**­­**

**Smart Room**

***Sistema di automatizzazione della gestione di una Smart Room***

Muccioli Federico

federico.muccioli5@studio.unibo.it

0000971342

Room Mobile App

Descrizione

La room mobile app consente di gestire la smart room tramite la tecnologia Bluetooth, comunicando direttamente con il room controller. Questa applicazione è stata sviluppata in Java per Android e installata e testata su un dispositivo fisico dotato di sistema operativo Android 11.

L'app presenta due principali schermate o attività: la prima consente di gestire i dispositivi Bluetooth, mentre la seconda si occupa della connessione e dell'invio di messaggi.

Discovery activity

Per ottenere l'elenco dei dispositivi bluetooth è necessario richiedere l'attivazione del bluetooth del dispositivo e ottenere i permessi. Una volta fatto ciò si otterrà una lista di dispositivi già accoppiati col device e la possibilità di effettuare una discovery per ottenere l'elenco dei dispositivi invece disponibili. Eseguita una discovery i dispositivi disponibili invieranno un segnale intercettato poi dal BroadcastReceiver venendo poi aggiunti alla lista dei dispositivi disponibili.

Per quanto riguarda la UI è stato utilizzato un linear layout vertical che si adatta automaticamente alle dimensioni del device e presenta due ListView con i nomi dei dispositivi.

Alla ListView dei dispositivi disponibili viene impostato un Adapter che riceve come argomento una lista di stringhe con i nomi dei dispositivi. Quando viene rilevato un nuovo dispositivo, la lista di stringhe viene aggiornata e viene richiamato il metodo notifyDataSetChanged() dell'Adapter, per aggiornare la ListView.

Per stabilire una connessione, è sufficiente selezionare il dispositivo desiderato da una delle ListView e si verrà reindirizzati alla seconda schermata dell'applicazione.

Immagine che contiene tavolo

Descrizione generata automaticamente

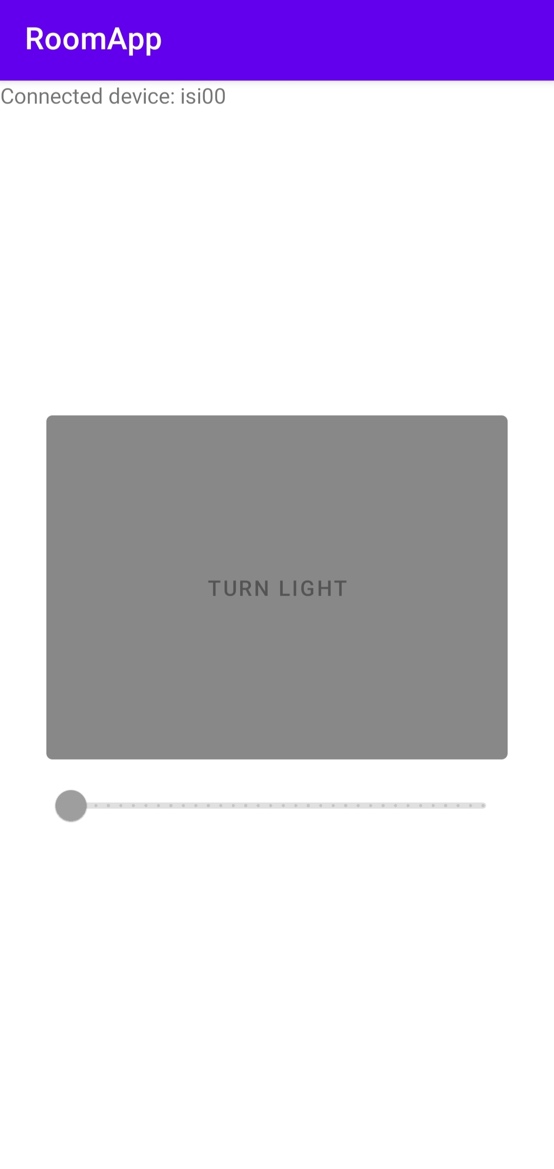
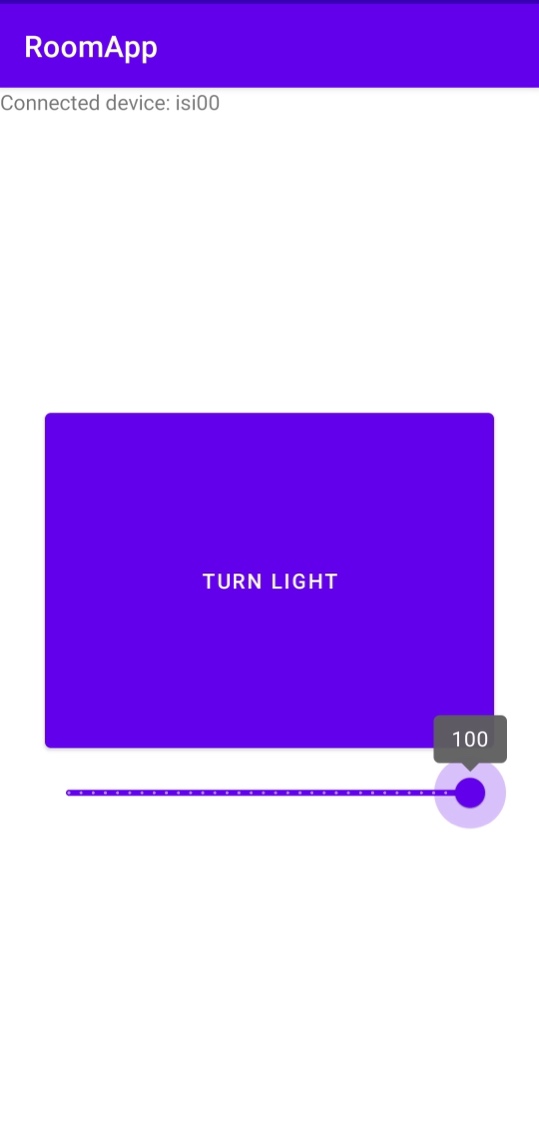
Controller activity

La schermata presenta una text view con il dispositivo al quale si sta tentando di connettersi, un bottone per accendere/spegnere la luce e uno slider per impostare le tapparelle da 0 a 100.

Come layout anche qui è stato utilizzato un LinearLayout questa volta orizzontale con impostazione gravity = center.

Finché la connessione con il dispositivo non sarà istanziata gli elementi dell'interfaccia saranno disabilitati e mostrati da una colorazione grigia.

Per la connessione sarà istanziato un nuovo thread così come per l'invio dei comandi e queste operazioni saranno delegate alla classe BluetoothRoomChannel. Questa classe prenderà come parametri del costruttore il device al quale ci si vuole connettere, l'identificativo (UUID) e un handler definito con un unico metodo enable che permette in questo caso di attivare i componenti dell'attività. Una volta istanziato un BluetoothRoomChannel si avrà un socket tramite il quale richiamando il metodo run della classe verrà eseguita una connessione con un nuovo thread ottenendo l'output stream e, se senza errori, verranno abilitati i componenti attraverso il metodo enable dell'handler. D'ora in poi per inviare messaggi basterà richiamare i relativi metodi di questa classe turnLight() e setRollerBlinds(percentage) che istanzieranno nuovi thread per la codifica e comunicazione.

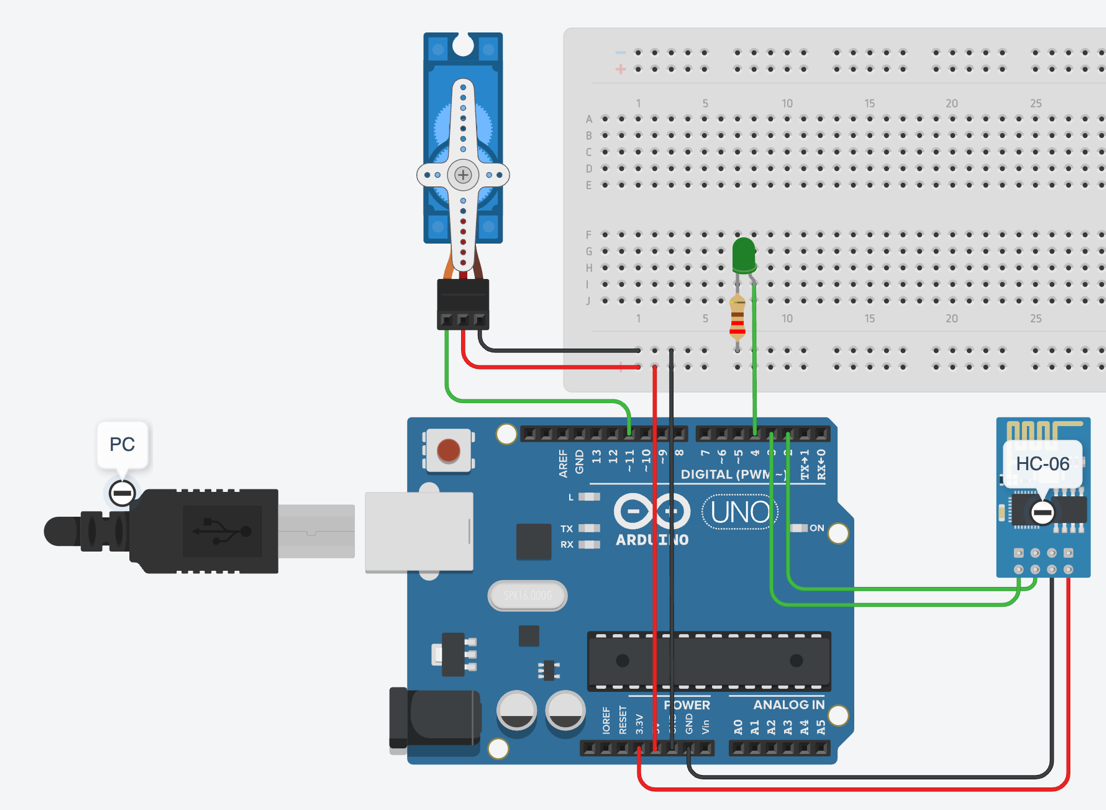
 

Room Controller

Descrizione

Il sottosistema del room controller è composto dal microcontrollore Arduino, il quale è collegato tramite una porta seriale al room service e tramite il modulo bluetooth HC-06 alla room app. Il microcontrollore Arduino riceve gli ordini di controllo e li esegue, pilotando la luce (simulata da un led) e le tapparelle (simulate da un servomotore). Una volta completata l'esecuzione del comando, Arduino aggiorna il room service con lo stato attuale dei componenti controllati, fornendo così informazioni sulla loro posizione e stato di accensione o spegnimento.

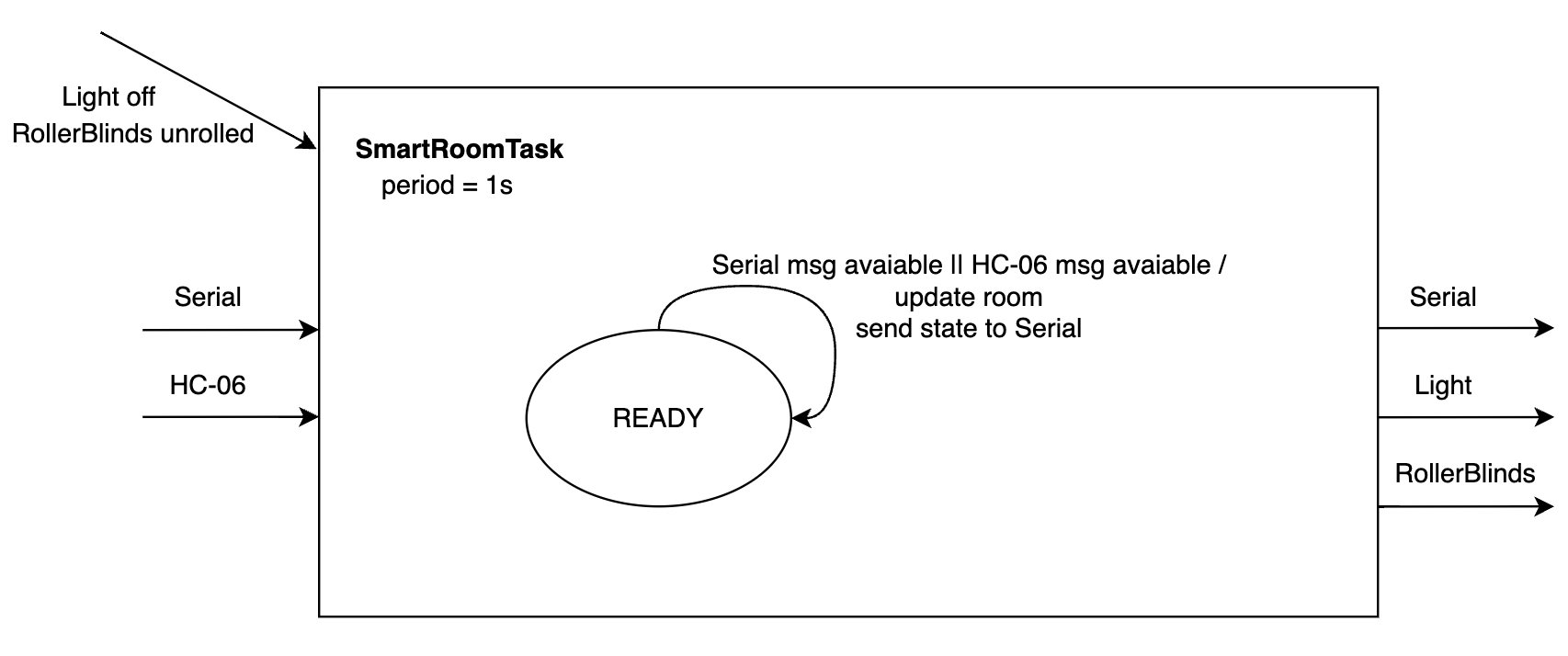
Schema elettrico



Scheduling e Task

Nell'implementazione del sistema è stato definito un unico task che si trova costantemente nello stesso stato, il quale verifica la ricezione di un messaggio, ne decodifica il contenuto, esegue il comando sugli attuatori e invia una notifica al room service contenente i valori attuali dei componenti controllati.

Per l'esecuzione del task è stato scelto un periodo di un secondo, in modo da garantire un'esecuzione abbastanza rapida dei comandi dal punto di vista dell'umano senza eccessiva reattività, limitando i consumi. Inoltre, per limitare sempre i consumi del microcontrollore durante l'attesa dell'esecuzione del task, è stato impostato lo stato di sleep di Arduino con funzione IDLE, che consente comunque l'esecuzione dei timer necessari al funzionamento delle periferiche.

Al fine di garantire l'estendibilità del sistema in caso di integrazione di nuovi task, è stato implementato uno scheduler periodicizzato dal timer uno attraverso la libreria TimerOne.h.

Comunicazione bluetooth

Poiché i pin 0 (RX) e 1 (TX) sono già utilizzati per la comunicazione seriale tra Arduino e il PC, il modulo Bluetooth è stato configurato come un dispositivo seriale tramite la libreria SoftwareSerial.h. In particolare, la comunicazione è stata simulata tramite i pin 3 e 4. Tuttavia, a causa della mancanza degli interrupt su questi pin, la verifica della ricezione di un messaggio avverrà solo tramite polling.

Scelta pin

Poiché la libreria utilizzata per la comunicazione seriale sfrutta il timer1, è stata utilizzata un'altra libreria che utilizza il timer2 per gestire il servo motore. Tuttavia, poiché il timer2 è anche utilizzato per la modulazione di larghezza di impulso (PWM) sui pin 3, 11, non è possibile utilizzare questa funzionalità quando il motore è attivo su questi pin. Per questo motivo, si è scelto di utilizzare i pin 3 e 11 per i dispositivi attuali che non richiedono la PWM, al fine di lasciare liberi questi pin per eventuali dispositivi futuri che ne necessitano.

Gestione comunicazione

Poiché i comandi possono essere impartiti al room controller sia dalla room app che dal room service, è importante che il room service sia sempre a conoscenza dello stato attuale della stanza. Per questo motivo, ad ogni iterazione, Arduino deve inviare un messaggio di notifica al service contenente lo stato attuale dei componenti controllati, in modo da garantire che lo storico delle informazioni sia sempre corretto e aggiornato.

Poiché il room controller può ricevere messaggi da diversi sistemi contemporaneamente, è stato necessario adottare una soluzione che permettesse di gestire efficacemente l'arrivo di questi messaggi. Per questo motivo, è stata implementata una sorta di cache attraverso una lista FIFO con capacità massima di dieci messaggi, sufficiente per il periodo del task di un secondo.

Protocollo comunicazione

I messaggi con il room controller avvengono attraverso uno scambio di stringhe. La prima lettera della stringa definisce l'attuatore: "l" per light e "r" per le roller blinds. I caratteri rimanenti rappresentano una cifra numerica, che nel caso delle tapparelle rappresenta la percentuale di apertura compresa tra 0 e 100, mentre nel caso della luce, 0 rappresenta la luce spenta, 1 rappresenta la luce accesa e 2 rappresenta il cambio di stato attuale. Il room service riceverà un comando relativo a un singolo componente alla volta, mentre la notifica inviata dal room controller al room service avverrà tramite una singola stringa che conterrà i valori attuali dei vari componenti controllati, separati da una e commerciale "&".

Room Service

Descrizione

Il room service si occupa di diverse mansioni, riceve input dai sensori della room sensor board che poi elabora per impartire comandi, comunica con il room controller per impartire comandi e ricevere lo stato della stanza ogni volta che varia, memorizza lo storico di tutte le variazioni della smart room che poi può essere richiesto dalla dashboard. In pratica il room service è il sistema principale di gestione della smart room che comunica con gli altri sistemi utilizzando diverse tecnologie.

Room controller

Il room service comunica con il room controller attraverso la seriale per impartire ordini e ricevere lo stato dei componenti della stanza. La comunicazione avviene attraverso la classe RoomCommChannel che estende la classe SerialCommChannel che gestisce effettivamente la comunicazione seriale tramite la libreria jssc. Il room service quando avrà bisogno di impartire ordini al room controller sfrutterà i metodi setLight e setRollerBlinds della classe RoomCommChannel che effettua una codifica dei messaggi in base al protocollo di stringhe descritto nel paragrafo riguardante il room controller e invierà i messaggi attraverso il metodo sendMsg della classe da cui ha esteso. Quando invece Arduino comunicherà lo stato della stanza al programma Java, questo messaggio verrà intercettato, decodificato e verrà richiamato il metodo del controller che lo memorizzerà in un file Json.

Room Dashboard

ecc...