**­­**

**Smart Room**

***Sistema di automatizzazione della gestione di una smart room***

Muccioli Federico

federico.muccioli5@studio.unibo.it

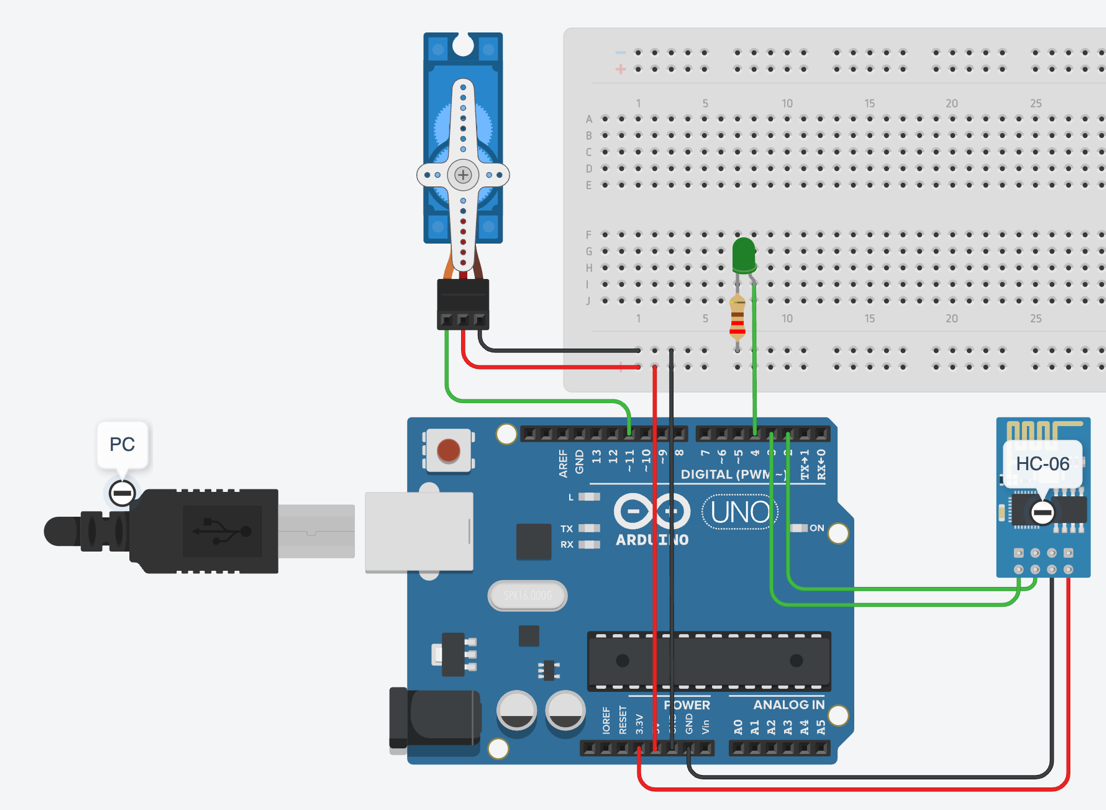
0000971342

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.

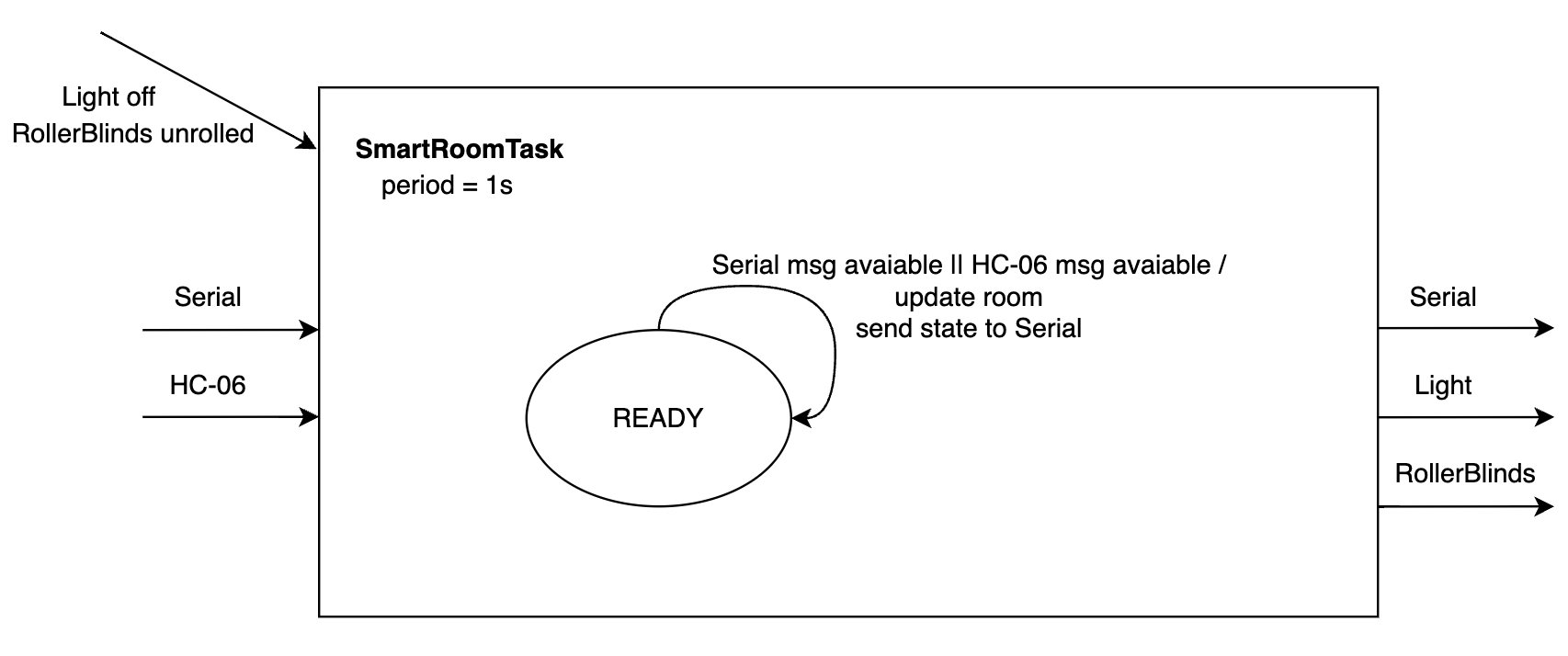
Schema elettrico



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 contenente lo stato attuale dei componenti controllati, in modo da garantire che lo storico delle informazioni sia sempre corretto e aggiornato.

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 "&".