

Relazione del progetto “Smart Room”

Giacomo Amadio e Daniel Pellanda

26/04/2023

Indice

1	Analisi	2
1.1	Requisiti	2
1.2	Analisi e modello del dominio	3
2	Architettura	4
2.1	Arduino - Room controller	4
2.2	ESP8266 - Sensors	6
2.3	Android - Mobile App	6
2.4	Backend - Room Service	6
2.5	Frontend - Dashboard	7
3	Extra	8
3.1	Schema breadboard	8
3.2	Video dimostrativo	9

Capitolo 1

Analisi

1.1 Requisiti

Smart Room è un sistema di gestione e controllo intelligente di una stanza per regolare l'illuminazione e l'apertura o chiusura delle persiane.

Requisiti funzionali

- Smart Room dovrà essere in grado di gestire autonomamente accensione o spegnimento delle luci e apertura o chiusura delle persiane monitorando lo stato della stanza.
- Smart Room fornirà all'utente un'interfaccia web per tenere traccia dello stato della stanza e per regolare in maniera manuale quest'ultimo.
- Smart Room renderà possibile all'utente collegarsi con la stanza via app per mobile usando il bluetooth.

Requisiti non funzionali

- Smart Room dovrà garantire la responsività della stanza a cambiamenti dell' ambiente o in base alle preferenze dell' utente.

1.2 Analisi e modello del dominio

Smart Room per gestire la stanza deve raccogliere dei dati dall'ambiente circostante con i **Sensori** per poi comunicarli al **Backend** che li fornisce al **Controller** il quale eventualmente si occuperà di modificare lo stato della stanza.

Smart Room offre all'utente la possibilità di collegarsi alla stanza via web con **Frontend** per monitorarne lo stato e controllare luci e pesiane da remoto.

Smart Room per offrire all'utente maggiore controllo sulla stanza permette di collegarvisi via bluetooth utilizzando dispositivi mobili con **Room-app**, così da poterla gestire con comodità quando si è nelle vicinanze.

Gli elementi costitutivi il problema sono sintetizzati in Figura 1.1.

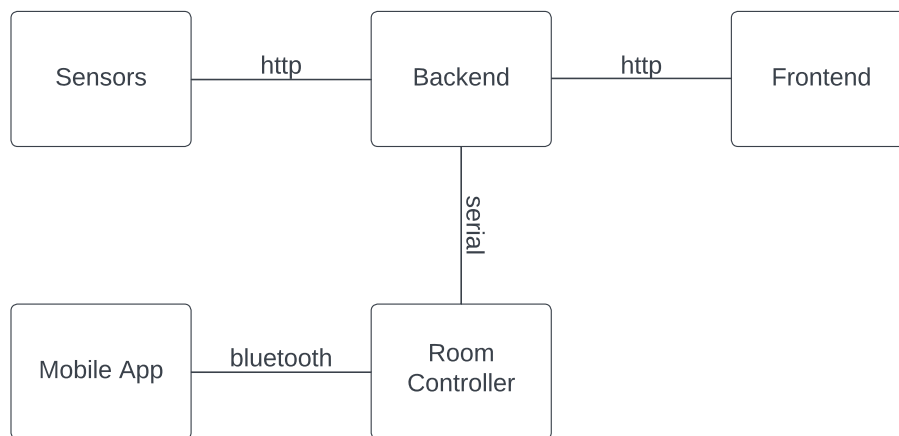


Figura 1.1: Schema dell'analisi del problema, con rappresentati i componenti principali ed i rapporti fra loro

Capitolo 2

Architettura

2.1 Arduino - Room controller

Per il `Room controller` è stata utilizzata un'architettura task-based e una Finite State Machine sincrona per definire il comportamento dei task, rappresentata in Figura 2.1.

FSM

Per quanto riguarda la FSM sono stati definiti 3 stati per poter distinguere i vari tipi di controllo a cui è sottoposta la stanza.

- `AUTO` – In questo stato la stanza legge i dati dai sensori di movimento e luminosità e si gestisce autonomamente.
- `REMOTE` – In questo stato la stanza viene controllata manualmente da remoto con `Room Dashboard`.
- `BLUETOOTH` – In questo stato la stanza viene controllata manualmente via bluetooth con `Room app`.

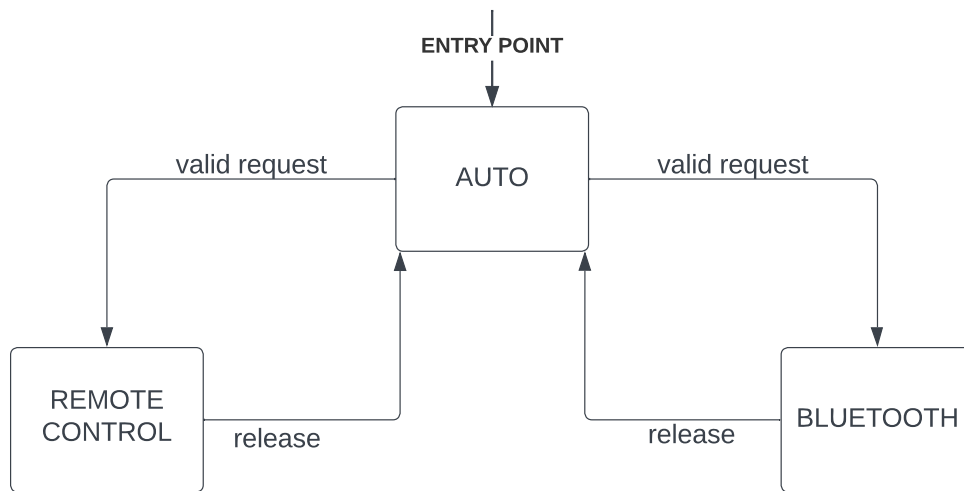


Figura 2.1: Schema della Finite State Machine di Room controller

Task

Utilizzando un'architettura task-based che si appoggia su una FSM sincrona, i task vengono messi periodicamente in esecuzione da uno scheduler. Nella soluzione da noi proposta sono stati usati questi 3.

- **CLOCK** – Si occupa di aggiornare un clock software, per simulare lo scorrere del tempo.
- **COMMUNICATION** – Si occupa delle comunicazioni via seriale con **Room Service** e via bluetooth **Room app**.
- **ROOM CONTROL** – Si occupa della gestione della stanza in base ai dati ricevuti.

2.2 ESP8266 - Sensors

Per la `Room sensor board` è stata utilizzata un'architettura task-based.

Task

Utilizzando un'architettura task-based i task vengono messi periodicamente in esecuzione da uno scheduler. Nella soluzione da noi proposta sono stati usati questi 2.

- **SENSORS CHECK** – Si occupa di leggere i valori dei sensori e aggiornare lo stato del led.
- **HTTP COMMUNICATION** – Si occupa della comunicazione via http dei rilevamenti dei sensori a `Room service`.

2.3 Android - Mobile App

La `Room Mobile App` permette all'utente di prendere il controllo remoto della stanza potendo scegliere se accendere o spegnere le luci oppure se aprire o chiudere le tende.

Activities and Fragments

L'applicazione è composta da una singola attività divisa da due fragment principali:

- **LOAD FRAGMENT** – Questo è il fragment lanciato all'avvio dell'attività. Si occupa di impostare la connessione Bluetooth con il `Room Controller`, predisponendo un'interfaccia dove l'utente può selezionare il dispositivo a cui collegarsi.
- **FORM FRAGMENT** – Dopo aver impostato la connessione, questo fragment si occupa dell'invio delle richieste di controllo remoto della stanza.

2.4 Backend - Room Service

`Room Service` si occupa della gestione dell'interscambio di dati fra `Controller`, `Sensor Board` e `Dashboard`.

Thread

Per garantire una comunicazione più robusta e responsiva la soluzione proposta fa uso di 2 thread:

- HTTP COMMUNICATION – Si occupa di predisporre il server HTTP dalla quale **Sensor Board** e **Dashboard** potranno ricevere e inviare i propri dati.
- SERIAL COMMUNICATION – Si occupa della comunicazione seriale con **Room Controller**.

2.5 Frontend - Dashboard

La parte di **Dashboard** consiste in una Web App che fornisce all'utente una panoramica dello stato corrente della stanza, inoltre permette anche di assumere il suo controllo da remoto.

La Web App reperisce i dati della stanza facendo periodiche richieste **GET** al server HTTP (predisposto dal **Room Service**), con essi provvede ad aggiornare il grafico e i vari parametri della panoramica.

Le richieste di controllo della stanza sono inviate al server tramite messaggi **POST** contenenti i nuovi parametri d'applicare.

Capitolo 3

Extra

3.1 Schema breadboard

Arduino

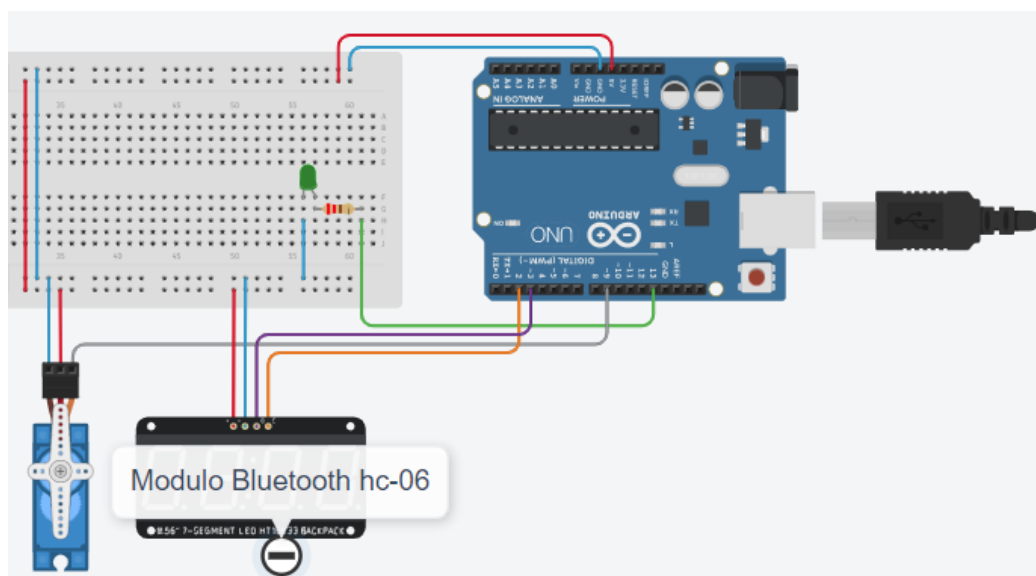


Figura 3.1: Schema del circuito della scheda di controllo della stanza

ESP8266

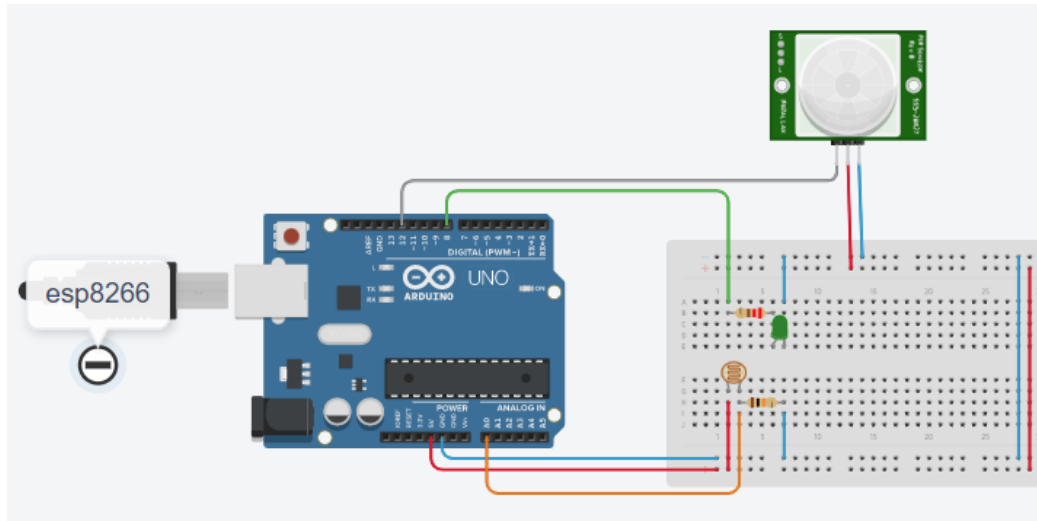


Figura 3.2: Schema del circuito della scheda di controllo dei sensori

3.2 Video dimostrativo

<https://www.youtube.com/watch?v=m94VzzqPApo>