10/11/2017 Consegna n. 2 Simone Magnani - 754082

Smart Garage

Premessa

Per la seconda consegna, ho deciso di strutturare il progetto in n.3 Task principali: il GarageManager, il ParkAssistant e il SerialManager. La scelta è motivata dal fatto che dalla consegna ho individuato appunto 3 compiti distinti, ovvero la gestione del Garage, l'assistenza al parcheggio e la gestione della seriale. In particolare quest'ultimo Task è stato aggiunto per semplificare il codice e realizzare un programma concettualmente più corretto.
Analizzerò i vari compiti nelle varie sezioni.

SerialManager

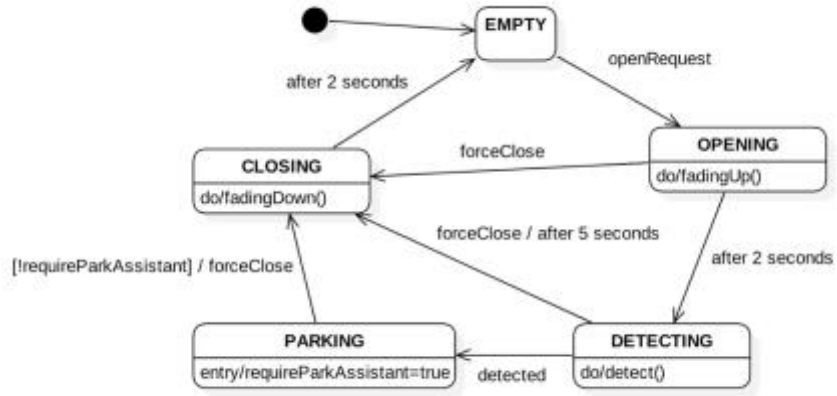
Ho implementato questo task responsabile della lettura dei dati in arrivo nella seriale con la quale Arduino comunica con l'interfaccia grafica Java. Il task non è a stati, o meglio è mono stato, in quanto dopo una riflessione ho deciso di non implementare all'interno di questo task la parte complementare, ovvero la scrittura su seriale. Essendo adibito dunque solo alla lettura, ho deciso di mettere un'unica funzione che legge in continuazione i dati (se disponibili). E' interessante notare che nel caso non ci siano più dati da leggere, la stringa risultante è inizializzata nuovamente a stringa vuota: utilizzando il principio della macchina a stati finiti sequenziale e decidendo come poter posizionare i task in ordine di esecuzione, tale soluzione non presenta alcun problema di concorrenza/inutilizzo della stringa letta. Mi spiego meglio con un esempio:

Caso d'uso: Il programma è in attesa della stringa "open" per aprire il garage, e dalla seriale riceve in sequenza i caratteri o-p-e-n.
Dilemma: Il task che attendeva la scritta open riesce a leggere "open" prima che il SerialManager reinizializzi la stringa?
Risposta: assolutamente si, in quanto la stringa viene aggiornata di un char alla volta; l'ultimo char che riceve è 'n', ed il ciclo seguente di clock il SerialManager reinizializzerebbe la stringa, ma prima di reinvocare il SerialManager lo scheduler attiva il task in attesa di tale stringa che riuscirà a leggerla efficacemente.

Per ogni altro caso d'uso (ad esempio il programma riceve dati quando i task non ne necessitano), grazie al lavoro di tale task vengono letti e scartate tale informazioni, in quanto sono arrivate in un momento che il programma non ne necessitava.

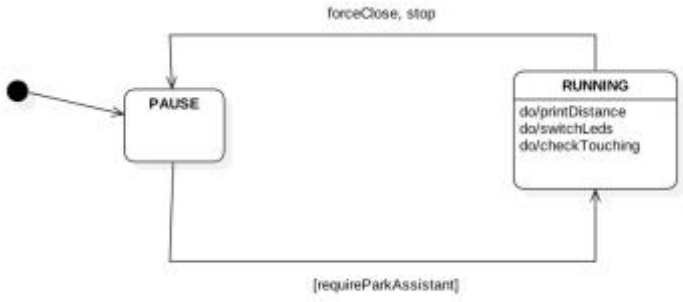
GarageManager

Per introdurre tale task allego un'immagine che riporta i vari stati con gli eventi che scatenano i vari cambiamenti:



Come si può osservare, lo stato iniziale è EMPTY e in questo stato l'unica cosa che il programma ha bisogno di fare è aspettare una richiesta d'apertura (SerialManager legge "open"). In seguito passa allo stato OPENING, in cui esegue il fading del led e, se non viene richiesta una chiusura forzata, passa a DETECTING, in cui verifica se effettivamente vi è la macchina. A questo punto vi sono due possibilità: o la macchina viene rilevata e viene messo in azione il task ParkAssistant, oppure non viene rilevata (o chiusura forzata) e il sistema si chiude e torna nello stato iniziale.
Dallo stato PARKING il task esce o tramite chiusura forzata (il ParkAssistant farà i suoi check per verificare se effettivamente la richiesta può essere accettata), o quando il ParkAssistant ha finito il suo lavoro.
Si poteva suddividere il problema in vari sotto problemi e sotto task, ad esempio crearne uno apposito per i vari fading che il sistema deve fare, ma la scelta di inserire queste operazioni all'interno di tale task è data dal fatto che in ogni caso le specifiche forniscono indicazioni di "sequenze d'azione e di tempo" che suggeriscono di non suddividere il task in più tasks (Se avessi creato un task apposito per la gestione delle luci, il GarageManager avrebbe comunque impiegato quei due secondi ad aspettare inutilmente il completamento del fading).

ParkAssistant



Il task adibito all'assistenza al parcheggio è stato il più semplice da progettare, ma non per questo il più semplice da implementare. Il task presenta due semplice stati: PAUSE e RUNNING. Da PAUSE il sistema esce solamente quando viene richiesta l'assistenza al parcheggio (requireParkAssistant=true dal GarageManager) e in questo stato ci ritorna o tramite una buona riuscita di una chiusura forzata, o quando è verificato l'evento stop dalla seriale (lettura di "stop"). Durante lo stato di RUNNING vengono eseguite tutte le operazioni richieste dalla consegna, ovvero vengono settati i led proporzionalmente alla distanza rilevata, viene stampato a video la distanza e, nel caso certe condizioni siano verificate, altre indicazioni relative al parcheggio (come il Touching e il Ok can stop). Ho scelto di non suddividere tali problemi in vari stati, in quanto concettualmente mi sembrava più corretto impostarlo cosi, dato che tutte le operazioni sono eseguite solamente se il Task è attivo (esempio congruo alla vita reale).