

# UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE ED ELETTRICA E  
MATEMATICA APPLICATA



Project work - Sistemi Embedded

## Induction Cooker

Deliverable 2

Membri del gruppo

Nome e cognome	Matricola	E-mail	Work hours
Adinolfi Teodoro	0622701902	t.adinolfi2@studenti.unisa.it	6
Amato Emilio	0622701903	e.amato16@studenti.unisa.it	6
Bove Antonio	0622701898	a.bove57@studenti.unisa.it	6

ANNO ACCADEMICO 2022/2023

---

## CAPITOLO 2

---

# MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

Questo capitolo si concentra sulla modellazione del sistema attraverso l'uso di Stateflow, uno strumento potente all'interno dell'ambiente di simulazione Simulink di MATLAB. Stateflow è particolarmente efficace nella gestione di logiche basate su stato, condizioni decisionali e comportamenti di sistema che cambiano nel tempo.

### 2.1 Button chart

In questa sezione discuteremo dell'implementazione del modulo *button*, utilizzato rispettivamente per pilotare l'accensione e lo spegnimento del sistema, e l'aumento e la diminuzione della potenza. Il modulo è stato reso flessibile per operare in una delle due modalità attraverso un segnale di input *mode* da fornire che, a seconda che sia 0 o 1, porta il bottone ad operare rispettivamente secondo la logica di *accensione e spegnimento del sistema* oppure *aumento e diminuzione della potenza*.

Questo perchè rispettivamente,

- quando si opera in modalità *accensione e spegnimento*, il sistema deve garantire che alla pressione del bottone si porti istantaneamente nello stato di accensione,

## 2. MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

---

mentre quando si trova nello stato acceso e deve essere spento, richiede una pressione prolungata pari ad 1 secondo

- invece, quando si opera in modalità *aumento e diminuzione della potenza*, è necessaria in entrambi i casi una pressione prolungata di 1 secondo per la commutazione

Dopo aver descritto la logica di base, proseguiamo con la presentazione del chart che illustra il funzionamento. In particolare, iniziamo con l'esporre il significato delle variabili utilizzate:

Tipo	Nome	Valore	Descrizione
input data	<i>b_in</i>		variabile di input relativa alla pressione del bottone
input data	<i>delay</i>		variabile di input relativa al delay impostato per la lettura
input data	<i>mode</i>		variabile di input relativa alla modalità operativa del bottone
output data	<i>b_short</i>		variabile di output per l'accensione del sistema
output data	<i>b_long</i>		variabile di output per lo spegnimento del sistema, l'aumento e la diminuzione della potenza
constant data	<i>PRESSED</i>	true	costante che indica la pressione del bottone
constant data	<i>RELEASED</i>	false	costante che indica il rilascio del bottone

Tabella 2.1: Tabella variabili button

Il modulo si divide in 5 stati principali:

- **wait\_pressed**
- **pressed**
- **wait\_longpressed**
- **longpressed**
- **wait\_released**

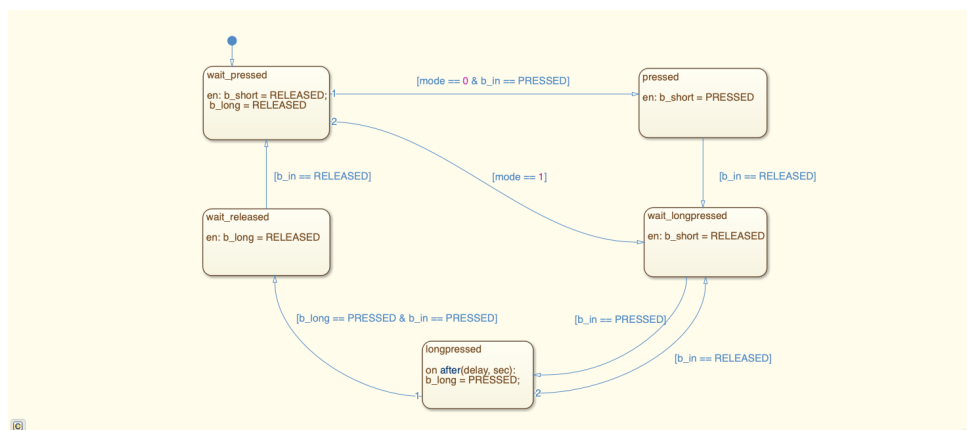


Figura 2.1: Button StateFlow diagram

A questo punto, iniziamo a descrivere il funzionamento del modulo:

1. esso parte dallo stato iniziale **wait\_pressed**, alla cui attivazione  $b\_short$  e  $b\_long$  vengono impostati a *RELEASED*, dopodichè alla pressione del bottone segnalata mediante la variabile di input  $b\_in$ , se la modalità impostata mediante  $mode$  è 0, la transizione farà confluire nello stato **pressed** in cui  $b\_short$  verrà impostato a *PRESSED* e permanendo in quest'ultimo fin quando il bottone non viene rilasciato, confluendo nello stato **wait\_longpressed**, che rappresenta l'attesa della pressione prolungata in cui vi si confluisce anche a partire dallo stato iniziale **wait\_pressed** se la modalità impostata è 1
2. dallo stato **wait\_longpressed**, che rappresenta l'attesa della pressione prolungata utilizzata in entrambe le modalità o per spegnere il sistema oppure per aumentare o diminuire la potenza, si attende la pressione del bottone segnalata mediante la variabile di input  $b\_in$  la quale permette di confluire nello stato **longpressed** in cui, se la pressione permane per il *delay* impostato, che nel sistema modellato rappresenta 1 secondo, fa sì che la variabile di uscita  $b\_long$  venga impostata a *PRESSED*. Se il bottone viene rilasciato prima del *delay* impostato, la transizione associata farà confluire dallo stato **longpressed** a **wait\_longpressed** in attesa di una nuova pressione, mentre se il bottone resta premuto nello stato **longpressed**, dopo 1 secondo la transizione in uscita farà confluire dallo stato corrente allo stato **wait\_released**, in cui la variabile di uscita  $b\_long$  viene messa a *RELEASED* in quanto già catturata la pressione del bottone e vi ci permane fin quando il bottone  $b\_in$  è *pressed*

3. infine, dallo stato **wait\_released**, solo quando il bottone viene rilasciato la variabile *b\_in* verrà impostata a *RELEASED* e farà confluire dallo stato corrente allo stato di **wait\_pressed**, in cui vi ci permane fino ad una prossima pressione se la modalità impostata è *accensione e spegnimento*, altrimenti sarà subito attivata la transizione verso lo stato **wait\_longpressed** per operare nell'altra modalità.

## 2.2 Blinking Led chart

In questa sezione discuteremo dell'implementazione del modulo *power\_led* introdotto per ottenere l'effetto di *blinking* necessario per il soddisfacimento delle specifiche sul led della potenza. Il modulo si divide in due stati principali:

- **off**
- **blink**

La macchina si trova inizialmente nello stato **off**, in questo contesto l'uscita del modulo sarà pari a zero, ciò corrisponde ad un led spento. Un segnale di abilitazione attiverà il lampeggio del led innescando una transizione verso il superstato **blink**. In questo caso si è scelta l'introduzione del superstato per far sì che il led possa essere spento in ogni momento, indipendentemente dal sottostato attivo in quel momento.

Quando la macchina si trova nello stato di **blink** una transizione farà sì che i due sottostati si alternino con un periodo fornito in ingresso al modulo. Analizziamo ora con maggior dettaglio il funzionamento di tali sottostati:

- **blink\_on**
- **blink\_off**

Il sottostato **blink\_on** è designato come sottostato iniziale del superstato **blink** e mantiene in uscita il segnale *out* ad alto, questo significa che il led sarà inizialmente acceso non appena ricevuto il segnale di abilitazione e si spegnerà soltanto dopo *period* millisecondi.

Il sottostato **blink\_off** è uno stato che possiamo definire di attesa in cui il valore dell'uscita è tenuto basso per *period* secondi in modo da poter realizzare l'effetto di lampeggio.

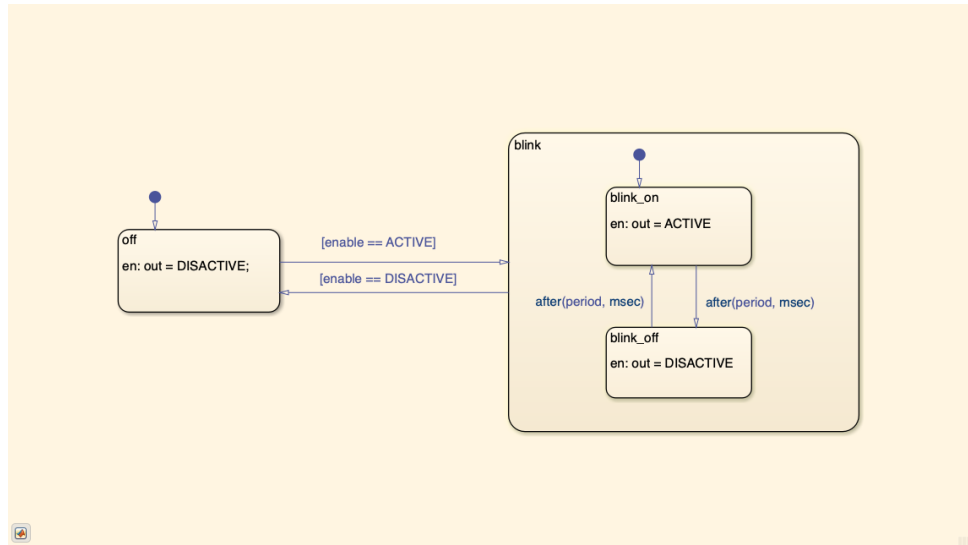


Figura 2.2: power\_led StateFlow diagram

Tipo	Nome	Valore	Descrizione
output data	<i>out</i>		variabile di output atta a pilotare il led
input data	<i>period</i>		variabile di input per impostare il periodo di lampeggio
input data	<i>enable</i>		variabile di input per abilitare il lampeggio
costant data	<i>ACTIVE</i>	1	costante per segnalare l'attivazione del led
costant data	<i>DISACTIVE</i>	0	costante per segnalare lo spegnimento del led

Tabella 2.2: Tabella variabili power\_led

### 2.3 Induction Cooker chart

Dopo aver descritto i componenti di base, proseguiamo con la presentazione del chart che illustra il cuore del sistema. In particolare, iniziamo con l'esporre il significato delle variabili utilizzate, come mostra la seguente tabella:

---

## 2. MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

---

Tipo	Nome	Valore	Descrizione
input data	<i>p0_short</i>		variabile di input per l'accensione del sistema
input data	<i>p0_long</i>		variabile di input per lo spegnimento del sistema
input data	<i>p1</i>		variabile di input per l'aumento della potenza
input data	<i>p2</i>		variabile di input per la diminuzione della potenza
input data	<i>pr</i>		variabile di input per il rilevamento della pentola
output data	<i>l1</i>		variabile di output per il led relativo all'accensione/spegnimento del sistema
output data	<i>l2</i>		variabile di output per il led relativo al livello di potenza in erogazione
output data	<i>l3</i>		variabile di output per il led relativo alla manifestazione dell'anomalia
output data	<i>blinking_led</i>		variabile di output relativa al periodo di blinking del led <i>l2</i>
local data	<i>err_status</i>		variabile locale relativa al manifestarsi dell'anomalia
constant data	<i>ON</i>	true	costante per l'attivazione dei led
constant data	<i>OFF</i>	false	costante per lo spegnimento dei led
constant data	<i>ERR</i>	true	costante che indica l'avvenuta anomalia

## 2. MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

---

constant data	<i>ANOMALY_DELAY</i>	10 (s)	costante che indica il tempo che deve trascorrere dalla rilevazione dell'anomalia affinché il sistema si riporti nello stato <i>inactive</i>
constant data	<i>POWER_DELAY</i>	5 (s)	costante che indica il tempo che deve trascorrere prima che sia effettivamente erogata la potenza a seguito di una sua variazione
constant data	<i>POT_IS_PRESENT</i>	true	costante che indica la presenza della pentola
constant data	<i>PRESSED</i>	true	costante che indica la pressione di uno dei pulsanti presenti
constant data	<i>DELAY_300</i>	2000 (ms)	costante che indica il periodo di blinking del led <i>l2</i> quando il sistema eroga 300W
constant data	<i>DELAY_500</i>	1000 (ms)	costante che indica il periodo di blinking del led <i>l2</i> quando il sistema eroga 500W
constant data	<i>DELAY_1000</i>	500 (ms)	costante che indica il periodo di blinking del led <i>l2</i> quando il sistema eroga 1000W
constant data	<i>DELAY_1500</i>	250 (ms)	costante che indica il periodo di blinking del led <i>l2</i> quando il sistema eroga 1500W

Tabella 2.3: Tabella variabili *induction cooker*

Proseguiamo quindi con la descrizione della struttura del chart, analizzando i diversi stati e le relative transizioni:

- **off**: è lo stato iniziale dell'intero sistema e in esso vengono inizializzati i tre led come spenti. Attraverso la transizione  $[p0\_short == PRESSED]$  è possibile spostarsi nello stato *inactive*



- **inactive**: in questo stato il sistema è attivo alla potenza di 0W e il led *l1* viene acceso. In particolare:
  - con la transizione  $[p0\_long == PRESSED]$  si ritorna nello stato *off*
  - con la transizione  $[p1 == PRESSED \ \& \ pr == POT\_IS\_PRESENT]$  si entra nello stato *power\_300w*, descritto in seguito
- **active**: è un superstato utilizzato per realizzare la decomposizione parallela (AND) tra gli stati *powers* e *anomaly\_guard*. In particolare:
  - con la transizione  $[p0\_long == PRESSED]$  si entra nello stato *off*, in maniera tale da poter spegnere il sistema indipendentemente dallo stato corrente in cui ci si trova
  - con la transizione  $[err\_status == ERR]$  si entra nello stato *inactive*, al fine di procedere con la disattivazione del fornello nel caso in cui l'anomalia si propaghi per 10 secondi
- **powers**: è un superstato che, quando attivo, indica la potenza che il fornello sta erogando. Al suo interno troviamo gli stati:
  - **power\_300w**: rappresenta la situazione in cui il fornello eroga 300W di potenza. In particolare, è anch'esso un superstato in cui troviamo lo stato **waiting\_300w** utilizzato come stato di attesa al fine del completamento del cambiamento di potenza trascorsi i 5 secondi e in combinazione con il fatto che la pentola sia presente (come indica la transizione  $after(Power\_DELAY, sec)[pr == POT\_IS\_PRESENT]$ ), e lo stato **working\_300w** che rappresenta l'effettivo cambio di potenza e all'interno del quale viene attivato il led *l2* con l'opportuno *blinking\_period*. Inoltre,
    - \* con la transizione  $[p2 == PRESSED \ \& \ pr == POT\_IS\_PRESENT]$  si entra nello stato *waiting\_inactive*
    - \* con la transizione  $[p1 == PRESSED \ \& \ pr == POT\_IS\_PRESENT]$  si passa nello stato *power\_500w* rappresentante la potenza successiva
  - **waiting\_inactive**: è uno stato di transizione che consente il passaggio allo stato *inactive* solo se sono trascorsi i 5 secondi necessari al cambio di

potenza, dal momento che anche 0W è considerata tale. Inoltre, qualora l'utente dovesse premere il pulsante di aumento della potenza (ovvero,  $[p1 == PRESSED \ \&\ pr == POT\_IS\_PRESENT]$ ) quando la macchina si trova in questo stato, si rientra nuovamente in *power\_300w*

- **working\_500w**: come lo stato *working\_300w*
- **working\_1000w**: come lo stato *working\_300w*
- **working\_1500w**: come lo stato *working\_300w*
- **anomaly\_guard**: è il superstato responsabile della gestione dell'anomalia, al cui interno troviamo due stati:
  - **guard**: rappresenta uno stato in cui la macchina controlla qualora si verificasse una rimozione della pentola, momento in cui attraverso la transizione  $[pr == \sim POT\_IS\_PRESENT]$  si entra nello stato *anomaly\_detected*
  - **anomaly\_detected**: l'ingresso in questo stato porta all'accensione del led *l3* che segnala l'anomalia e:
    - \* se la pentola viene riposizionata entro 10 secondi, si ritorna nello stato *guard*
    - \* altrimenti, si pone *err\_status = ERR* che fa scattare la transizione del superstato *active*, tornano nello stato *inactive* in cui il sistema è acceso alla potenza di 0W, come richiesto

## 2. MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

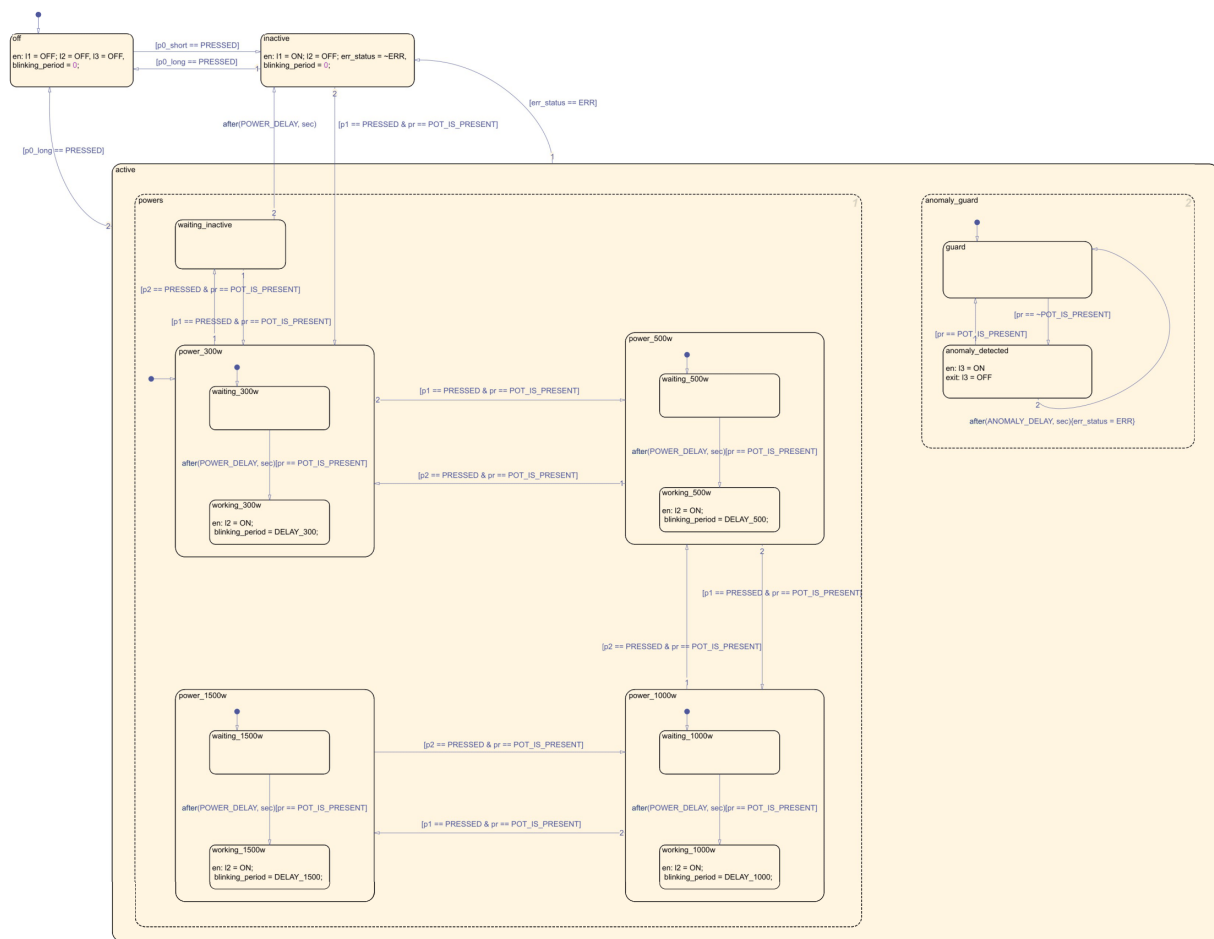


Figura 2.3: induction\_cooker StateFlow diagram