

# Smart Garden

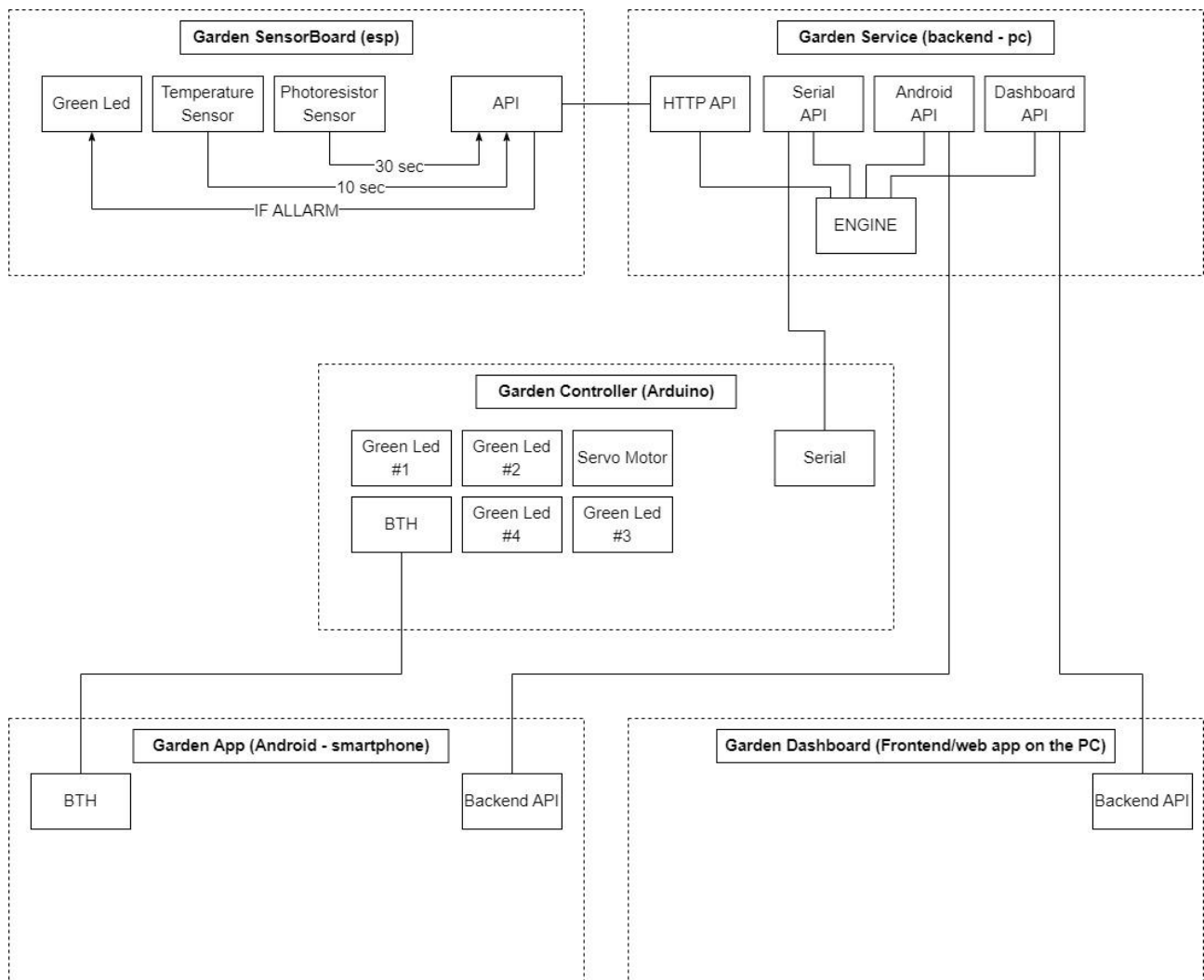
LORENZO BOMBARDINI - [LORENZO.BOMBARDINI@STUDIO.UNIBO.IT](mailto:LORENZO.BOMBARDINI@STUDIO.UNIBO.IT)  
STEFANO BABINI – [STEFANO.BABINI5@STUDIO.UNIBO.IT](mailto:STEFANO.BABINI5@STUDIO.UNIBO.IT)

# Sommario

- Overview ..... 2
- Garden Service..... 3
- Garden SensorBoard ..... 3
- Garden App..... 3
- Garden Dashboard ..... 4
- Garden Controller ..... 5
- Video dimostrativo: ..... 6

# Overview

Lo Smart Garden si compone di cinque unità in costante comunicazione tra loro. La sensoristica principale è installata sul SensorBoard, la quale invia le informazioni al Service. Il service elabora le informazioni, espone delle API REST e gestisce il Controller comunicando tramite un canale seriale. La Dashboard comunica con le API del Service ed espone le informazioni all'utente. L'app comunica con le API del service e direttamente con il controller tramite un modulo bluetooth.



# Garden Service

Hardware: Pc

Software: Python 3, Flask

Il Garden Service è il centro dello Smart Garden, si compone di due thread separati: uno per esporre le API web di Flask, e un thread, che viene eseguito ogni 10 secondi, con lo scopo di aggiornare le variabili di sistema in base a quanto rilevato dalla SensorBoard e inviare il risultato dell'elaborazione tramite Seriale al Controller.

# Garden SensorBoard

Hardware: ESP32-S3

Software: Wiring

Garden SensorBoard legge i dati dall'ambiente circostante e li invia al service mediante delle chiamate Post alle API Rest esposte dal Service.

Garden SensorBoard legge il sensore di temperatura e legge la fotoresistenza mappando i valori in un range da 0 a 7 inclusi. Compone poi un file JSON e lo invia in POST alle API del Service.

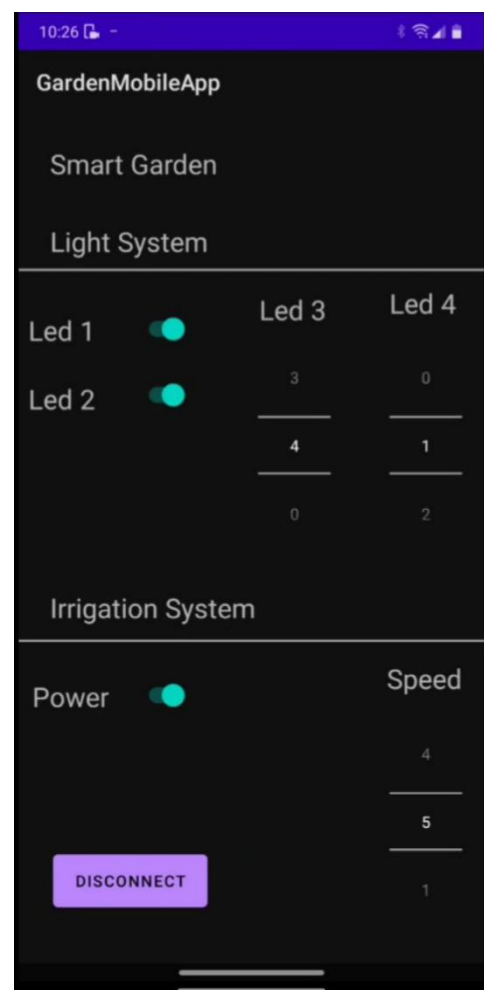
# Garden App

Hardware: Smartphone Android

Software: Android Sdk, Java

Garden App si interfaccia con il Service tramite API e controlla che il sistema non sia in stato di ALARM, nel caso permette di disattivarlo premendo sull'icona rossa con il punto esclamativo.

Garden App permette di collegarsi in modo manuale al Controller, per fare ciò viene avviata una routine che prevede una chiamata API al Service, il quale controlla l'avvenuto collegamento del Controller al Bluetooth e il passaggio dello stato del sistema in MANUAL poi conferma all'APP di poter procedere con il collegamento e sbloccare l'interfaccia di controllo all'utente. Una routine inversa viene eseguita per la disconnessione.

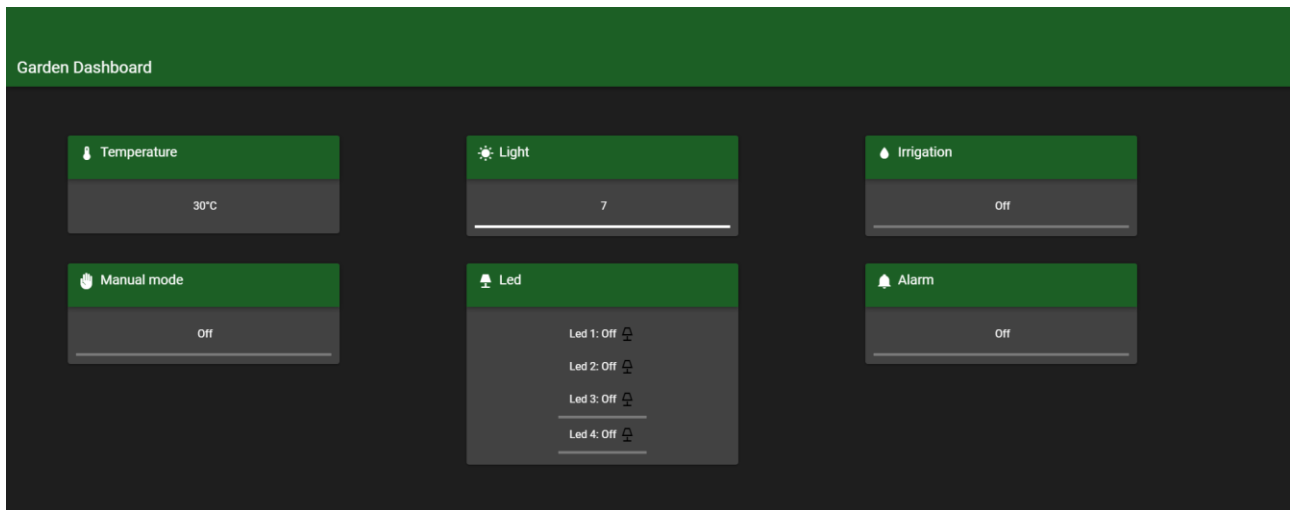


# Garden Dashboard

Hardware: Pc-Docker

Software: Vuejs, Vuetify, Javascript

Garden Dashboard è un container Docker che al suo interno presenta un webserver NGNIX che espone sulla porta 8080 un frontend costruito con un framework javascript, Vue. La dashboard visualizza varie Card indipendenti l'una dall'altra che sono autonome nel chiamare ciascuna la propria route API.



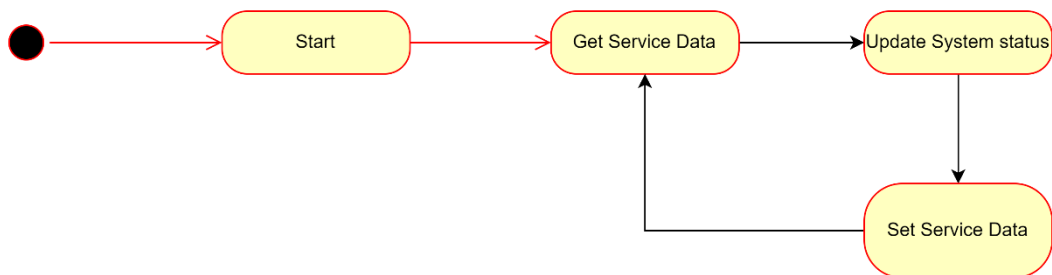
# Garden Controller

Hardware: Arduino UNO

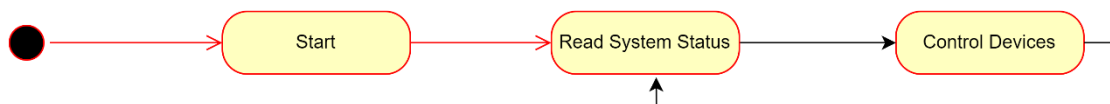
Software: Wiring

Garden Controller è l'attuatore di tutto il sistema SG, contiene il sistema di illuminazione e il sistema di irrigazione. Comunica attraverso il canale seriale al Service e attraverso il modulo bluetooth al Garden APP. Si compone di tre FSM sincrone.

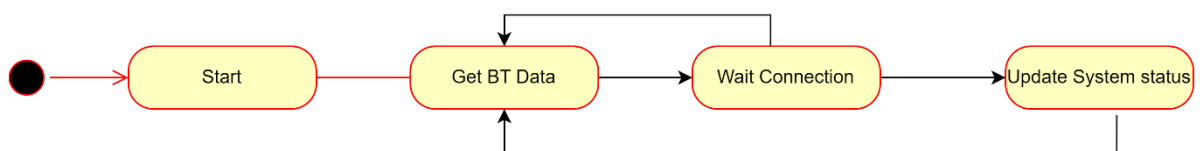
SerialCommunicationTask Sync FSM



DeviceControllerTask Sync FSM



BTSerialCommunicationTask Sync FSM



- `SerialCommunicationTask`:

Ha il compito di comunicare con il Service, inizializza i propri componenti poi si aspetta di ricevere comunicazioni sul seriale. Se lo stato del sistema che riceve è AUTO provvede ad aggiornare lo stato degli attuatori secondo i valori ricevuti, poi risponde al Service con un valore che rappresenta la connessione o meno del Controller all'app.

- `BTSerialCommunicationTask`:

Ha il compito di comunicare con GardenApp attraverso il modulo Bluetooth, se riceve lo stato di tentata connessione aggiorna lo stato di connessione che poi verrà comunicato da un altro task al Service. Se lo stato del sistema è MANUAL provvede ad aggiornare lo stato degli attuatori secondo i valori ricevuti.

- `DeviceControllerTask`:

Legge lo stato attuale degli attuatori e implementa tutta la logica necessaria al loro funzionamento.

## Video dimostrativo:

[Assignment03.mp4](#)