

Progetto #2 - Smart Exp

by Giovanni Messina (giovanni.messina3@studio.unibo.it) matricola: 0000890082

Il progetto è stato modellato con un approccio a stati nel loop principale.

Si avrà uno stato iniziale, preExp, in cui si settano tutti i valori per iniziare l'istanza dell'esperimento. E' inoltre implementato lo sleep del sistema tramite la libreria *avr* che addormenta il microcontrollore se il pir non rileva oggetti e lo risveglia alla rilevazione tramite interrupt.

Premendo il pulsante Start si cambia stato e parte l'esperimento: quando il sonar rileva un qualsiasi oggetto entro il suo raggio massimo, parte uno scheduler con i task e il relativo periodo, elaborato in modo che il valore scelto dal potenziometro sia un multiplo di 10, così da evitare criticità nello scheduler.

Lo scheduler darà inizio a una macchina a stati finiti sincrona integrata a task.

Questi task sono i componenti che dovranno rilevare i dati all'esterno (temperatura, sonar) oppure riportare i dati elaborati (servo, printCine; quest'ultimo ha il compito di mostrare nella seriale posizione, velocità e accelerazione).

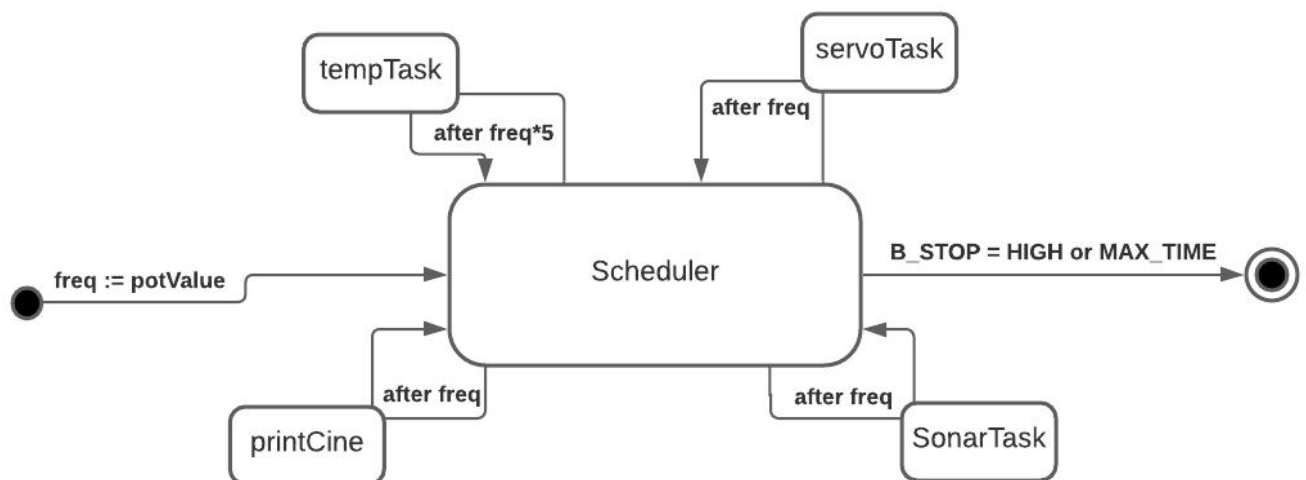


Fig.1 - Diagramma degli stati con implementazione dello scheduler.

Lo scheduler è lo stato master che richiama, ad ogni periodo denominato *freq*, i task collegati ad esso.

A questo punto, lo scheduler inizia ad eseguire i task secondo il periodo. Solo la task della temperatura viene eseguita ogni 5 "giri", in quanto la temperatura, necessaria al calcolo corretto del sonar, non ha cambiamenti rapidi. A fine esperimento si attenderà una conferma della seriale per poter ritornare allo stato iniziale.

La comunicazione tra il microcontrollore e Viewer avviene tramite un linguaggio, molto

semplice, che specifica all'applicazione lo stato in cui si trova Arduino.

Librerie utilizzate:

-Arduino-

<avr/sleep.h> - per gestire la sleep mode

<TimerOne.h>- per gestire il timer separato dello scheduler

<ServoTimer2.h>- poichè Servo.h gestisce il Timer1 e con lo scheduler ci sarebbe conflitto, ho usato questa libreria che lo gestisce con il Timer2

<DHT.h> usata per gestire il sensore di temperatura e umidità DHT11

<MapFloat.h> usata per mappare i valori del Servo con i valori della velocità

-Java-

jfreeChart usata per creare i grafici

Problematiche riscontrate:

- Se il periodo è > 500ms il sistema si blocca all'avvio dello scheduler, questo succede anche con solo un task (anche solo con la stampa nella seriale, il printCine non stampa nulla, segno che non esegue il task).
E' strano perchè ha tutto il tempo per eseguire il task, capirei se fosse il contrario e quindi troppo piccolo il periodo e non abbia avuto il tempo di completare il task.
- Alcune volte, all'avvio dello scheduler, i task sembra che si addormentino e non partano, pertanto è necessario spegnere e riaccendere l'Arduino per "risvegliare" i task. Succede così raramente che non è stato possibile trovare il bug.