

## Sistemi Embedded & IoT – Assignment #02

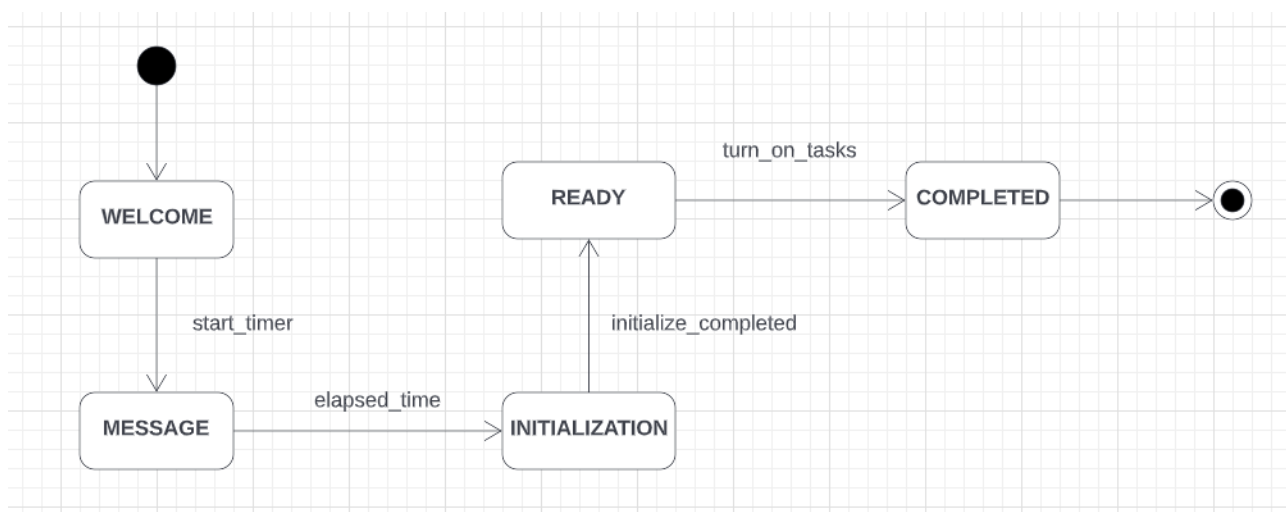
Progetto: smart coffee machine

### Arduino

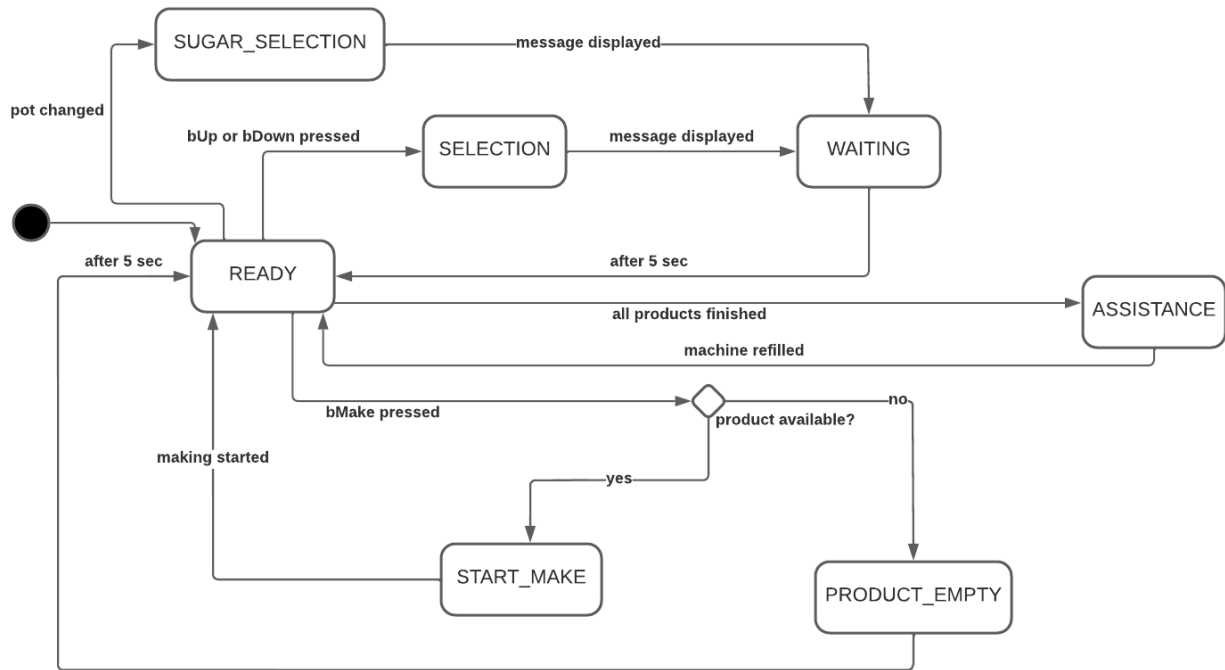
Nella cartella doc è presente il circuito progettato su Tinkercad (nota: il display fornito dal sito non è I2C, pertanto vi mancano i relativi collegamenti).

I task principali che vanno in esecuzione durante il programma sono 6:

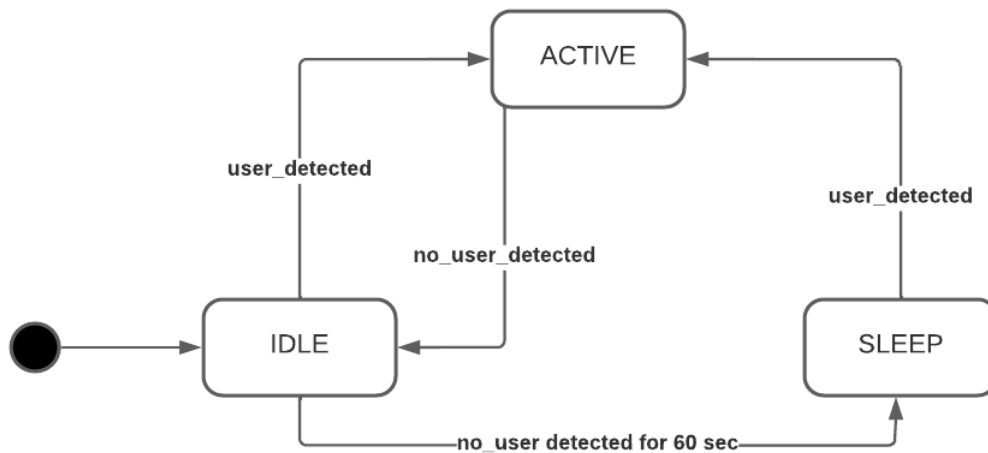
- **BootTask:** esegue le operazioni iniziali di avvio (come, ad esempio, la scritta del messaggio iniziale di benvenuto) e procede all'inizializzazione di tutte le bevande (Coffee, Tea, Chocolate). Una volta completato il task termina e manda in esecuzione *selectionTask*, *userPresenceTask* e *selfCheckTask*.



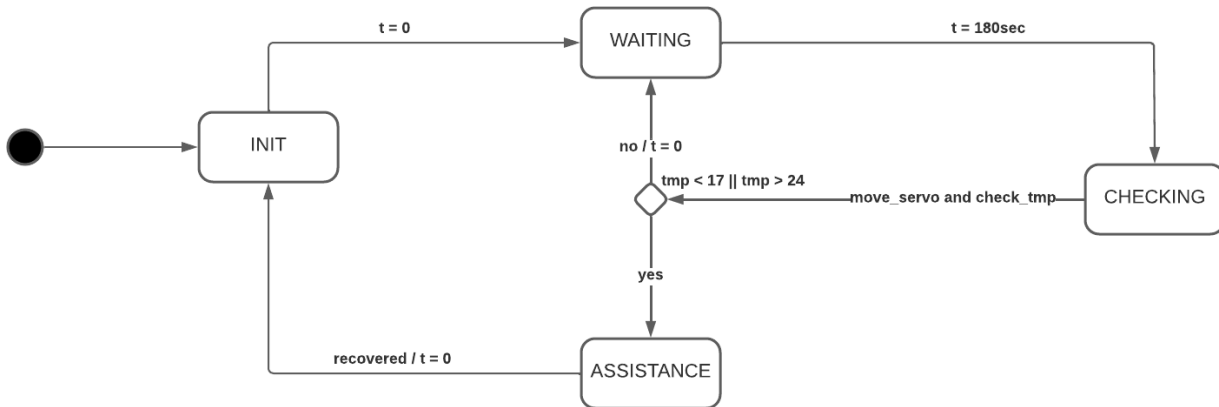
- **SelectionTask:** è il task che gestisce il menù di selezione della bevanda e del quantitativo di zucchero. Se la bevanda selezionata non è disponibile (ovvero ha quantità disponibile pari a 0) fornisce un messaggio di errore. Se nessuna bevanda risulta disponibile, il sistema viene mandato in *Assistenza* e potrà essere sbloccato solo dalla GUI che provvederà ad effettuare il “refill” delle bevande. Una volta premuto il pulsante di preparazione della bevanda, il task disabilita *userPresenceTask* e *selfCheckTask* (oltre a sé stesso) ed abilita *makingTask*.



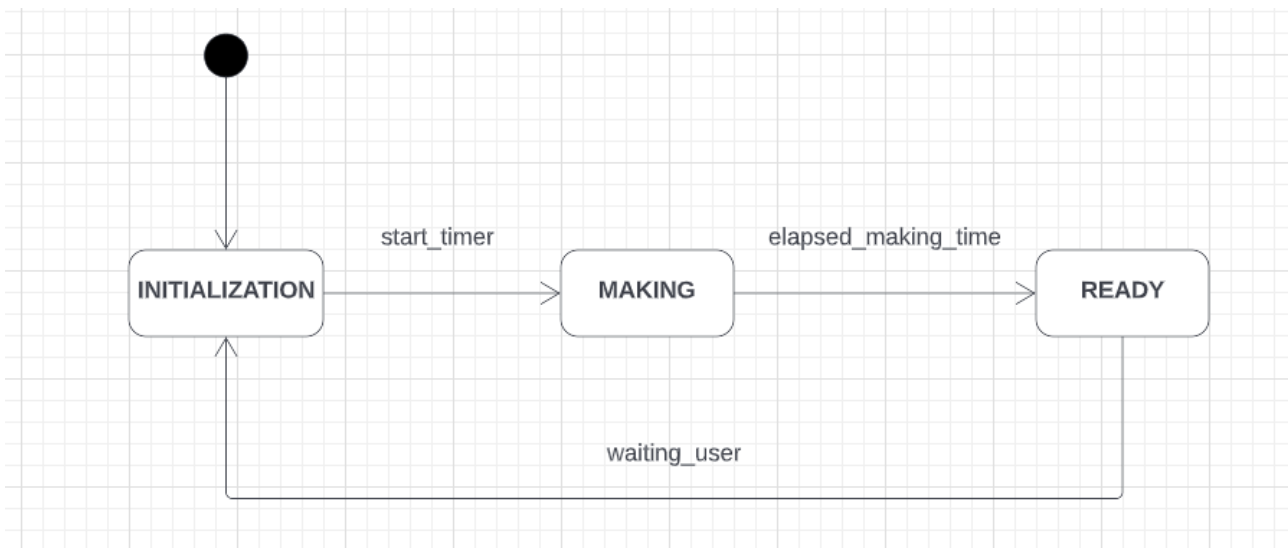
- **UserPresenceTask:** è il task su cui lavora il **Pir** per il rilevamento dell'utente. Dopo 60 secondi che nessun utente viene rilevato manda l'arduino in sleep, e potrà essere risvegliato solamente dal rilevamento di un nuovo utente.



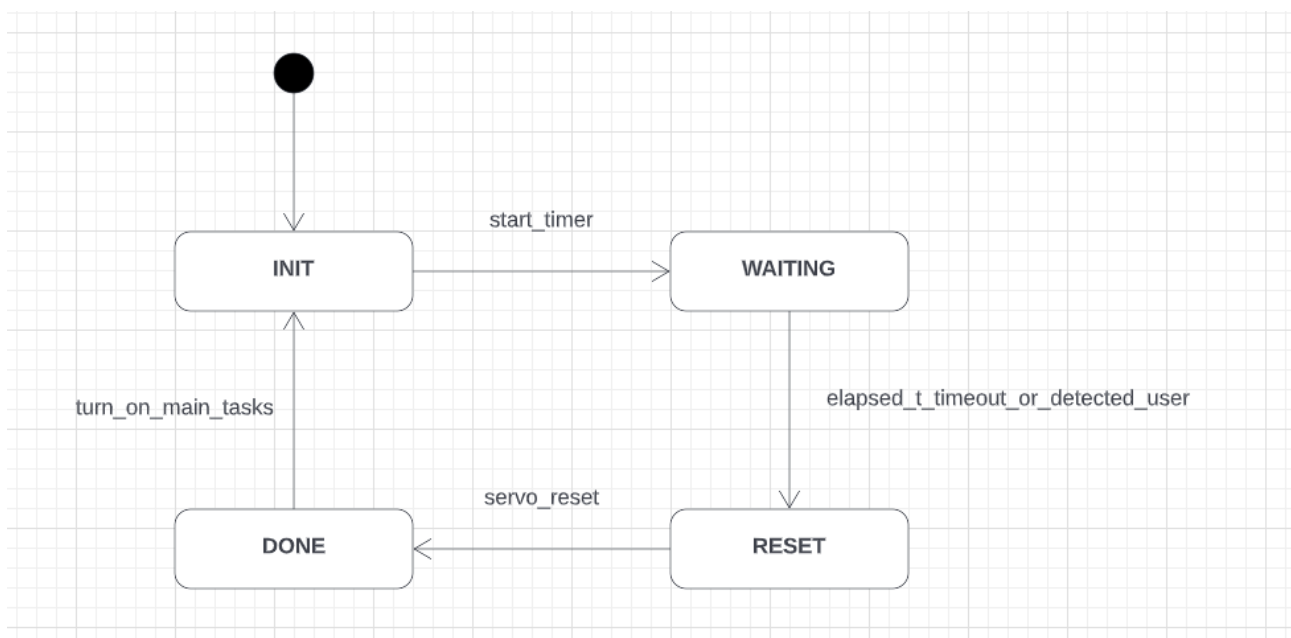
- **SelfCheckTask:** è il task che si occupa di eseguire i test diagnostici della macchina (ogni 180 sec). Quando è il momento fa muovere il servo avanti e indietro per 10 secondi e viene rilevata la temperatura dal sensore. Se è fuori dal range consentito (17-24 °C) il sistema viene mandato in *Assistenza* e potrà essere sbloccato solo dalla GUI.



- **MakingTask:** è il task che effettua la preparazione effettiva della bevanda selezionata. Provvede a muovere il Servo da 0 a 180 gradi in 10 secondi. Una volta completato, il task si disabilita e manda in esecuzione *waitingUserTask*.



- **WaitingUserTask:** è il task che attende che l'utente prelevi la bevanda. Controlla, tramite il sonar, se la distanza dell'utente dalla macchina è maggiore di 40 centimetri. In caso negativo, comunque dopo 5 secondi la bevanda viene considerata prelevata e il task si disabilita; a questo punto riabiliterà *selectionTask*, *userPresenceTask* e *selfCheckTask* e si potrà richiedere una nuova bevanda.



## Java

La GUI di smart coffee machine è stata progettata usando la libreria Swing. Al suo interno vi sono presenti due view:

- **LogView**, che fornisce un log dei messaggi che vengono ricevuti dall'applicazione stessa.
- La **GUI** vera e propria, che fornisce alcune informazioni, quali lo stato della macchina, la quantità di bevande disponibili per ogni tipo, oltre che a due pulsanti "Refill" e "Recover" per quando la macchina deve essere sbloccata perché si trova in stato di *Assistenza*.

La classe **MonitoringAgent**, infine, viene istanziato all'interno del **Controller** e gestisce i messaggi in ingresso e in uscita dalla seriale. I messaggi in ingresso vengono filtrati per capire quali informazioni trasportano, in modo da aggiornare la **GUI** correttamente. I messaggi in uscita devono notificare all'Arduino che la macchina è stata ripristinata correttamente e può uscire dalla modalità *Assistenza*.

