# Assignment 02 Smart Coffee Machine

Durante la fase di progettazione ho identificato in base al testo del problema una suddivisione in task che segue il naturale comportamento di una macchinetta del caffè.

Ho pensato di coordinare tra loro i task tramite delle variabili condivise (principalmente booleane) che informano i vari task sul compimento delle attività (es: fine produzione bevanda, consegna bevanda ecc.). Alcuni task hanno vincoli temporali (es: self-test effettuato periodicamente), altri sono vincolati dalla conclusione di un altro task (es: consegna della bevanda solamente in seguito alla produzione della stessa).

Oltre alle variabili di coordinazione dei task ho utilizzato delle variabili condivise per memorizzare la lista e le quantità dei prodotti disponibili, e due variabili per gestire rispettivamente il display e il servo motore il cui utilizzo è richiesto da più task.

Di seguito è presentata brevemente la suddivisione del problema in 6 task:

* Selection\_Task: si occupa della parte di selezione del prodotto, gestisce quindi i tre bottoni e il potenziometro. Esso è il primo di tre task complementari: selezione, produzione bevanda e consegna. Queste tre attività sono semanticamente indipendenti ma hanno dei vincoli che richiedono di eseguirle in un preciso ordine, perciò ho pensato di strutturarle in maniera simile. Per rappresentare questo vincolo ho deciso di identificare in ogni task uno stato di attesa del task precedente.
* Making\_Task: ha il compito di produrre la bevanda selezionata, gestisce quindi il movimento del servo motore, appena l’attività di selezione si è conclusa.
* Delivery\_Task: come precedentemente descritto, questo task è inizialmente in uno stato di attesa della produzione del prodotto. Successivamente si occupa tramite il sonar di verificare se la bevanda viene prelevata ed infine resetta la posizione del servo a consegna avvenuta.
* User\_Detecton\_Task: ha l’obbiettivo di rilevare tramite il PIR la presenza di un utente e di conseguenza di far entrare il sistema in sleep quando non è presente nessun utente per un determinato lasso di tempo. Questo task si alterna tra due stati (detected e not detected) nei quali agisce conseguentemente al suo scopo.
* Check\_Task: viene chiamato periodicamente, ha uno stato di attesa nel quale controlla di poter iniziare il check e due stati che identificano rispettivamente il check della temperatura e il self-test.
* Send\_Info\_Task: ha il compito di comunicare tramite la seriale con il Coffee Machine Manager. I suoi stati identificano 3 diversi comportamenti: la verifica della modalità della macchinetta (Idle, Working, Assistance), l’invio di messaggi e la ricezione. Al suo interno compone un messaggio (utilizzando le informazioni date dalle variabili globali) in un particolare formato, che verrà poi letto e interpretato lato Java, in modo da aggiornare la GUI.