

# RELAZIONE TERZO ASSEGNAIMENTO

## SMART GREEN HOUSE

A cura di Cristiano Aprigliano, Mariano Caldara, Scucchia Matteo

### INDICE

Funzionamento generale	pagina 1
Scelte implementative	pagina 1
Controller	pagina 1
Server	pagina 2
Client	pagina 2
Edge	pagina 2
Android	pagina 2
Macchina a stati finiti sincrona	pagina 3
Macchina a stati finiti asincrona	pagina 4

## Funzionamento generale

Il progetto segue le linee guida descritte nella consegna. E' possibile da espruino controllare l'umidità, che viene ogni 3 secondi mandata al server mediante ngrok. Nel server è implementata la logica e in base all'umidità ricevuta attiva l'irrigazione. Inoltre l'umidità ricevuta viene spedita al client che la mostra sottoforma di grafico. Il controller arduino è implementato con macchina a stati finiti sincrona e scambia messaggi via bluetooth con android.

## Scelte implementative

L'umidità è misurata in percentuale e viene spedita al server che la salva sottoforma di JSON, salvandone timestamp e valore. Lo stesso avviene con i valori dell'irrigazione, mentre per la segnalazione viene salvato solo il timestamp in cui si verifica. Anche lo status, auto o manual, è salvato nel medesimo JSON. In base ai dati ricevuti vengono generati eventi che vengono poi gestiti dalla macchina a stati finiti asincrona del server. Tali valori vengono mostrati in tempo reale sul client, il quale fa polling con ajax ogni 5 secondi al server, e li mostra sottoforma di grafico grazie alla libreria chart.js. La comunicazione tra server e arduino e arduino e android è stata realizzata nel modo più semplice possibile. Se arduino non riceve messaggi rimane nello stato in cui si trova, altrimenti cambia stato a seconda del messaggio ricevuto. Ad esempio se ad un certo punto dal server, in modalità auto, riceve il messaggio con contenuto "30", allora significa che deve aprire l'irrigazione con portata 30. Per la comunicazione tra arduino e bluetooth abbiamo utilizzato i permessi e le tecniche viste a lezione. In particolare il bluetooth tenta la connessione ogni 10 secondi se non è già connesso, così se ci si avvicina e si passa in modalità manuale in un qualsiasi momento è garantita una connessione in pochi secondi.

## Controller

Il controller arduino è stato realizzato con una macchina a stati finiti sincrona e la suddivisione in task. In particolare abbiamo identificato 4 task:

1. Distance Detection Task che, mediante il sonar, controlla se ci sono persone a una distanza minore di 30cm.
2. Automatic Task che quando riceve un messaggio dal server, apre l'irrigazione con portata specificata nel messaggio
3. Irrigation Task che si occupa di aprire la pompa dell'acqua e settare l'intensità del LED2 a seconda di tale portata
4. Manual Task che si attiva se viene rilevata una presenza a una distanza inferiore ai 30cm e quando riceve un messaggio bluetooth attiva l'irrigazione con portata specificata nel messaggio.

## Server

Il Server è stato realizzato in Java mediante i pattern visti a lezione, come architettura event loop. Abbiamo inoltre utilizzato la libreria VertX per la realizzazione del server. Il server riceve l'umidità dalla componente Edge e la salva in un file JSON. In base all'umidità fa partire l'irrigazione, che viene scritta sul medesimo JSON, e un timer che calcola il tempo di irrigazione. Se l'umidità si alza sopra un certo valore l'irrigazione si ferma, altrimenti prosegue fino allo scadere del timer  $t = 6500\text{ms}$ , e viene generata una segnalazione che viene anch'essa scritta su file JSON. Tutto questo avviene se è in modalità auto, altrimenti il controllo della logica passa alla componente Android.

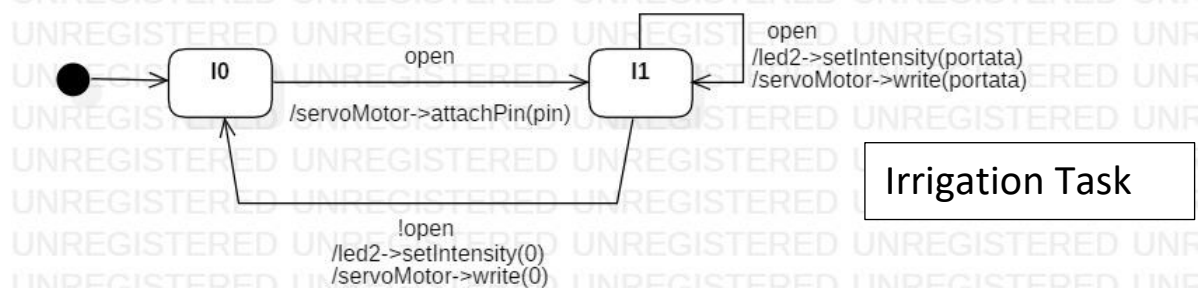
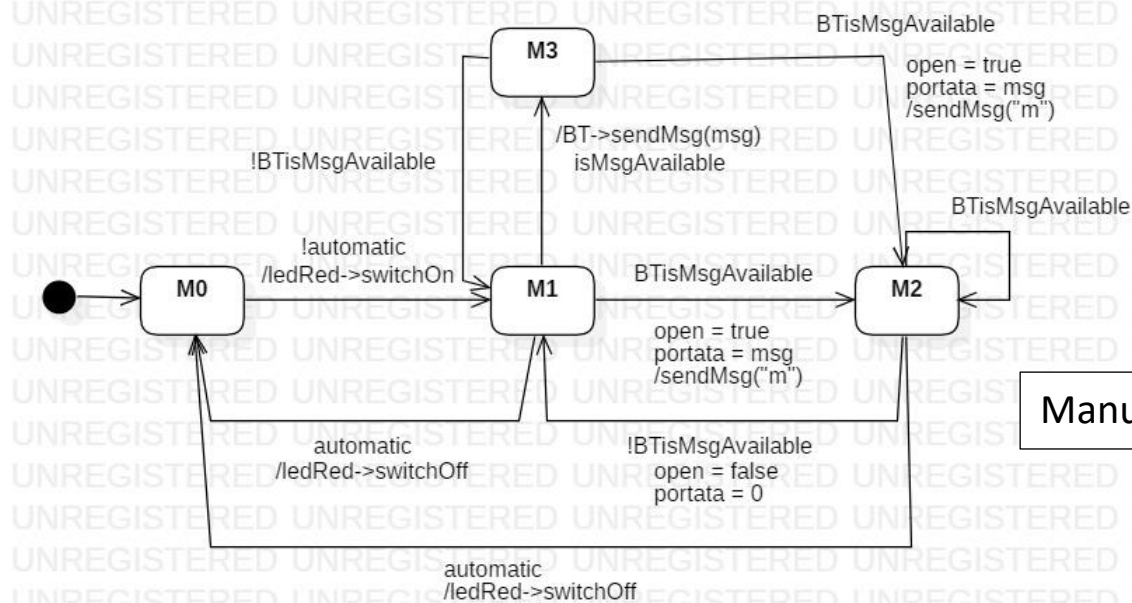
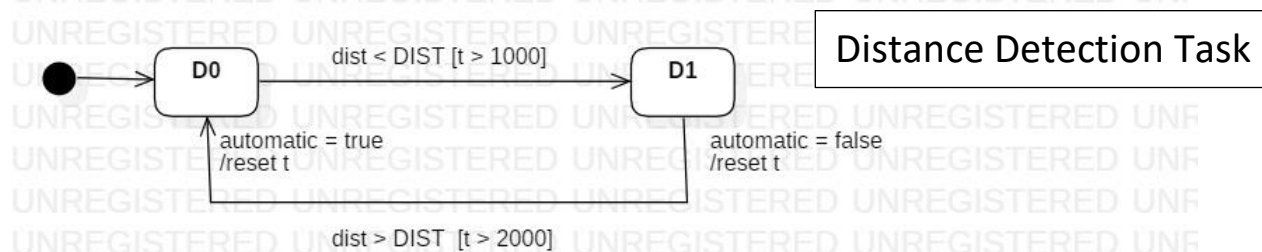
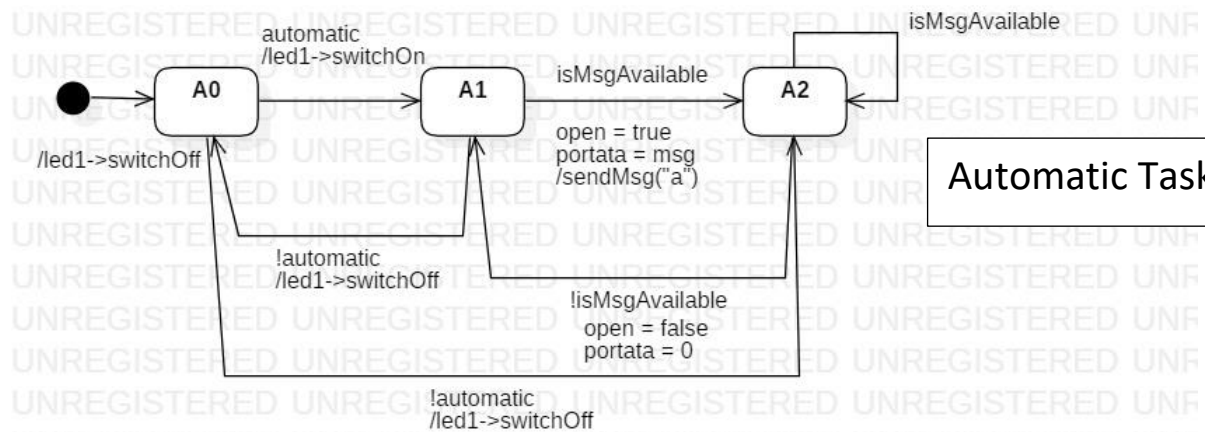
## Edge

La parte Edge è stata scritta in Javascript e fa uso di ngrok per passare i dati al server. Mediante una semplice proporzione il valore registrato dal potenziometro viene convertito in un valore percentuale e inviato al Server.

## Android

Nella parte Android sono stati aggiunti al Manifest i vari Intent e sono stati richiesti i vari permessi per la connessione bluetooth con Arduino. Il telefono, una volta che è stata rilevata una presenza nelle vicinanze dalla componente Controller può prendere il controllo della logica della serra, mandando ad Arduino messaggi specificando la portata dell'irrigazione. Inoltre è possibile visualizzare in tempo reale il valore dell'umidità presente nella serra.

## Macchina a stati finiti sincrona (Controller)



## Macchina a stati finiti asincrona (Server)

Il Server si basa su due concetti fondamentali, la macchina a stati finiti vera e propria e il thread che controlla l'invio e la ricezione dei messaggi. Quest'ultimo, in base ai messaggi ricevuti, si occupa di generare degli eventi che poi saranno gestiti dalla macchina a stati finiti.

