicare tramite delle variabili o flag globali e shared, visto che non ci sono problemi di concorrenza essendo il mio uno scheduler non preemptive. Ho deciso però di farle comunicare con il pattern Subject-Observer in cui il subject notifica tutti gli observer collegati dell'avvenimento di un evento. Per la lettura degli eventi non ho usato una coda ma una semplice variabile con un flag che indica se c'è un nuovo evento. La variabile potrebbe essere sovrascritta prima che la task possa leggerla e perdere così l'evento. Grazie all'ordinamento delle task nello scheduling questo non sembra succedere ma se si volesse estendere l'applicazione sarebbe necessario implementare una coda di eventi.

3.2 Periodo di scheduling

Quanti millisecondi? Ho scelto un periodo di scheduling di 100ms ma l'unica task che ha bisogno di tale velocità è quella per la lettura dell'input dell'utente sui bottoni. Con 100ms riesco a catturare quasi tutti gli eventi di click e mi permette comunque di avere abbastanza tempo per eseguire tutte le altre task con un margine piuttosto grande. C'erano problemi con la rotazione del servo in quanto impiega circa 2ms per ogni grado di rotazione, ma la separazione in più stati (invece di uno stato OPEN due stati OPENING e OPEN) mi permette di usare pochissimo tempo cpu.

4 Video del funzionamento

Link al video condiviso su onedrive: [clicca qui](#) per il video Il video è presente anche su github

O questo shortener: <https://shorturl.at/FIf5s>

5 Link alla repo su github

<https://github.com/LucaVenturi/EsIot-Second-Assignment>