Università degli studi di Salerno
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE ED ELETTRICA E MATEMATICA APPLICATA

Sistemi Embedded

Project Work

A.A. 2024-2025

Group 21 members:

Senatore Annachiara	xxxxxxxxxx	a.senatore150@studenti.unisa.it
Infante Angelo	xxxxxxxxxx	a.infante32@studenti.unisa.it

Sommario

1. Introduzione 4
2. Progettazione 5
2.1. Risorse per lo sviluppo 5
2.2. Requisiti del Sistema 5
2.3. User Stories 8
2.3.1. US 1: Attivazione/Disattivazione Dispositivo 8
2.3.2. US 2: Configurazione del Dispositivo 8
2.3.3. US 3: Configurazione Veneziane 9
2.3.4. US 4: Configurazione Luci 9
2.3.5. US 5: Configurazione Temperatura 9
2.3.6. US 6: Gestione Pulsante Luci 9
2.3.7. US 7: Gestione Pulsante Luci Temporizzate 10
2.3.8. US 8: Settaggio Parametri 10
2.3.9. US 9: Gestione valori non validi 10
2.4. Use Cases 11
2.4.1. Use Case 1: L'utente vuole attivare il sistema 11
2.4.2. Use Case 2: Creazione prima Configurazione 12
2.4.3. Use Case 3: Modifica parametro di una Configurazione Esistente 13
2.4.4. Use Case 4: Gestione delle Luci tramite Pulsante LUCI 14
2.4.5. Use Case 5: Accensione Temporizzata Luci tramite Pulsante LUCI_TMP 15
2.4.6. Use case 6: Il sistema stampa periodicamente il suo stato 16
2.4.7. Use Case 7: Indicazione Visiva della Temperatura 16
2.4.8. Use case 8: Visualizzazione Angolo Veneziane 17
2.4.9. Use case 9: Visualizzazione Luci 18
2.5. Activity Diagrams 19
2.5.1. Attività: Gestione modalità configurazione 19
2.5.2. Attività: Avvio sistema e pressione pulsante ATTIVA 22
2.5.3. Attività: Pressione pulsante CONFIG 24
2.5.4. Attività: Pressione pulsante LUCI 26
2.5.5. Attività: Pressione pulsante LUCI TMP 28
3. Implementazione modello con Simulink/Stateflow 30
3.1. Descrizione del modello 30
3.3. Scelte progettuali ed implementative 30
3.4. Input/Output del modello 31
3.4.1. Input/Output con blocchi standard Simulink 31
3.4.2. Input/Output con blocchi della libreria per supporto STM32 33
3.5. Analisi delle funzioni MATLAB 34
3.6. Analisi delle variabili del modello 37
3.7. Analisi degli stati del modello 39
3.7.1. Inizializzazione 39
3.7.2. Stato_Attivo 40
3.7.3. Gestore_logica 41
3.7.4. Stato_Operativo 42
3.7.5. Stato_di_configurazione 43

3.7.6. Accenzione_luci 46
3.7.7. Configurazione_riscaldamento 48
3.7.8. Regolazione_veneziane 50
3.7.9. Led_utente 52
3.7.10. Gestione_Pulsanti_Luci 52
4. Configurazione hardware 55
4.1. Pin setup 55
4.3. Configurazione file .ioc 56
4.3.1. Clock Configuration 56
4.3.2. GPIO Configuration 57
4.3.3. Timer Configuration 57
4.3.4. UART Configuration 58
4.4. Overview configurazione circuito 59

1. Introduzione

Il presente progetto si propone di sviluppare il **firmware per un sistema integrato di domotica**, focalizzandosi sulla gestione intelligente di tre funzionalità chiave: la **regolazione delle veneziane**, l'**accensione delle luci** e il **controllo del riscaldamento**. L'obiettivo è realizzare un sistema affidabile e intuitivo che migliori l'esperienza utente.

Il dispositivo sarà caratterizzato dall'utilizzo di componenti hardware specifici, tra cui un **servomotore** per la regolazione delle veneziane, un **relè** per l'attivazione delle luci, un **LED RGB** per indicare la temperatura impostata e un **LED utente** per segnalare lo stato di attivazione. L'interazione con il sistema avverrà tramite quattro *pulsanti* dedicati: **ATTIVA** per abilitare o disabilitare il dispositivo, **CONFIG** per accedere alla modalità di configurazione, **LUCI** per il controllo diretto delle luci e **LUCI_TEMP** per il controllo della temporizzazione.

Il sistema prevederà una modalità di configurazione via **UART**, accessibile solo quando il dispositivo è attivo, che consentirà agli utenti di impostare i valori desiderati per ciascuna funzionalità. Particolare attenzione sarà posta alla gestione delle configurazioni iniziali e alla validazione dei dati inseriti, garantendo che il sistema operi sempre con parametri validi.

Le configurazioni includeranno l'angolo di apertura delle veneziane (da 0° a 180° con risoluzione di 20°), lo stato e la durata della temporizzazione delle luci (tra 5 e 30 secondi), e il range di temperatura (da 16° a 30° con risoluzione di 1°).

Il sistema fornirà feedback visivi e informativi agli utenti: il LED RGB indicherà la temperatura selezionata con diverse colorazioni (blu, viola, giallo, arancione, rosso), mentre il LED utente segnalerà lo stato di attività del dispositivo. In modalità operativa, inoltre, il sistema notificherà periodicamente il proprio stato tramite l'interfaccia UART, fornendo un riepilogo dei valori attualmente impostati per ogni funzionalità. La gestione delle luci sarà flessibile, consentendo l'accensione e lo spegnimento sia tramite configurazione UART che attraverso il pulsante LUCI, inclusa la possibilità di attivazione temporizzata (LUCI_TEMP).

2. Progettazione

Durante la fase di progettazione e modellazione del presente progetto didattico, sarà frequente l'utilizzo di terminologia legata all'apertura e chiusura delle veneziane, all'accensione e spegnimento delle luci, alla regolazione della temperatura e verranno utilizzati come interscambiabili i termini sistema e dispositivo (in questi casi per dispositivo si intenderà il sistema che viene eseguito sul dispositivo). In questo capitolo, verranno formalizzati i requisiti che il sistema dovrà avere e i casi d'uso in cui dovrà essere utilizzato, tramite anche l'utilizzo di diagrammi per facilitarne la comprensione.

2.1. Risorse per lo sviluppo

Le risorse utilizzate per lo sviluppo di questo sistema sono:

- MATLAB 2024 con Simulink e Stateflow per l'implementazione del modello.
- Librerie MATLAB esterne di supporto per lo sviluppo del sistema.
- Embedded Coder per la traduzione del modello in codice C, eseguibile dal microprocessore.
- Fritzing per la virtualizzazione del circuito per il testing del sistema sui componenti reali.
- Applicazione PuTTY per la simulazione di un terminale UART.
- Microprocessore ARM Cortex-M della tipologia nucleo STM32G474RE.
- Componenti hardware esterni per l'esecuzione del sistema:
    - Breadboard
    - Jumper
    - Led
    - Led RGB
    - Servomotore SG90
    - Relè JQC3F
    - Resistori
    - Pulsanti

2.2. Requisiti del Sistema

Il sistema deve consentire la gestione delle seguenti funzionalità:

- Regolazione apertura delle veneziane.
- Accensione luci.
- Riscaldamento.

I componenti hardware che verranno utilizzati saranno:

- Un servomotore, utilizzato per la regolazione dell'apertura delle veneziane.
- Un relè utilizzato per l'accensione e lo spegnimento delle luci.
- Un led RGB per indicare la temperatura del termostato.
- Un led utente per indicare lo stato di attivazione del dispositivo.
- Un pulsante per consentire entrata/uscita dalla modalità di configurazione (CONFIG).
- Un pulsante per attivare/disattivare il dispositivo (ATTIVA).
- Un pulsante per accendere/spegnere le luci (LUCI).
- Un pulsante per avviare l'accensione delle luci in modalità temporizzata (LUCI TMP).

Il sistema inoltre, tramite interfaccia UART, riesce a gestire gli input dell'utente e gli output che deve fornire.

Al primo avvio il sistema non dispone di alcuna configurazione valida per le varie funzionalità, di conseguenza l'utente, tramite interfaccia UART, deve essere obbligato ad inserirne una valida per

uscire dallo stato di configurazione. Mentre il dispositivo si trova in modalità configurazione, viene mostrato un menù all'utente per guidarlo nella configurazione delle funzionalità.

Le configurazioni valide per le 3 funzionalità sono:

- Veneziane: ○
    - 0° completamente chiuse ○
    - 180° completamente aperte
    - Valori da 0° a 180° selezionabili con la risoluzione di 20°
- Luci:
    - Accese
    - Spente
    - Durata dell'accensione temporizzata nel range [5 - 30] secondi, quando questa è attiva.
        - Una durata pari a 0 indica che la temporizzazione è inattiva.
- Temperatura:
    - Range 16° - 30° configurabile con risoluzione di 1°

Inserire valori errati comporta il rigetto della configurazione. Il sistema dovrà essere in grado anche di, in caso di rigetto delle configurazioni, utilizzare, se presente, l'ultima configurazione considerata valida.

Quando il dispositivo è attivo e non è in modalità configurazione deve mostrare, tramite interfaccia UART, il proprio stato corrente relativo alle sue funzionalità.

All'accensione il dispositivo non è attivo. Quando viene attivato tramite pulsante, il led utente deve lampeggiare ad una frequenza di 2Hz e per le varie funzionalità vengono impostati i valori di configurazione. Quando è inattivo invece vengono settati i seguenti valori di default:

- Veneziane chiuse.
- Luci spente. / Temporizzazione inattiva.
- Temperatura al minimo.

Il dispositivo fornisce un feedback circa la temperatura selezionata tramite il led RGB in accordo alla seguente semantica:

- Da 16 a 18: colore blu.
- Da 19 a 21: colore viola.
- Da 22 a 24: colore giallo.
- Da 25 a 27: colore arancione.
- Da 28 a 30: colore rosso.

Ora di seguito verranno gestite le casistiche relative alla pressione dei pulsanti LUCI e LUCI TMP.

- Pressione pulsante LUCI:
    - Se le luci sono spente, vengono accese.
    - Se le luci sono accese, vengono spente.
    - Se le luci sono accese con l'accensione temporizzata, vengono spente in anticipo.
- Pressione pulsante LUCI TMP:

- Se le luci sono spente, vengono accese per il periodo impostato e poi spente.
- Se le luci sono accese in modalità non temporizzata, rimangono accese per il periodo imposto e poi spente.
- Se le luci sono già state avviate in accensione temporizzata, la pressione non ha alcun effetto.

2.3. User Stories

Partendo dai requisiti del sistema, verranno ora definite delle User Stories, in accordo con i principi SOLID e seguendo gli standard per la loro scrittura, al fine di formalizzare il lavoro per lo sviluppo del firmware e rendere più semplice la divisione dei compiti tra il team di sviluppo. Esse sono utili anche per stabilire se una funzionalità è stata implementata correttamente, tramite il controllo dei criteri di accettazione (opportunamente formalizzanti secondo gli standard) e per tenere traccia del lavoro svolto.

2.3.1. US 1: Attivazione/Disattivazione Dispositivo

Come utente,
voglio poter attivare e disattivare il dispositivo,
in modo da controllarne il funzionamento generale.

Criteri di accettazione

- Dato il dispositivo non attivo, quando l'utente preme il pulsante ATTIVA, il dispositivo diventa attivo.
- Dato il dispositivo attivo, quando l'utente preme il pulsante ATTIVA, il dispositivo diventa non attivo
- Quando il dispositivo è attivo, il LED utente lampeggia.
- Quando il dispositivo è inattivo, il LED utente è spento.
- Quando il dispositivo è inattivo, le veneziane sono chiuse (0°), le luci sono spente e la temporizzazione è inattiva e la temperatura è impostata al minimo (16°).
- Dato il dispositivo non attivo, quando i pulsanti, eccetto il pulsante ATTIVA, vengono premuti, la pressione viene ignorata.

2.3.2. US 2: Configurazione del Dispositivo

Come utente,
voglio poter entrare in modalità configurazione,
in modo da creare una configurazione valida.

Criteri di accettazione

- Dato il dispositivo non attivo, quando viene attivato e non è presente una configurazione valida (solo prima attivazione), l'utente è obbligato ad inserire una configurazione valida.
- Dato il dispositivo attivo, quando l'utente preme il pulsante CONFIG, il dispositivo entra/esce dalla modalità di configurazione.
- Quando il dispositivo si trova in modalità configurazione, trasmette un menù tramite interfaccia UART per selezionare le funzionalità da configurare.
- Dato il dispositivo in stato di configurazione, quando l'utente inserisce una configurazione non valida, essa viene rigettata dal sistema e viene impostata l'ultima configurazione considerata valida.
- Dato il dispositivo in stato di configurazione, quando l'utente inserisce una configurazione valida per tutte le funzionalità e salva, il dispositivo esce dallo stato di configurazione e imposta la configurazione salvata.

2.3.3. US 3: Configurazione Veneziane

Come utente,
voglio poter regolare l'apertura delle veneziane tramite interfaccia,
in modo da impostare il valore desiderato.

Criteri di accettazione

- Dato il dispositivo in modalità configurazione, quando l'utente inserisce un valore valido per l'apertura/la chiusura delle veneziane tramite UART, il valore viene salvato.
- Dato il dispositivo in modalità configurazione, quando l'utente inserisce un valore non valido per l'apertura/la chiusura delle veneziane tramite UART, il valore non viene salvato.

2.3.4. US 4: Configurazione Luci

Come utente,
voglio poter configurare le luci tramite interfaccia,
in modo da gestirne l'accensione.

Criteri di accettazione

- Dato il dispositivo in modalità configurazione, quando l'utente seleziona il valore per accendere/spegnere le luci tramite UART, il valore viene salvato.
- Dato il dispositivo in modalità configurazione, quando l'utente seleziona il valore per temporizzare l'accensione delle luci tramite UART, il valore viene salvato.
- Dato il dispositivo in modalità configurazione, quando l'utente seleziona un valore non valido per accendere/spegnere le luci tramite UART, il valore non viene salvato.
- Dato il dispositivo in modalità configurazione, quando l'utente seleziona un valore non valore per temporizzare l'accensione delle luci tramite UART, il valore non viene salvato.

2.3.5. US 5: Configurazione Temperatura

Come utente,
voglio poter configurare la temperatura tramite interfaccia,
in modo da gestirne l'accensione.

Criteri di accettazione

- Dato il dispositivo in modalità configurazione, quando l'utente inserisce un valore valido per il termostato tramite UART, il valore viene salvato.
- Dato il dispositivo in modalità configurazione, quando l'utente inserisce un valore non valido per il termostato tramite UART, il valore non viene salvato.

2.3.6. US 6: Gestione Pulsante Luci

Come utente,
voglio poter accendere e spegnere le luci manualmente tramite pulsante,
in modo da gestire le luci anche al di fuori dello stato di configurazione.

Criteri di accettazione

- Dato il dispositivo attivo e con almeno una configurazione valida, quando le luci sono spente e l'utente preme il pulsante LUCI, le luci vengono accese.
- Dato il dispositivo attivo e con almeno una configurazione valida, quando le luci sono accese e l'utente preme il pulsante LUCI, le luci vengono spente.
- Dato il dispositivo attivo e con almeno una configurazione valida, quando le luci sono accese in modalità temporizzata e l'utente preme il pulsante LUCI, le luci vengono spente in anticipo.

2.3.7. US 7: Gestione Pulsante Luci Temporizzate

Come utente,
voglio poter accendere le luci in modalità temporizzata le luci manualmente tramite pulsante,
in modo da gestire la temporizzazione delle luci.

Criteri di accettazione

- Dato il dispositivo attivo e con almeno una configurazione valida, quando le luci sono spente e l'utente preme il pulsante LUCI_TMP, le luci vengono accese per il periodo impostato e poi spente.
- Dato il dispositivo attivo e con almeno una configurazione valida, quando le luci sono accese in modalità non temporizzata e l'utente preme il pulsante LUCI_TMP, le luci rimangono accese per il periodo impostato e poi spente.
- Dato il dispositivo attivo e con almeno una configurazione valida, quando le luci sono accese in modalità temporizzata e l'utente preme il pulsante LUCI_TMP, la pressione non ha effetto.

2.3.8. US 8: Settaggio Parametri

Come utente,
voglio settare i parametri per le varie funzionalità,
in modo da personalizzare il funzionamento del sistema.

Criteri di accettazione

- Dato il dispositivo in modalità configurazione e con una configurazione valida appena inserita, quando l'utente salva, vengono settati i parametri della configurazione appena salvata.
- Dato il dispositivo in modalità configurazione e con una configurazione valida precedentemente inserita, quando l'utente esce, vengono settati i parametri della configurazione valida precedente.
- Quando il dispositivo è attivo e non in stato di configurazione, viene mostrato sull'interfaccia UART il suo stato corrente.

2.3.9. US 9: Gestione valori non validi

Come utente,
voglio essere informato quando inserisco valori non validi,
in modo da potermi correggere.

Criteri di accettazione

- Dato il dispositivo attivo e in modalità di configurazione, quando l'utente inserisce valori non validi per le varie funzionalità, essi vengono rigettati e viene mostrato un messaggio di errore sull'interfaccia UART.

2.4. Use Cases

Di seguito verranno presentati i principali casi d'uso di interesse per il sistema.

2.4.1. Use Case 1: L'utente vuole attivare il sistema

Attori

- Utente.
- Sistema.

Descrizione

- L'utente preme il pulsante ATTIVA per attivare il sistema quando esso è in stato non attivo.

Precondizioni

- Il sistema è non attivo.
- Il pulsante ATTIVA viene premuto.

Postcondizioni

- Il sistema passa allo stato attivo.
- Il LED utente lampeggia.
- Viene caricata l'ultima configurazione valida del sistema.

Flusso Principale

1. L'utente preme il pulsante ATTIVA.
2. Il sistema verifica che il dispositivo è non attivo e passa allo stato attivo.
3. Il LED utente inizia a lampeggiare.
4. L'ultima configurazione valida viene caricata.

Scenari Alternativi

- Se il pulsante ATTIVA viene ripremuto, il sistema passa allo stato non attivo.
- Se il pulsante ATTIVA viene premuto ed è la prima attivazione del sistema e non è presente una configurazione valida, il sistema entra in modalità configurazione (e non nello stato operativo).

2.4.2. Use Case 2: Creazione prima Configurazione

Attori

- Utente.
- Sistema.

Descrizione

- Al primo avvio l'utente preme il pulsante ATTIVA e attraverso il menù contestuale crea la prima configurazione valida.

Precondizioni

- Il sistema è non attivo.
- Il sistema non dispone di alcuna configurazione valida.
- Il pulsante ATTIVA viene premuto.
- L'utente è in grado di interagire con l'interfaccia UART.

Postcondizioni

- Il sistema è attivo.
- Viene creata la prima configurazione valida.

Flusso Principale

1. L'utente preme il pulsante ATTIVA per la prima volta.
2. Il sistema entra in modalità configurazione e visualizza il menù di configurazione tramite UART.
3. L'utente seleziona la funzionalità da configurare.
4. Per ogni parametro da configurare che l'utente seleziona, viene mostrato il range/ la risoluzione di validità da rispettare.
5. Per ogni parametro da configurare l'utente inserisce un valore valido per la configurazione.
6. L'utente configura tutti i parametri.
7. L'utente salva la configurazione.
8. La prima configurazione valida viene caricata.

Scenari Alternativi

- Se il pulsante CONFIG viene premuto ma non è presente alcuna configurazione valida, l'uscita viene impedita finché non viene salvata almeno una configurazione valida per tutte le funzionalità.
- Se l'utente inserisce dei valori non validi, il sistema informa l'utente dell'errore e torna al menù contestuale iniziale per consentirgli di riprovare.
- Se l'utente preme CONFIG mentre il dispositivo è inattivo, il comando viene ignorato e il sistema non entra in modalità configurazione.

2.4.3. Use Case 3: Modifica parametro di una Configurazione Esistente

Attori

- Utente.
- Sistema.

Descrizione

- L’utente dopo aver salvato una configurazione valida preme il tasto CONFIG per modificare uno o più parametri di essa.

Precondizioni

- Il sistema è attivo.
- L’utente ha già inserito una configurazione valida.
- L’utente ha già salvato una configurazione valida.
- Il pulsante CONFIG viene premuto.

Postcondizioni

- Viene creata una nuova configurazione valida.

Flusso Principale

1. L’utente preme il pulsante CONFIG.
2. Il menù di configurazione viene visualizzato tramite UART, mostrando le opzioni disponibili.
3. L'utente seleziona il parametro da modificare.
4. Il sistema chiede l'inserimento del nuovo valore desiderato, specificando il range/la risoluzione validi.
5. L'utente inserisce il nuovo valore valido.
6. Il menù di configurazione viene ripresentato, permettendo all'utente di modificare un altro parametro o di procedere al salvataggio.
7. L'utente seleziona l'opzione "Salva”.
8. Il sistema esegue il salvataggio della configurazione appena definita che diventa la nuova configurazione valida.

Scenari Alternativi

- Se l’utente inserisce dei valori non validi, la modifica viene scartata ed il valore del parametro rimane quello precedente; il sistema notifica l'errore e permette all'utente di ritentare o scegliere un'altra opzione.
- Se il pulsante CONFIG viene premuto nuovamente senza aver effettuato il salvataggio dei parametri modificati, il sistema esce dallo stato di configurazione e imposta l’ultima configurazione valida precedentemente salvata.

2.4.4. Use Case 4: Gestione delle Luci tramite Pulsante LUCI

Attori

- Utente.
- Sistema.

Descrizione

- L'utente preme il pulsante LUCI per controllare lo stato di accensione/spegnimento delle luci o per interrompere un'accensione temporizzata in corso.

Precondizioni

- Il sistema è in stato "Attivo".
- Esiste almeno una configurazione valida salvata.
- Il pulsante LUCI viene premuto.

Postcondizioni

- Lo stato delle luci viene modificato.
- La modifica viene mostrata all'utente.

Flusso Principale

1. L'utente preme il pulsante LUCI.
2. Il sistema rileva la pressione del pulsante.
3. Il sistema determina lo stato attuale delle luci e agisce come segue:
    a) Se le luci sono spente: le luci vengono accese (modalità normale).
    b) Se le luci sono accese in modalità non temporizzata, le luci vengono spente.
    c) Se le luci sono accese in modalità temporizzata, le luci vengono spente in anticipo.

Scenari Alternativi

- Se il pulsante LUCI viene premuto quando il sistema è inattivo o quando non è stata ancora salvata nessuna configurazione valida la pressione del pulsante viene ignorata.
- Se il pulsante LUCI viene premuto quando il sistema è in modalità configurazione ed è già stata salvata almeno una configurazione valida, il sistema si comporta secondo il flusso principale (punto 3.) accendendo/spegnendo le luci in base al loro stato.
- Se il sistema passa in modalità non attiva (il pulsante ATTIVA viene premuto) le luci si spengono.

2.4.5. Use Case 5: Accensione Temporizzata Luci tramite Pulsante LUCI_TMP

Attori

- Utente.
- Sistema.

Descrizione

- L'utente preme il pulsante LUCI_TMP per avviare un ciclo di accensione delle luci con una durata predefinita configurata, permettendo un controllo temporizzato dell'illuminazione.

Precondizioni

- Il sistema è in stato "Attivo".
- La temporizzazione luci è attiva.
- È stata impostata una durata di temporizzazione delle luci valida, diversa da 0.
- Il pulsante LUCI_TMP viene premuto.

Postcondizioni

- Le luci vengono accese per la durata configurata.
- Dopo la durata configurata, le luci si spengono automaticamente.

Flusso Principale

1. L'utente preme il pulsante LUCI_TMP.
2. Il sistema rileva la pressione del pulsante.
3. Il sistema determina lo stato attuale delle luci e agisce come segue:
   a) SE le luci sono attualmente spente (e non in un ciclo di temporizzazione), le luci vengono accese e restano accese per la durata di temporizzazione configurata.
   b) SE le luci sono attualmente accese in modalità non temporizzata (ovvero "sempre accese"), le luci rimangono accese ma al termine della durata di temporizzazione settata in fase di configurazione vengono spente automaticamente.
   c) SE le luci sono già in un ciclo di accensione temporizzata (già avviate da LUCI_TMP), la pressione del pulsante LUCI_TMP viene ignorata e non ha alcun effetto. Il ciclo di temporizzazione in corso continua indisturbato.

Scenari Alternativi

- Se Il pulsante LUCI_TMP viene premuto quando la temporizzazione è non attiva (L'utente inserisce 0 in fase di configurazione della temporizzazione), il sistema ignora la pressione del pulsante LUCI_TMP e non avvia alcun ciclo di accensione temporizzata.
- Se il pulsante LUCI_TMP viene premuto quando il sistema è inattivo o non esiste nessuna configurazione valida salvata, la pressione del pulsante viene ignorata.
- Se il pulsante LUCI_TMP viene premuto durante la fase di configurazione, ma esiste una configurazione valida salvata, il sistema segue il flusso principale.
- Se il sistema passa in modalità non attiva (il pulsante ATTIVA viene premuto) le luci si spengono.

2.4.6. Use case 6: Il sistema stampa periodicamente il suo stato

Attori

- Sistema.

Descrizione

- Il sistema fornisce autonomamente e periodicamente all'utente un riepilogo dello stato attuale dei parametri configurati (angolo veneziane, stato luci, temperatura) tramite l'interfaccia UART.

Precondizioni

- Il sistema è in stato "Attivo".
- È stata salvata almeno una configurazione.
- Il sistema non è in modalità di configurazione.
- È trascorso il periodo di tempo predefinito per la notifica (ogni 10 secondi).

Postcondizioni

- Il sistema mostra un messaggio di stato, contenente i valori correnti di veneziane, luci e temperatura.
- Il timer per la prossima notifica di stato è riavviato.

Flusso Principale

1. Il sistema entra o si trova nello stato "Attivo" e non è in modalità di configurazione.
2. Il timer per la notifica dello stato viene avviato.
3. All'ingresso in questa condizione o allo scadere di un timer precedente, il sistema acquisisce i valori correnti dei parametri.
4. Il sistema stampa un messaggio di stato coerente con i valori correnti dei parametri.
5. Il sistema attende 10 secondi.
6. Trascorsi i 10 secondi, il flusso riprende dal passo 2, continuando a inviare aggiornamenti periodici.

Scenari Alternativi

- Se il sistema passa in modalità di configurazione o diventa inattivo, la trasmissione periodica dello stato viene interrotta. Riprenderà solo quando il sistema tornerà in stato "Attivo" e non in modalità configurazione.

2.4.7. Use Case 7: Indicazione Visiva della Temperatura

Attori

- Sistema.
- Utente.

Descrizione

- Il sistema fornisce un'indicazione visiva immediata del range di temperatura attualmente impostato per il riscaldamento.

Precondizioni

- Il sistema è in stato "Attivo".
- È stata salvata almeno una configurazione.

Postcondizioni

- Il LED si illumina con il colore corrispondente al range di temperatura della configurazione attiva.

Flusso Principale

1. L'utente inserisce e salva una configurazione valida.
2. Il sistema verifica il valore numerico della temperatura impostata.
3. Il sistema agisce come segue:
   a) Se la temperatura è impostata tra 16° e 18°, il LED RGB si illumina di colore blu.
   b) Se la temperatura è impostata tra 19° e 21, il LED RGB si illumina di colore viola.
   c) Se la temperatura è impostata tra 22° e 24°, il LED RGB si illumina di colore giallo.
   d) Se la temperatura è impostata tra 25° e 27°, il LED RGB si illumina di colore arancione.
   e) Se la temperatura è impostata tra 28° e 30°, il LED RGB si illumina di colore rosso.
4. Questo processo si ripete ogni volta che il valore della temperatura viene modificato e viene saltava una nuova configurazione valida.

Scenari Alternativi

- Se il sistema passa in modalità non attiva (il pulsante ATTIVA viene premuto) l'indicatore di temperatura si illumina di colore BLU (come definito in fase di progettazione nel caso di sistema non attivo).

2.4.8. Use case 8: Visualizzazione Angolo Veneziane

Attori

- Sistema.
- Utente.

Descrizione

- Il sistema visualizza l'angolo di apertura delle veneziane impostato nella configurazione salvata.

Precondizioni

- Il sistema è in stato "Attivo".
- È stata salvata almeno una configurazione.

Postcondizioni

- L’utente visualizza il valore dell’angolo delle veneziane impostato nella configurazione salvata.

Flusso Principale

1. L’utente inserisce e salva una configurazione valida.
2. Il sistema verifica il valore dell’angolo delle veneziane impostato nella configurazione salvata.
3. Il sistema imposta il valore del nuovo angolo sul servomotore corrispondente alle veneziane.
4. Questo processo si ripete ogni volta che il valore delle veneziane viene modificato e viene salvata una nuova configurazione valida.

Scenari Alternativi

- Se il sistema passa in modalità non attiva (il pulsante ATTIVA viene premuto), le veneziane vengono chiuse (angolo 0°).

2.4.9. Use case 9: Visualizzazione Luci

Attori

- Sistema.
- Utente.

Descrizione

- Il sistema mostra lo stato attuale di accensione/spegnimento delle luci.

Precondizioni

- Il sistema è in stato "Attivo".
- È stata salvata almeno una configurazione.

Postcondizioni

- L’utente visualizza lo stato delle luci impostato nella configurazione salvata.

Flusso Principale

1) L’utente inserisce e salva una configurazione valida.
2) Il sistema verifica il valore del parametro luci.
3) Il sistema agisce come segue:
   a) Se il parametro luci è impostato su “Luci accese”, il sistema accende i led corrispondenti alle luci.
   b) Se il parametro luci è impostato su “Luci spente”, il sistema spegne i led corrispondenti alle luci.

4) Questo processo si ripete ogni volta che il valore delle luci viene modificato in fase di configurazione e viene salvata una nuova configurazione valida.

Scenari Alternativi

- Se il pulsante LUCI viene premuto, il sistema si comporta come spiegato nello Use Case 4.
- Se il pulsante LUCI_TMP viene premuto, il sistema si comporta come spiegato nello Use Case 5.
- Se il sistema passa in modalità non attiva (il pulsante ATTIVA viene premuto), le luci vengono spente.

2.5. Activity Diagrams

Gli Activity diagrams sono dei diagrammi utilizzati per mostrare, in maniera più semplificata, il flusso di operazioni all'interno del sistema, in relazione a degli scenari tipici. Per la loro realizzazione sono stati utilizzati elementi derivati dalla sintassi UML, in particolare l'inizio di un'attività è indicato tramite l'utilizzo di un cerchio pieno, mentre la fine tramite l'utilizzo di due cerchi concentrici, di cui quello interno è pieno.
Sono state considerate cinque attività principali da analizzare (escludendo casi particolari):

- Gestione modalità configurazione.
- Pressione del pulsante ATTIVA.
- Pressione del pulsante CONFIG.
- Pressione del pulsante LUCI.
- Pressione del pulsante LUCI TMP.

2.5.1. Attività: Gestione modalità configurazione

Questo diagramma descrive il flusso di operazioni per la gestione dello stato di configurazione del sistema. Si evolve seguendo le interazioni dell'utente, andando a gestire tutti i vari casi possibili, sia di errore sia di successo. Inoltre, per giungere allo stato finale, deve essere presente, come da requisito, una configurazione valida.

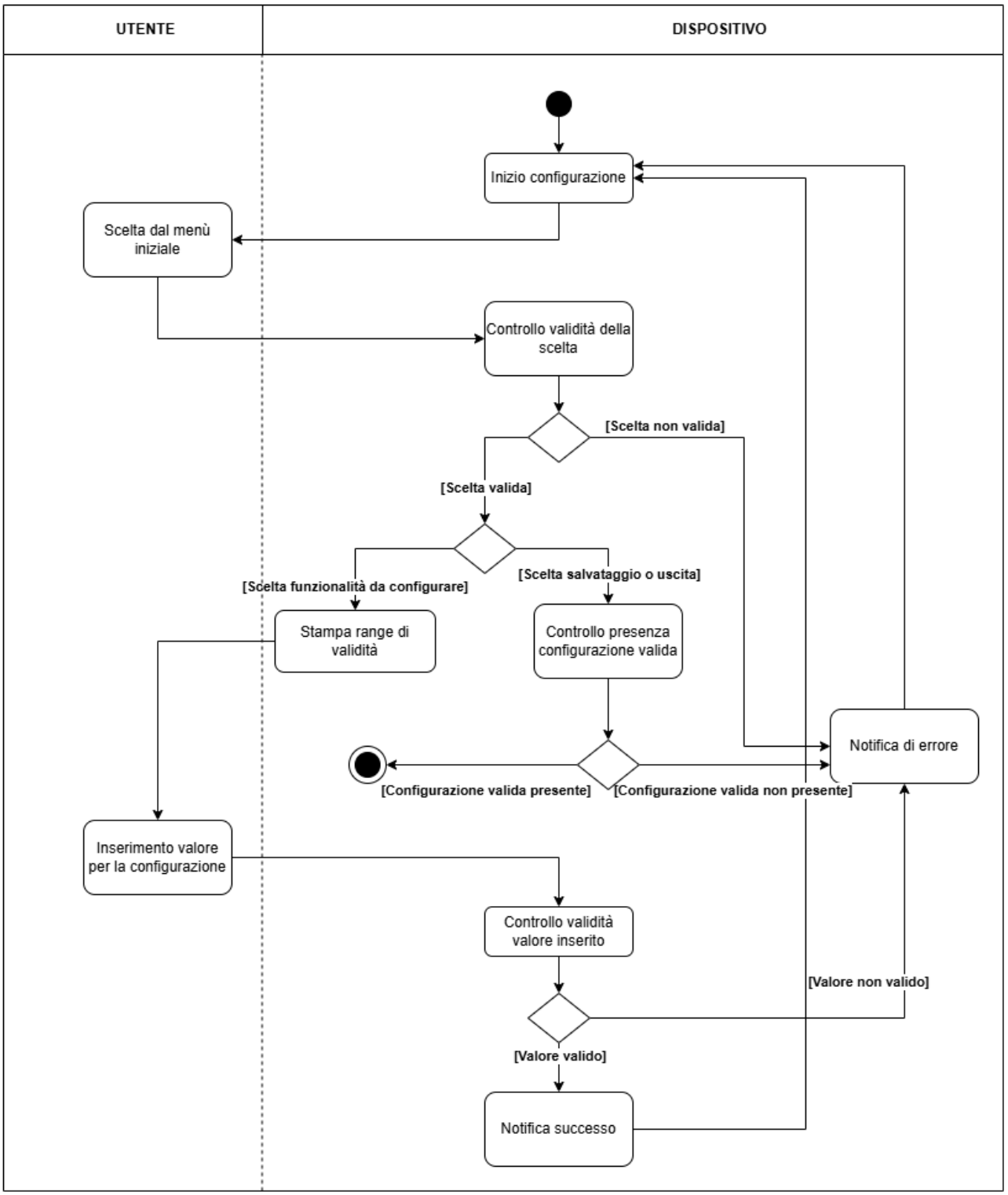
Descrizione flusso:

1) Inizio
   a. Il sistema entra in modalità configurazione
2) Inizio configurazione
   a. Il sistema mostra, tramite interfaccia UART, un menù dal quale l'utente può selezionare l'operazione che vuole effettuare.
3) Scelta dal menù iniziale
   a. L'utente effettua la scelta dal menù mostrato.
4) Controllo validità della scelta
   a. Il sistema controlla se la scelta fatta dall'utente è valida e di che tipo è.
   b. Se la scelta è valida ed è relativa ad una funzionalità da configurare allora il sistema mostra il range di validità per quella funzionalità.
   c. Se la scelta è valida ed è relativa all'uscita dalla modalità o al salvataggio di una configurazione allora il sistema controlla la presenza di una configurazione valida.
   d. Se la scelta non è valida viene mostrato un messaggio di errore dal sistema.
5) Stampa range di validità.

a. Il sistema mostra il range dei valori che considera validi per la funzionalità scelta dall'utente per la configurazione. I diversi range sono specificati nei requisiti.

6) Inserimento valore per la configurazione
   - L'utente inserisce il valore che vuole impostare per la funzionalità scelta.
7) Controllo validità valore inserito
   - Il sistema controlla se il valore inserito dall'utente rispetta il range per essere considerato valido.
   - Se il valore è valido allora il sistema mostra un messaggio per notificare il successo dell'operazione.
   - Se il valore non è valido il sistema mostra un messaggio per notificare il fallimento dell'operazione.
8) Notifica successo
   - Il sistema, tramite interfaccia UART, notifica all'utente che l'operazione da lui effettuata è conclusa con successo.
9) Controllo presenza configurazione valida
   i) Il sistema controlla se è presente una configurazione valida per ogni funzionalità del sistema.
   ii) Se è presente una configurazione valida per ogni funzionalità del sistema e se l'utente ha scelto di uscire allora il sistema imposta la configurazione valida precedente all'inizio della configurazione attuale, se invece ha scelto di salvare allora il sistema imposta i nuovi parametri impostati dall'utente. Subito dopo esce dalla modalità di configurazione.
   iii) Se non è presente una configurazione valida per ogni funzionalità del sistema allora l'utente viene notificato dal sistema, tramite un messaggio di errore, e resta nella modalità di configurazione finché non imposta una configurazione valida. Questo è il caso relativo al primo avvio del sistema.
10) Notifica di errore
    a) Il sistema, tramite interfaccia UART, notifica all'utente che l'operazione da lui effettuata non è andata a buon fine.
    b) Dopo la notifica, si ricomincia il processo di configurazione dalla visualizzazione del menù per le scelte.
11) Fine
    a) Il sistema esce dalla modalità di configurazione ed entra nella modalità operativa, dove imposta i parametri configurati per tutti i dispositivi output delle relative funzionalità.

Diagramma:


UTENTE
DISPOSITIVO
Inizio configurazione
Scelta dal menù iniziale
Controllo validità della scelta
[Scelta non valida]
[Scelta valida]
[Scelta funzionalità da configurare]
[Scelta salvataggio o uscita]
Stampa range di validità
Controllo presenza configurazione valida
Notifica di errore
[Configurazione valida presente]
[Configurazione valida non presente]
Inserimento valore per la configurazione
Controllo validità valore inserito
[Valore non valido]
[Valore valido]
Notifica successo

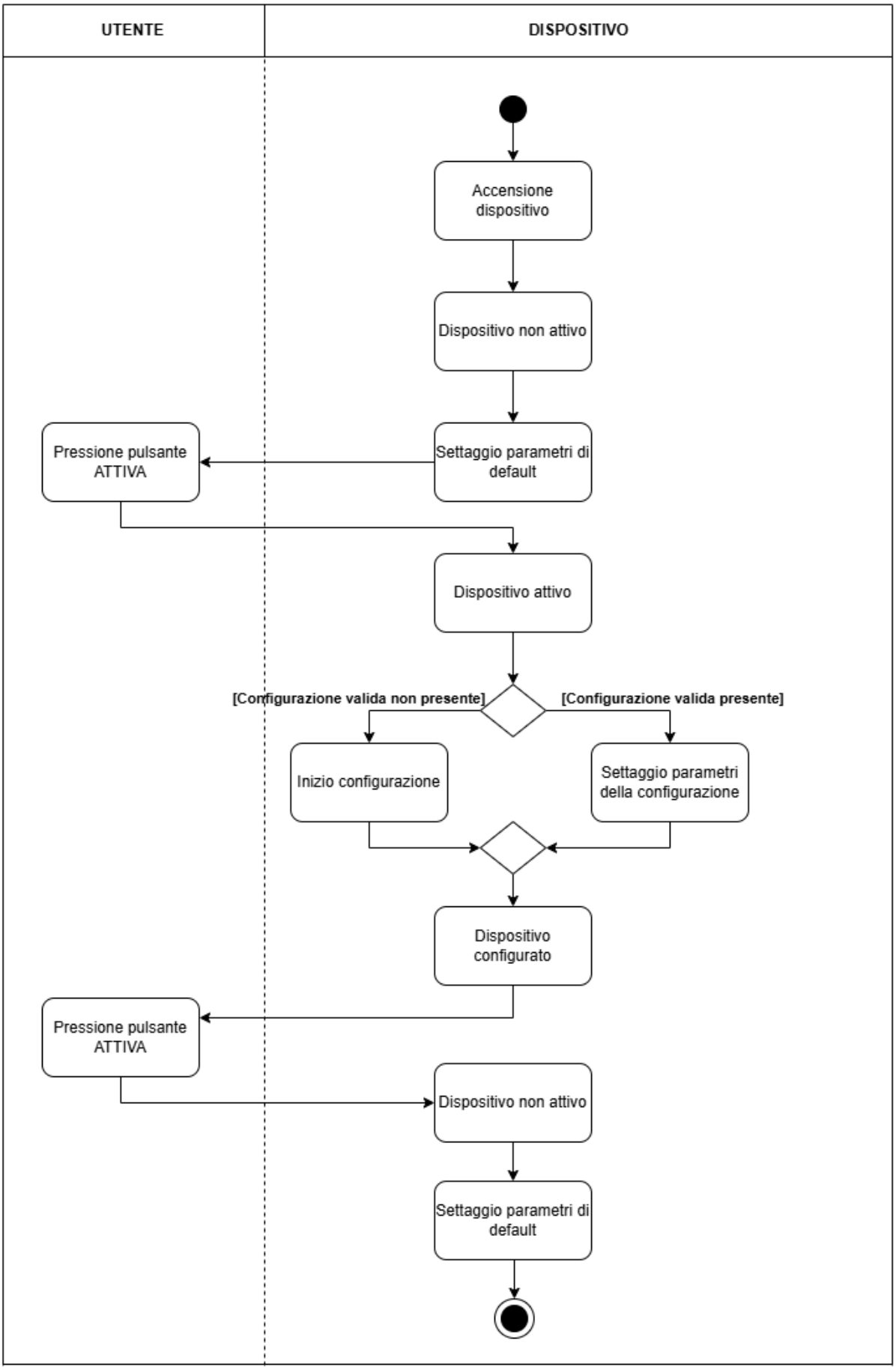
2.5.2. Attività: Avvio sistema e pressione pulsante ATTIVA

Questo diagramma descrive il flusso di operazioni per l'avvio e l'attivazione del sistema. In particolare, è stato considerato, come caso d'esempio, la doppia pressione del pulsante ATTIVA dopo l'avvio del sistema, per poter mostrare come il dispositivo passa dallo stato inattivo allo stato attivo e viceversa.

Descrizione flusso:

1) Inizio
    a. Il sistema viene collegato ad una fonte di alimentazione.
2) Accensione dispositivo
    a. Il sistema si avvia.
3) Dispositivo non attivo
    a. Il sistema, subito dopo essersi avviato, entra nello stato non attivo.
4) Settaggio parametri di default
    a. Quando il sistema si trova nello stato non attivo, tutte le funzionalità vengono configurate con i loro parametri di default, stabiliti all'interno dei requisiti.
5) Pressione pulsante ATTIVA
    a. L'utente preme il pulsante ATTIVA per attivare il sistema (poiché si trova nello stato non attivo).
6) Dispositivo attivo
    a. Il sistema entra nello stato attivo e il led utente inizia a lampeggiare
    b. Se è presente una configurazione valida, il sistema procede al settaggio dei parametri di quella configurazione.
    c. Se non è presente una configurazione valida, il sistema entra nella modalità di configurazione.
7) Inizio configurazione
    a. Il sistema entra nella modalità di configurazione, obbligando l'utente ad inserire una configurazione valida.
8) Settaggio parametri della configurazione
    a. Il sistema, presa la configurazione valida, imposta i relativi parametri a tutte le funzionalità.
9) Dispositivo configurato
    a. Dopo aver impostato correttamente una configurazione il dispositivo entra nello stato operativo, pronto all'utilizzo, stampando periodicamente il suo stato dell'interfaccia UART.
10) Pressione pulsante ATTIVA
    a. L'utente preme nuovamente il pulsante ATTIVA, stavolta per disattivare il sistema (poiché si trova nello stato attivo).
11) Dispositivo non attivo
    a. Il sistema rientra nello stato non attivo, spegnendo il led utente.
12) Settaggio parametri di default
    a) Uguale al caso 4 precedentemente descritto.
13) Fine
    a) Quest'attività si conclude con il sistema non attivo, dopo aver simulato la doppia pressione del pulsante ATTIVA.

Diagramma:


UTENTE
DISPOSITIVO
Accensione dispositivo
Dispositivo non attivo
Settaggio parametri di default
Pressione pulsante ATTIVA
Dispositivo attivo
[Configurazione valida non presente]
[Configurazione valida presente]
Inizio configurazione
Settaggio parametri della configurazione
Dispositivo configurato
Pressione pulsante ATTIVA
Dispositivo non attivo
Settaggio parametri di default

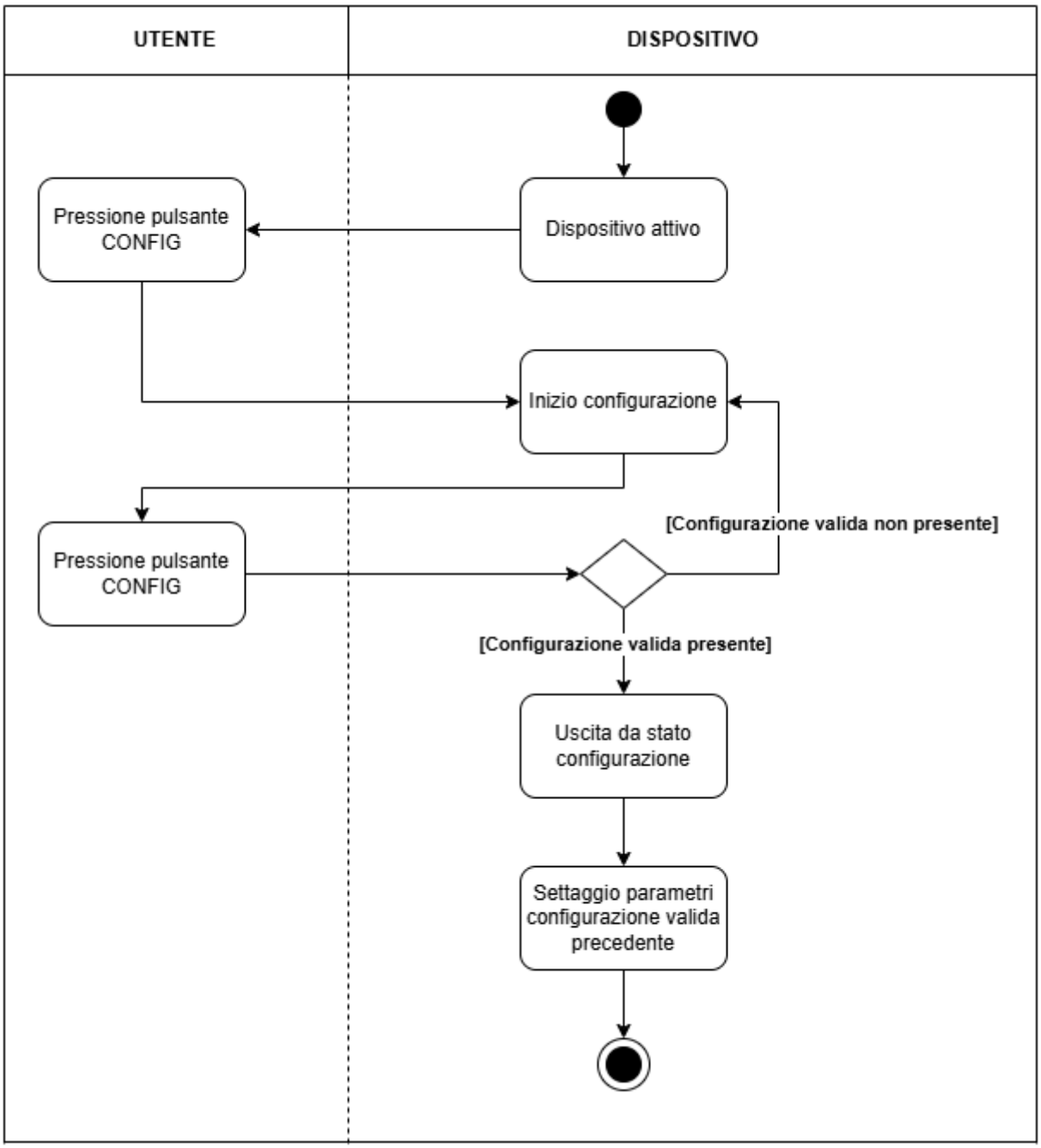
2.5.3. Attività: Pressione pulsante CONFIG

Questo diagramma descrive il flusso di operazioni relative alla pressione del pulsante CONFIG. In particolare, è stato preso come caso d'esempio la doppia pressione del pulsante CONFIG, per mostrare il flusso sia in entrata, sia in uscita, dalla modalità di configurazione del sistema.

Descrizione flusso:

1) Inizio
   a. Il sistema è operativo e funzionante
2) Dispositivo attivo
   a. Il sistema si trova nello stato attivo, ma non in modalità configurazione.
3) Pressione pulsante CONFIG
   a. L'utente preme il pulsante CONFIG per entrare in modalità configurazione.
4) Inizio configurazione
   a. Il sistema entra in modalità configurazione e permette all'utente di scegliere i parametri per le varie funzionalità.
5) Pressione pulsante CONFIG
   a. L'utente preme nuovamente il pulsante CONFIG, stavolta per uscire dalla modalità di configurazione, e il sistema deve controllare se è presente una configurazione valida per ogni funzionalità.
   b. Se non è presente una configurazione valida (caso primo avvio), l'operazione di uscita della modalità di configurazione fallisce e l'utente viene riportato al menù dove selezionare le funzionalità da configurare.
   c. Se è presente una configurazione valida, il sistema esce dallo stato di configurazione.
6) Uscita da stato di configurazione
   a. Il sistema esce dallo stato di configurazione per passare allo stato operativo.
7) Settaggio parametri configurazione valida precedente
   a. Il sistema imposta come parametri delle varie funzionalità quelli relativi alla configurazione valida precedente all'ingresso nello stato di configurazione. Ciò comporta che, se durante lo stato di configurazione, l'utente avesse inserito delle nuove configurazioni ma, al posto di salvarle, è uscito, esse verranno perse.
8) Fine
   a. Ingresso del sistema nello stato operativo.

Diagramma:


UTENTE
DISPOSITIVO
Pressione pulsante CONFIG
Dispositivo attivo
Inizio configurazione
Pressione pulsante CONFIG
[Configurazione valida non presente]
[Configurazione valida presente]
Uscita da stato configurazione
Settaggio parametri configurazione valida precedente

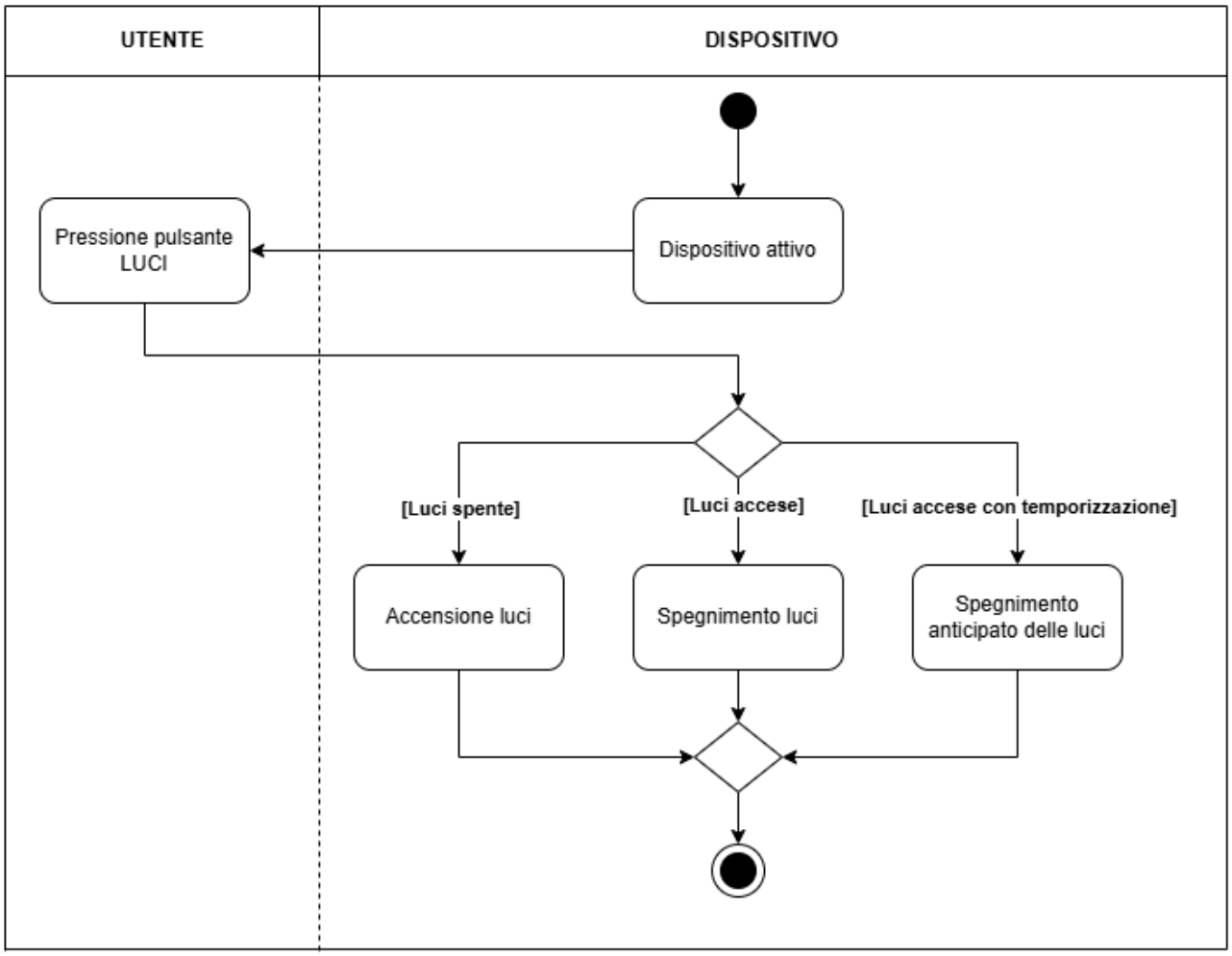
2.5.4. Attività: Pressione pulsante LUCI

Questo diagramma descrive il flusso di operazioni relative alla pressione del pulsante LUCI. Mostra gli effetti della pressione del pulsante in base allo stato corrente del sistema. La pressione di questo pulsante è disabilitata se il sistema si trova in modalità non attiva oppure, se in modalità attiva, non è in possesso di almeno una configurazione valida impostata.

Descrizione flusso:

1. Inizio
    - Sistema operativo e funzionante.
2. Dispositivo attivo
    - Il sistema si trova nello stato attivo, in possesso di almeno una configurazione valida.
3. Pressione pulsante LUCI
    - L'utente preme il pulsante LUCI per gestire le luci in base allo stato del sistema.
    - Se le luci sono attualmente spente, si accendono.
    - Se le luci sono attualmente accese, si spengono.
    - Se le luci sono attualmente accese in modalità temporizzazione, si spengono anticipatamente.
4. Accensione luci
    - L'output per il dispositivo relativo alle luci viene abilitato per la loro accensione.
5. Spegnimento luci
    - L'output per il dispositivo relativo alle luci viene abilitato per il loro spegnimento.
6. Spegnimento anticipato delle luci
    - L'output per il dispositivo relativo alle luci viene abilitato per il loro spegnimento anticipato.
7. Fine
    - Il sistema resta nello stato attivo in cui si trova, con uno stato corrente diverso in base all'effetto della pressione del pulsante.

Diagramma:


UTENTE
DISPOSITIVO
Pressione pulsante LUCI
Dispositivo attivo
[Luci spente]
[Luci accese]
[Luci accese con temporizzazione]
Accensione luci
Spegnimento luci
Spegnimento anticipato delle luci

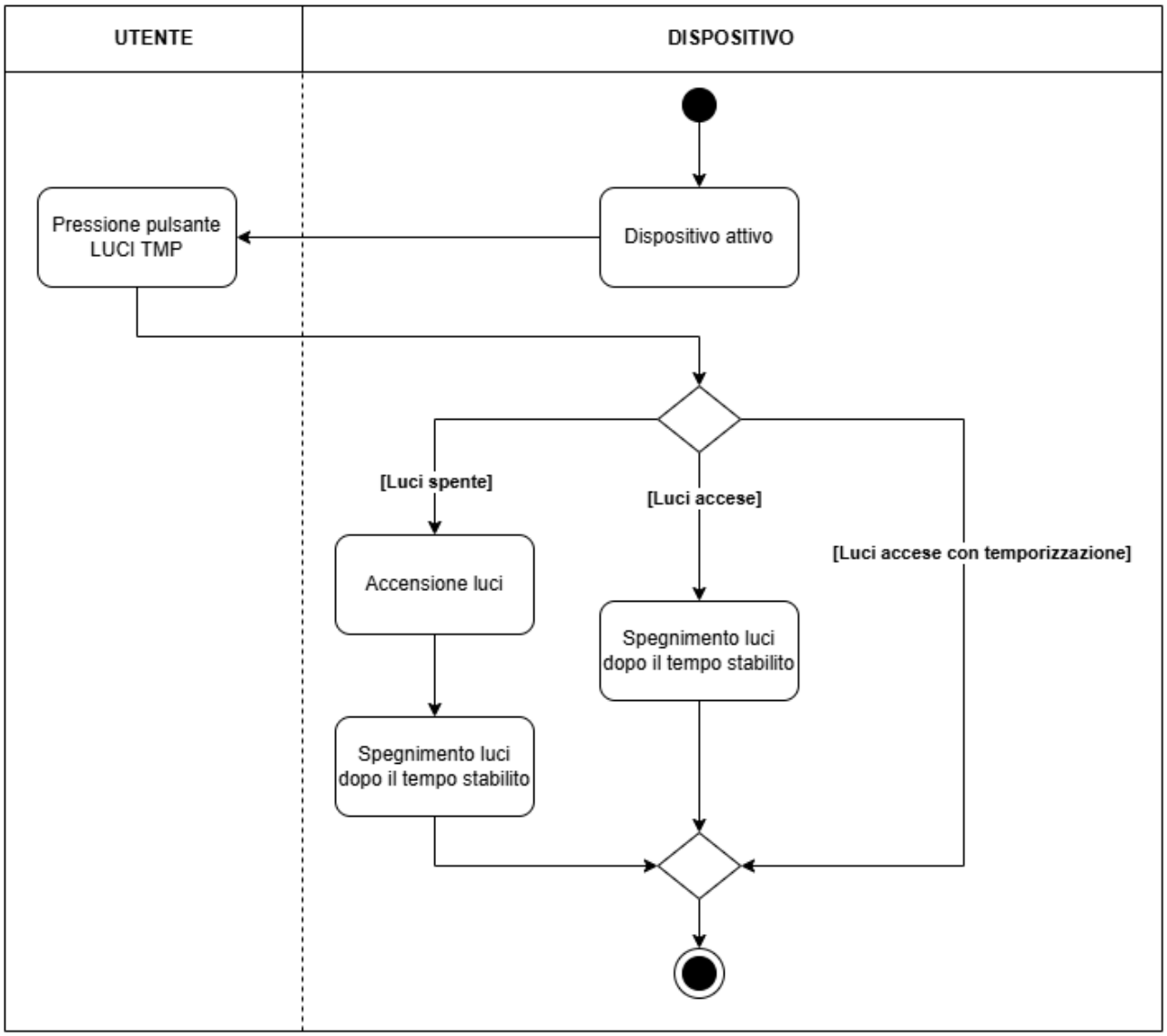
2.5.5. Attività: Pressione pulsante LUCI TMP

Questo diagramma descrive il flusso di operazioni relative alla pressione del pulsante LUCI TMP. Mostra gli effetti della pressione del pulsante in base allo stato corrente del sistema. La pressione di questo pulsante è disabilitata se il sistema si trova in modalità non attiva oppure, se in modalità attiva, non è in possesso di almeno una configurazione valida impostata oppure il periodo di tempo impostato è 0.

Descrizione flusso:

1. Inizio
    - Sistema operativo e funzionante.
2. Dispositivo attivo
    - Il sistema si trova nello stato attivo, in possesso di almeno una configurazione valida, con periodo per la temporizzazione con un valore valido impostato diverso da 0.
3. Pressione pulsante LUCI TMP
    - L'utente preme il pulsante LUCI TMP per gestire la modalità temporizzata delle luci in base allo stato del sistema.
    - Se le luci sono attualmente spente, si accendono per il periodo di tempo impostato e poi vengono spente.
    - Se le luci sono attualmente accese, rimangono accese per il periodo di tempo impostato, poi vengono spente.
    - Se le luci sono attualmente accese in modalità temporizzazione, la pressione del pulsante non ha alcun effetto.
4. Accensione luci
    - L'output per il dispositivo relativo alle luci viene abilitato per la loro accensione.
5. Spegnimento luci dopo il tempo stabilito
    - Dopo il trascorrimento del periodo di tempo impostato, l'output per il dispositivo relativo alle luci viene abilitato per il loro spegnimento.
6. Fine
    - Il sistema resta nello stato attivo in cui si trova, con uno stato corrente diverso in base all'effetto della pressione del pulsante.

Diagramma:


UTENTE
DISPOSITIVO
Pressione pulsante LUCI TMP
Dispositivo attivo
[Luci spente]
[Luci accese]
[Luci accese con temporizzazione]
Accensione luci
Spegnimento luci dopo il tempo stabilito
Spegnimento luci dopo il tempo stabilito

3. Implementazione modello con Simulink/Stateflow

In questo capitolo verrà descritta la modellazione del sistema progettato in precedenza utilizzando l'ambiente stateflow di Simulink. Si inizierà con una visione complessiva del modello per poi passare ad un'analisi più dettagliata di tutte le singole componenti presenti.

3.1. Descrizione del modello

Il modello implementato per la realizzazione del nostro progetto è stato suddiviso in diversi subchart, al fine di garantire una maggiore leggibilità e una migliore decomposizione delle varie funzionalità. Questa scelta consente anche una più semplice individuazione e risoluzione di eventuali problematiche, oltre a facilitare l'apporto di modifiche future.
A supporto del modello sono state implementate diverse funzioni MATLAB, con l'obiettivo di gestire separatamente alcune funzionalità specifiche necessarie al corretto funzionamento del sistema. Questo approccio consente di evitare un'eccessiva complessità all'interno del modello principale.
In una prima fase di test, gli input e gli output erano costituiti esclusivamente da blocchi standard di Simulink, adeguati a simulare correttamente il comportamento del sistema. Successivamente, per completare il modello, tali blocchi sono stati sostituiti con blocchi appartenenti alla libreria Embedded Coder, specifica per il supporto hardware dei processori STM32. Questa sostituzione consente l'esecuzione del modello direttamente su un microprocessore STM32, previa opportuna configurazione di alcuni file aggiuntivi.
Va precisato che il modello simulato utilizzando i blocchi standard di Simulink differisce solo in alcuni aspetti dal modello finale. Le modifiche apportate riguardano esclusivamente l'adattamento necessario per rendere il modello compatibile con l'esecuzione su un microprocessore STM32 e per permettere l'interfacciamento corretto con un terminale UART. Non sono stati aggiunti stati o modificata la logica di funzionamento durante quest'adattamento.
Per il corretto funzionamento del sistema, sono state prese anche alcune scelte implementative per gestire casistiche non perfettamente esposte nei requisiti.

3.3. Scelte progettuali ed implementative

Di seguito verranno elencate le varie scelte implementative prese per la gestione di alcuni comportamenti del nostro sistema.

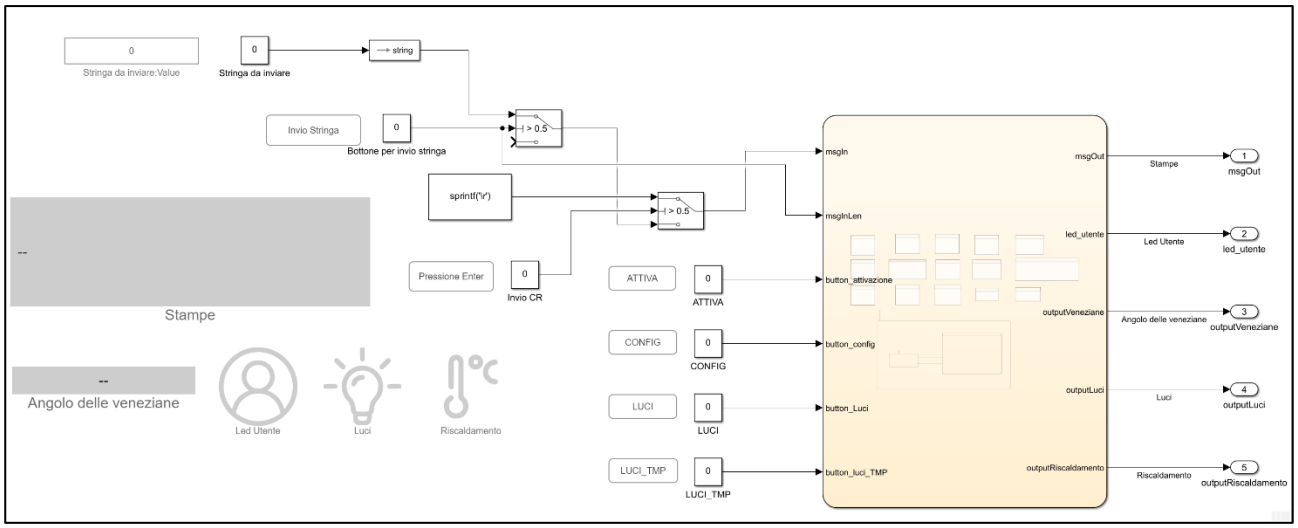
1. I pulsanti LUCI e LUCI TMP restano abilitati e funzionanti anche se il sistema si trova nello stato di configurazione.
2. Nell'inserimento del periodo di tempo relativo alla temporizzazione, l'utente è in grado di inserire anche valori con parte decimale non nulla. Questo è stato concesso poiché, da requisiti forniti, il tempo era l'unico valore senza una propria risoluzione.
3. L'input dell'utente, quando deve inserire i valori per l'angolo delle veneziane, il periodo di tempo per la temporizzazione e i gradi per il termostato, è limitato unicamente a valori numerici (anche il carattere punto quando deve inserire il periodo di tempo) e al tasto "Enter" per confermare l'invio del valore. Altri caratteri inseriti verranno direttamente ignorati dal sistema.
4. L'inserimento del periodo di tempo relativo alla temporizzazione avviene successivamente alla scelta di configurare le luci in modalità accese o spente. Quindi il flusso reale sarà:
    - L'utente sceglie se le luci saranno accese o spente.
    - Il sistema controlla l'input.
    - Se esso è valido, mostra il range di validità per inserire il periodo relativo alla temporizzazione.
    - L'utente ora è in grado di configurare la temporizzazione.

3.4. Input/Output del modello

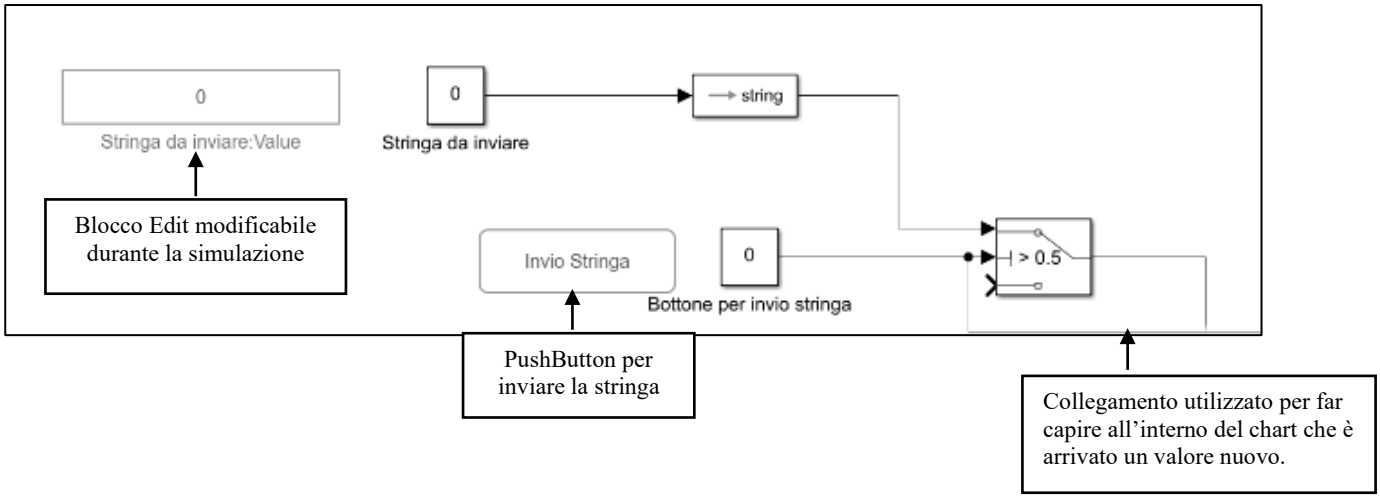
Ora verranno descritti i vari input utilizzati per il modello, sia quelli standard di Simulink, sia quelli per il microprocessore STM32.

3.4.1. Input/Output con blocchi standard Simulink

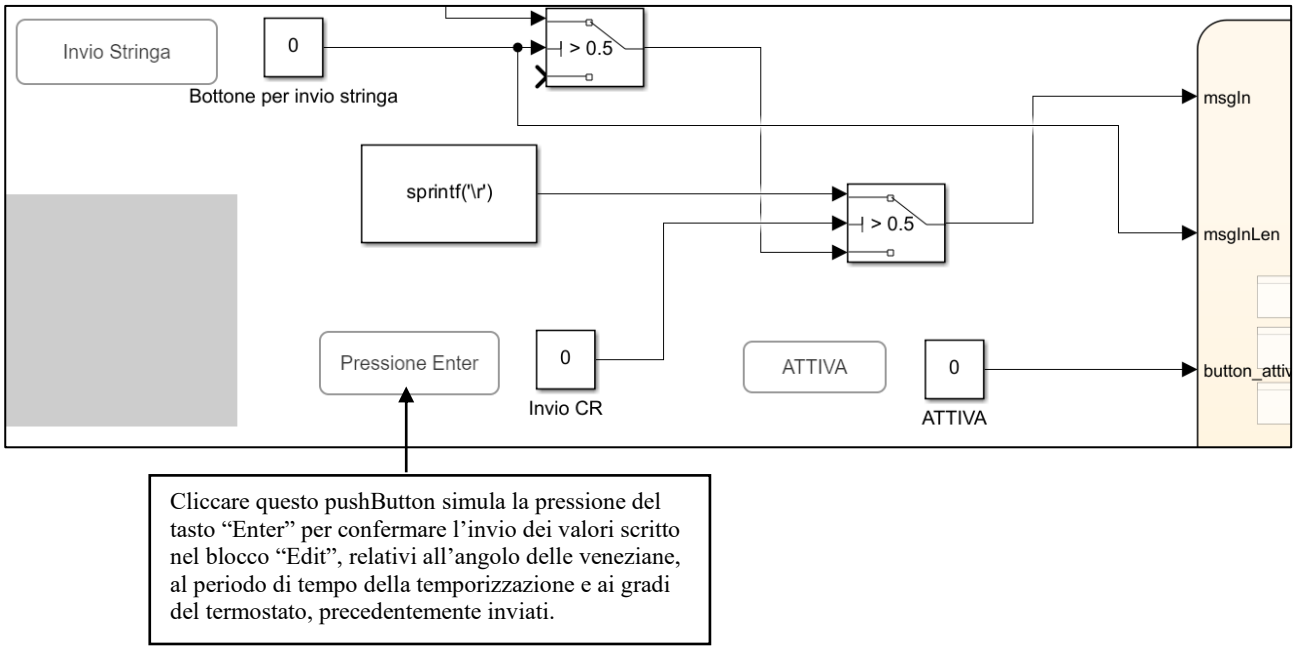
- Vista generale del modello:



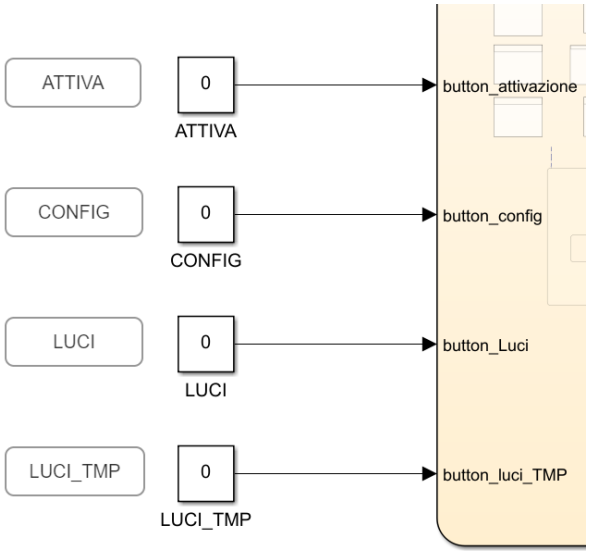
- Overview sulla simulazione dell'interfaccia UART in ingresso:



- Overview sulla simulazione della pressione del tasto “Enter”:

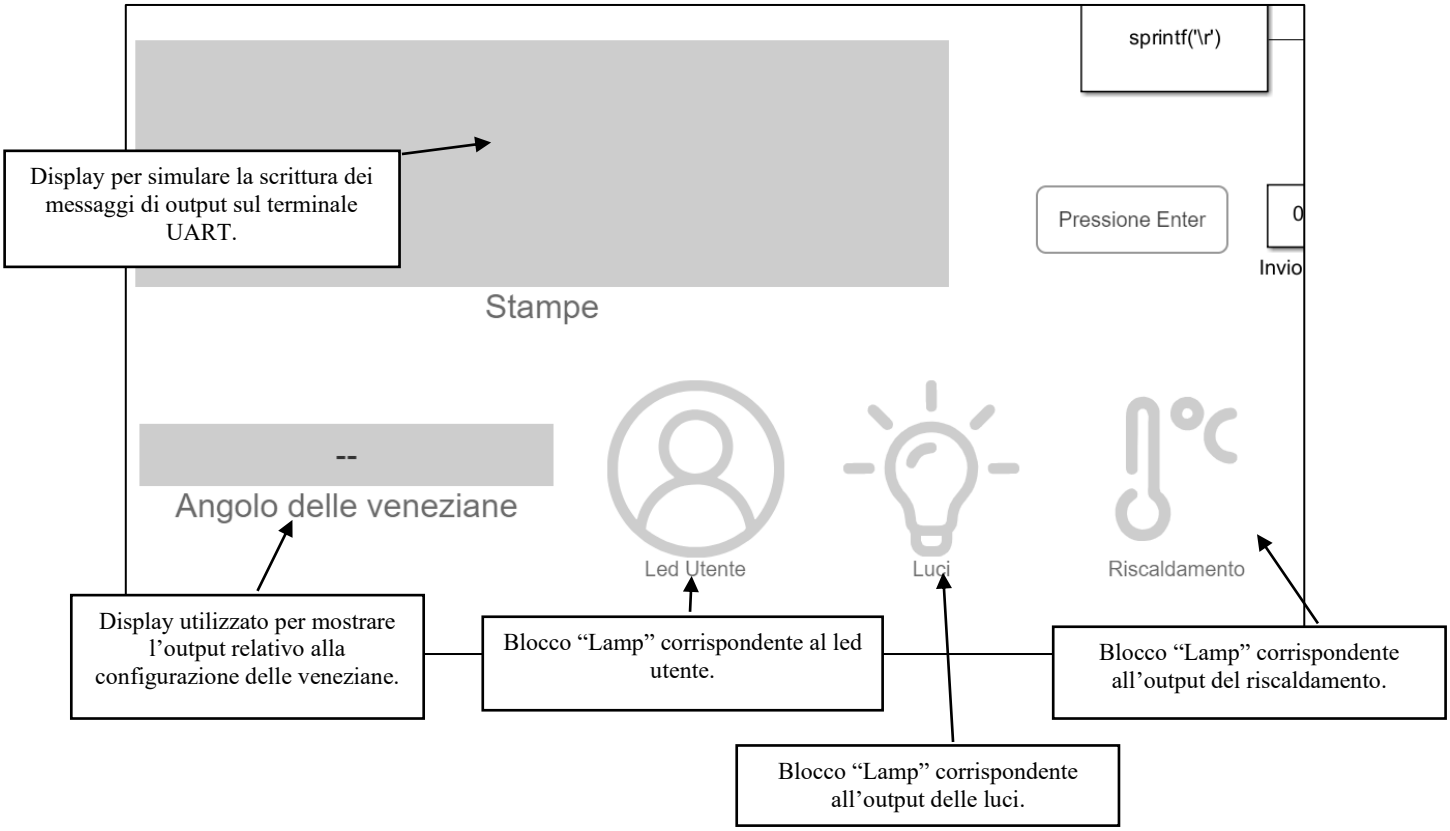


- Overview sui pulsanti corrispondenti all’input del modello:



  - PushButton ATTIVA: Pulsante utilizzato per passare dallo stato Attivo allo stato Inattivo e viceversa.
  - PushButton CONFIG: Pulsante utilizzato per passare dalla modalità configurazione a quella operativa e viceversa.
  - PushButton LUCI: Pulsante utilizzato per la gestione delle luci.
  - PushButton LUCI_TMP: Pulsante utilizzato per la gestione della temporizzazione delle luci.

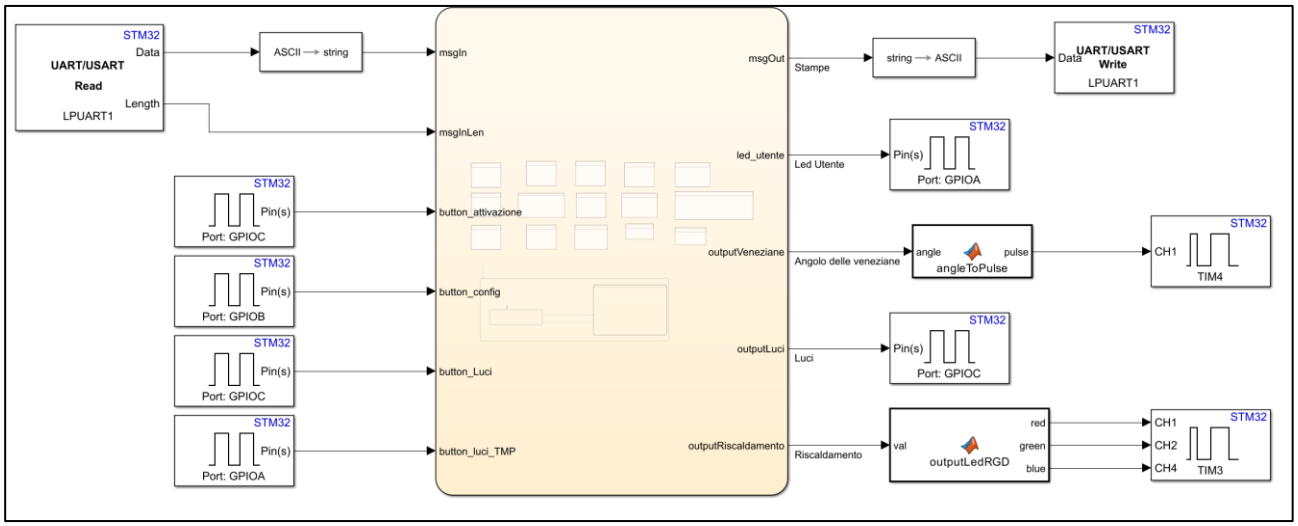
- Overview sugli output del modello:



3.4.2. Input/Output con blocchi della libreria per supporto STM32

I blocchi che verranno presentati in questo paragrafo poi verranno analizzati più nel dettaglio, successivamente, quando si parlerà della configurazione del file ".ioc" per la gestione delle periferiche del microprocessore STM32.

- Vista generale del modello:



- Analisi degli input (blocchi situati a sinistra del chart di Stateflow):
  - Il primo blocco, partendo dall'alto, rappresenta l'input proveniente dall'interfaccia UART. Questo blocco fornisce due ingressi al nostro modello:
    - Il segnale msgIn corrispondente al messaggio ricevuto in ingresso. Poiché l'uscita del blocco UART è codificata come uint8, il messaggio viene convertito in stringa tramite un apposito blocco di Simulink, prima di essere inoltrato al modello.
    - Il segnale msgInLen rappresenta la lunghezza del messaggio ricevuto in ingresso. Poiché il buffer dell'interfaccia UART è configurato per ricevere un carattere alla volta, questo parametro assume solo i valori 0 o 1. È un segnale fondamentale per indicare al sistema l'arrivo di un nuovo carattere dall'interfaccia UART.
  - Il secondo blocco, partendo dall'alto, rappresenta l'input derivante dal pulsante ATTIVA, configurato in modalità pull-down.
  - Il terzo blocco, partendo dall'alto, rappresenta l'input derivante dal pulsante CONFIG, configurato in modalità pull-down.
  - Il quarto blocco, partendo dall'alto, rappresenta l'input derivante dal pulsante LUCI, configurato in modalità pull-down.
  - Il quinto blocco, partendo dall'alto, rappresenta l'input derivante dal pulsante LUCI TMP, configurato in modalità pull-down.

- Analisi degli output (blocchi situati a destra del chart di Stateflow):
  - Il primo blocco, partendo dall'alto, corrisponde all'output che il sistema invia all'interfaccia UART. Questo output deve essere prima convertito da stringa (formato in cui viene gestito dal modello/sistema) a uint8 (formato accettato da UART). La lunghezza massima di conversione della stringa è stata impostata a 127 caratteri.
  - Il secondo blocco, partendo dall'alto, corrisponde all'output digitale relativo al led utente, alto se il led deve essere acceso, basso se deve essere spento.
  - Il terzo blocco, partendo dall'alto, corrisponde all'output relativo al servomotore delle veneziane ed è un timer, con un canale abilitato, che lavora in modalità PWM. Il segnale, quando è in uscita dal modello, deve essere convertito, tramite funzione MATLAB, nel pulse corrispondente per poter impostare correttamente il CCR in input al canale del timer. Si tiene presente che il dutyCicle si ottiene tramite l'operazione $\frac{pulse}{period} * 100\%$.
  - Il quarto blocco, partendo dall'alto, corrisponde all'output digitale relativo alle luci, alto se il devono essere acceso, basso se devono essere spente. Ciò deve essere gestito da un relè apposito, quindi il segnale di questo blocco sarà l'input di decisione del relè.
  - Il quinto blocco, partendo dall'alto, corrisponde all'output relativo al led RGB per la gestione del riscaldamento ed è un timer, con tre canali abilitati, che lavorano in modalità PWM. Il segnale in uscita dal modello viene dato in input ad una funzione MATLAB che da in output tre valori corrispondenti ai tre pulse dei tre canali del timer. Successivamente vengono impostati i rispettivi CCR dei canali del timer. Si tiene presente che il dutyCicle si ottiene tramite l'operazione $\frac{pulse}{period} * 100\%$.

3.5. Analisi delle funzioni MATLAB

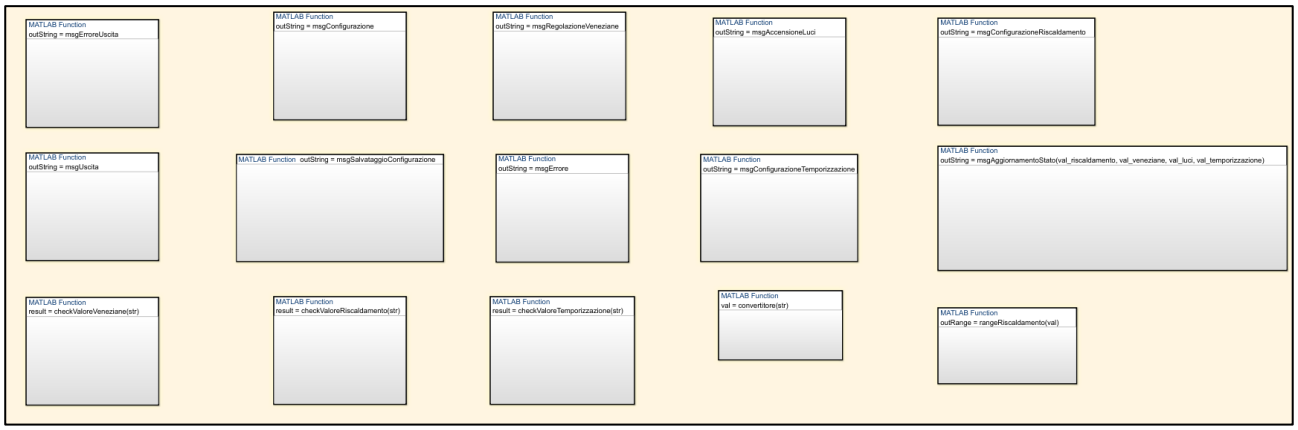
Sono state utilizzate numerose funzioni MATLAB per semplificare l'implementazione di determinati comportamenti e ridurre la complessità che il modello avrebbe avuto. Di seguito verranno analizzate singolarmente, chiarendo il loro scopo all'interno del modello.

- **angleToPulse,** funzione che prende in input un parametro *angle* e in output fornisce un parametro *pulse*. Essa viene utilizzate per convertire l'angolo in output dal modello nel corrispettivo *pulse* da dare in input al blocco STM32.
- **outputLedRGB,** funzione che prende in input un parametro *val* e in output fornisce un parametro *red*, un parametro *green* e un parametro *blue*. Essa viene utilizzata per convertire il valore del riscaldamento in output dal modello nei corrispettivi pulse dei tre canali del timer per la gestione del led RGB. I valori RGB scelti per i vari colori sono:
    - blu = 0 – 0 – 255
    - viola = 155 – 0 – 155
    - giallo = 255 – 90 – 0
    - arancione = 255 – 20 – 0
    - rosso = 255 – 0 – 0
- **msgErroreUscita**, funzione che non richiede input e restituisce una stringa di output *outString* contenente un messaggio di errore. Essa viene utilizzata per visualizzare, in caso di assenza di una configurazione valida e di tentativo di uscita dalla modalità di configurazione da parte dell'utente, un messaggio di errore.
- **msgConfigurazione**, funzione che non richiede input e restituisce una stringa di output *outString* contenente il menù di configurazione. Essa viene utilizzata per mostrare all'utente le opzioni di configurazione disponibili come la regolazione delle veneziane, l'accensione delle luci, la configurazione del riscaldamento, il salvataggio e l'uscita.
- **msgRegolazioneVeneziane**, funzione che non richiede input e restituisce una stringa *outString* contenente le istruzioni per la configurazione dei valori per la regolazione delle veneziane. Essa viene utilizzata per visualizzare un messaggio che guida l'utente nell'inserimento di un valore compreso tra 0 e 180 (con una risoluzione di 20).
- **msgAccensioneLuci**: funzione che non richiede input e restituisce una stringa *outString* contenente il menu di configurazione per l'accensione delle luci. Essa viene utilizzata per mostrare all'utente le opzioni di configurazione luci disponibili (luci accese o spente).
- **msgConfigurazioneRiscaldamento**, funzione che non richiede input e restituisce una stringa *outString* contenente le istruzioni per la configurazione del riscaldamento. Essa viene utilizzata per visualizzare nel terminale un messaggio che guida l'utente nell'inserimento di un valore di temperatura compreso tra 16 e 30, con una risoluzione di 1 grado.
- **msgConfigurazioneTemporizzazione**, funzione che non richiede input e restituisce una stringa *outString* contenente le istruzioni per la configurazione della temporizzazione delle luci. Essa viene utilizzata per guidare l'utente nell'inserimento di un valore compreso tra 5 e 30, con la possibilità di inserire 0 per disattivare la temporizzazione. In assenza di risoluzione è possibile anche inserire valori con parte decimale non nulla sempre all'interno del range di validità.
- **msgUscita**, funzione che non richiede input e restituisce una stringa *outString* contenente un messaggio informativo relativo alla fase di uscita dal sistema. Essa viene utilizzata per visualizzare un avviso che comunica all'utente il ripristino delle configurazioni precedenti prima della visualizzazione dello stato.
- **msgSalvataggioConfigurazione**, funzione che non richiede input e restituisce una stringa *outString* contenente un messaggio di conferma del salvataggio. Essa viene utilizzata per informare l'utente che la configurazione attuale è stata salvata correttamente.
- **msgErrore**, funzione che non richiede input e restituisce una stringa *outString* contenente un messaggio di errore relativo a un valore di input non valido. Essa viene utilizzata per informare l'utente dell'inserimento di un dato non corretto.
- **msgAggiornamentoStato**, funzione che prende in input quattro parametri: *val_riscaldamento*, *val_veneziane*, *val_luci* e *val_temporizzazione*. Restituisce una stringa *outString* contenente un riepilogo dettagliato dello stato operativo attuale del sistema. Essa viene utilizzata per generare un messaggio che riporta i valori attualmente impostati per

l'angolo delle veneziane (in gradi), la temperatura del riscaldamento (in gradi Celsius), lo stato delle luci (accese/spente) e l'eventuale temporizzazione delle luci (espressa in secondi o disattivata). La funzione viene utilizzata dopo il salvataggio di una configurazione o durante lo stato operativo, facilitando il monitoraggio e la verifica visiva della configurazione corrente del sistema. All'interno del blocco che contiene questa funzione è presente una funzione ausiliaria che effettua la conversione da valore numerico a string, essenziale per poter creare correttamente la stringa di output della funzione principale.

- **checkValoreVeneziane**, funzione che prende in input una stringa *str* e restituisce un valore booleano *result* indicante la validità dell'input. Essa viene utilizzata per verificare che il valore inserito per la regolazione delle veneziane sia numerico, compreso tra 0 e 180 e multiplo di 20, come richiesto dal sistema. In caso di validazione positiva, il valore può essere utilizzato per aggiornare la configurazione; in caso contrario, viene restituito un esito negativo per segnalare l'errore.
- **checkValoreRiscaldamento**, funzione che prende in input una stringa *str* e restituisce un valore booleano *result* indicante la validità dell'input. Essa viene utilizzata per verificare che il valore inserito per la configurazione del riscaldamento sia numerico, intero, compreso tra 16 e 30 inclusi. In caso di validazione positiva, il valore può essere utilizzato per impostare la temperatura; in caso contrario, viene restituito un esito negativo per segnalare un valore non conforme.
- **checkValoreTemporizzazione**, funzione che prende in input una stringa *str* e restituisce un valore booleano *result* indicante la validità dell'input. Essa viene utilizzata per verificare che il valore inserito per la temporizzazione delle luci sia numerico, eventualmente con parte decimale e che rientri nell'intervallo ammesso di 5 e 30 (inclusi), oppure 0 per disattivare la temporizzazione. In caso di input valido il valore può essere utilizzato per configurare il timer, in caso contrario, viene restituito un esito negativo per segnalare un errore di formato o di intervallo.
- **convertitore**, funzione che prende in input una stringa *str* e restituisce un valore numerico *val* corrispondente alla sua conversione. Essa viene utilizzata per trasformare una stringa contenente un numero intero o decimale in un valore numerico utilizzabile dal modello. La funzione gestisce eventuali cifre decimali e restituisce -1 in caso di formato non valido (es. caratteri non numerici o presenza multipla del punto decimale). Viene impiegata per interpretare correttamente l'input dell'utente e garantire che possa essere elaborato dal sistema.
- **rangeRiscaldamento**, funzione che prende in input un valore numerico *val* rappresentante la temperatura del riscaldamento e restituisce un intero *outRange* corrispondente all'intervallo di appartenenza. Essa viene utilizzata per categorizzare la temperatura in base a specifici intervalli di temperatura: [16-18], [19-21], [22-24], [25-27], [28-30]. Se il valore non rientra in nessuno di questi intervalli, la funzione restituisce -1 per indicare un errore o un valore fuori range. Questa classificazione facilita la gestione e il controllo della temperatura nel modello.

Overview sui blocchi funzione MATLAB:



3.6. Analisi delle variabili del modello

In questo paragrafo, prima di analizzare tutti gli stati del modello, verranno presentate le variabili utilizzate.

Nome	Tipologia	Descrizione
msgOut	OUTPUT	Variabile stringa fornita in output per il terminale UART.
msgIn	INPUT	Variabile stringa data in input dal terminale UART.
msgInLen	INPUT	Variabile double data in input dal terminale UART per segnalare l'arrivo di un nuovo messaggio.
valoreVeneziane	LOCALE	Variabile in cui viene inserita la configurazione relativa alle veneziane.
valoreRiscaldamento	LOCALE	Variabile in cui viene inserita la configurazione relativa al riscaldamento.
modLuci	LOCALE	Variabile in cui viene inserita la configurazione relativa alle luci.
valoreTemporizzazione	LOCALE	Variabile in cui viene inserita la configurazione relativa alla temporizzazione.
msg	LOCALE	Variabile stringa d'appoggio per la gestione dei buffer relativi ai valori di riscaldamento, veneziane e temporizzazione.
isValidRiscaldamento	LOCALE	Variabile booleana per verificare se il valore inserito relativo al riscaldamento è valido.

isValidVeneziane	LOCALE	Variabile booleana per verificare se il valore inserito relativo alle veneziane è valido.
isValidTemporizzazione	LOCALE	Variabile booleana per verificare se il valore inserito relativo alla temporizzazione è valido.
tmpVeneziane	LOCALE	Variabile temporanea dove salvare il valore delle veneziane da utilizzare in caso di reset.
tmpRiscaldamento	LOCALE	Variabile temporanea dove salvare il valore del riscaldamento da utilizzare in caso di reset.
tmpTemporizzazione	LOCALE	Variabile temporanea dove salvare il valore della temporizzazione da utilizzare in caso di reset.
tmpModLuci	LOCALE	Variabile temporanea dove salvare il valore delle luci da utilizzare in caso di reset.
Led_utente	OUTPUT	Variabile di output relativa alla gestione del led utente.
OFF	LOCALE	Costante numerica uguale a 0.
outputVeneziane	OUTPUT	Variabile di output per la gestione delle veneziane.
outputLuci	OUTPUT	Variabile di output per la gestione delle luci.
button_attivazione	INPUT	Variabile di input relativa al pulsante per attivare/disattivare il sistema.
outputRiscaldamento	OUTPUT	Variabile di output per la gestione del riscaldamento.
luci	LOCALE	Variabile contenente il valore corrente delle luci, utilizzabile per gestire lo stato corrente del sistema.
riscaldamento	LOCALE	Variabile contenente il valore corrente del riscaldamento, utilizzabile per gestire lo stato corrente del sistema.
tempo	LOCALE	Variabile contenente il valore corrente della temporizzazione, utilizzabile per gestire lo stato corrente del sistema.
veneziane	LOCALE	Variabile contenente il valore corrente delle veneziane, utilizzabile per gestire lo stato corrente del sistema.

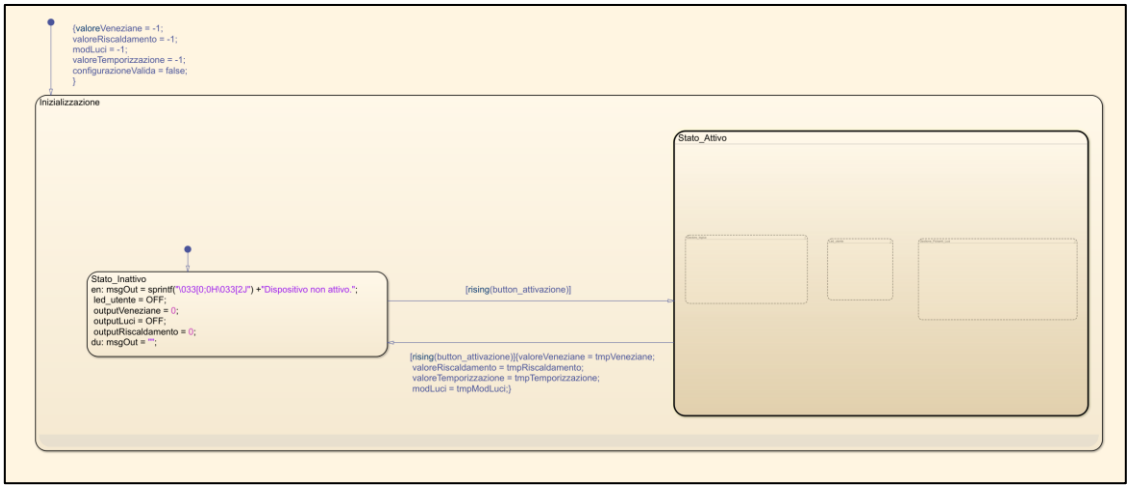
ON	LOCALE	Costante numerica uguale a 1.
button_config	INPUT	Variabile di input relativa al pulsante per entrare/uscire dallo stato di configurazione.
button_luci	INPUT	Variabile di input relativa al pulsante per accendere/spegnere le luci.
button_luci_TMP	INPUT	Variabile di input relativa al pulsante per la gestione dell'accensione temporizzata delle luci.
configurazioneValida	LOCALE	Variabile booleana per verificare se il sistema è in possesso o meno di una configurazione valida.
stopTimer	LOCALE	Variabile booleana per verificare se il timer relativo all'accensione temporizzata delle luci deve fermarsi oppure no.

3.7. Analisi degli stati del modello

Il modello implementato, come già detto in precedenza, è stato strutturato tramite l'utilizzo di diversi subchart, che contengono a loro volta gli stati relativi a specifici aspetti, collegati fra loro da transizioni. In questo paragrafo, si andranno ad analizzare tutti i singoli subchart e le varie transizioni per passare da uno stato all'altro.

3.7.1. Inizializzazione

Visualizzazione subchart:



Descrizione:

Questo subchart implementa l'inizializzazione del sistema e la comunicazione tra i due stati principali, quello Attivo e quello Inattivo. All'avvio, come si può notare dalla transizione di default del subchart, vengono settate le variabili relative alla configurazione a -1, consentendo poi la gestione efficiente della presenza di una configurazione valida per le varie funzionalità.

I due stati principali presenti in questo subchart sono lo stato Inattivo e il subchart corrispondente allo stato Attivo.

Stati:

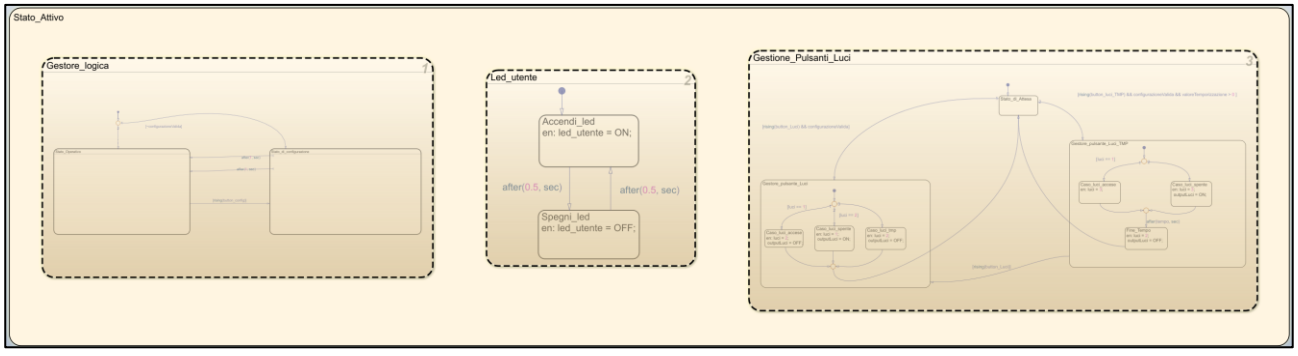
- **Stato_Inattivo**: in questo stato viene mostrato, tramite interfaccia UART, un messaggio per segnalare che il dispositivo non è attivo. Inoltre, vengono settati tutti gli output relativi alle funzionalità del sistema con i valori di default specificati nei requisiti.
- **Stato_Attivo**: verrà analizzato successivamente come subchart.

Transizioni:

- **Transizione da stato Inattivo a stato Attivo**: questo passaggio avviene quando viene rilevato un fronte di salita, tramite la funzione *rising* nativa di MATLAB, per l'input button_attivazione. Questo corrisponde ad una pressione del pulsante ATTIVA, in modalità pull-down, da parte dell'utente.
- **Transizione da stato Attivo a stato Inattivo**: questo passaggio avviene quando viene rilevato un fronte di salita, tramite la funzione *rising* nativa di MATLAB, per l'input button_attivazione. Questo corrisponde ad una pressione del pulsante ATTIVA, in modalità pull-down, da parte dell'utente. Alla transizione viene associata anche un'azione, utile quando il pulsante viene premuto mentre il dispositivo si trova nello stato di configurazione. Infatti, l'azione compiuta permette di associare alle variabili di configurazione i valori che avevano prima di entrare nello stato di configurazione. Questo permette di rigettare qualsiasi configurazione avvenuta dopo l'ingresso nello stato di configurazione se, nel mentre, viene premuto il pulsante per disattivare il dispositivo.

3.7.2. Stato_Attivo

Visualizzazione subchart:

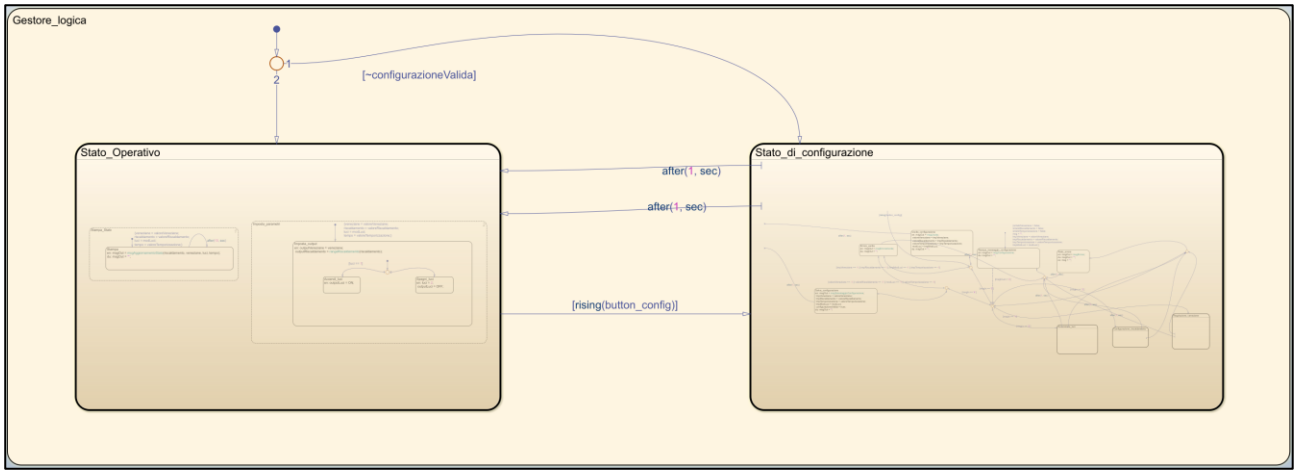


Descrizione:

Questo subchart implementa il comportamento del nostro sistema nello stato attivo. Esso è composto da tre ulteriori subchart, messi in parallelo tra loro, che corrispondono a tre aspetti del sistema che devono essere eseguiti contemporaneamente mentre il dispositivo si trova nello stato attivo. Questi tre subchart, chiamati Gestore_logica, Led_utente e Gestione_Pulsanti_Luci, verranno analizzati nel dettaglio successivamente.

3.7.3. Gestore_logica

Visualizzazione subchart:



Descrizione:

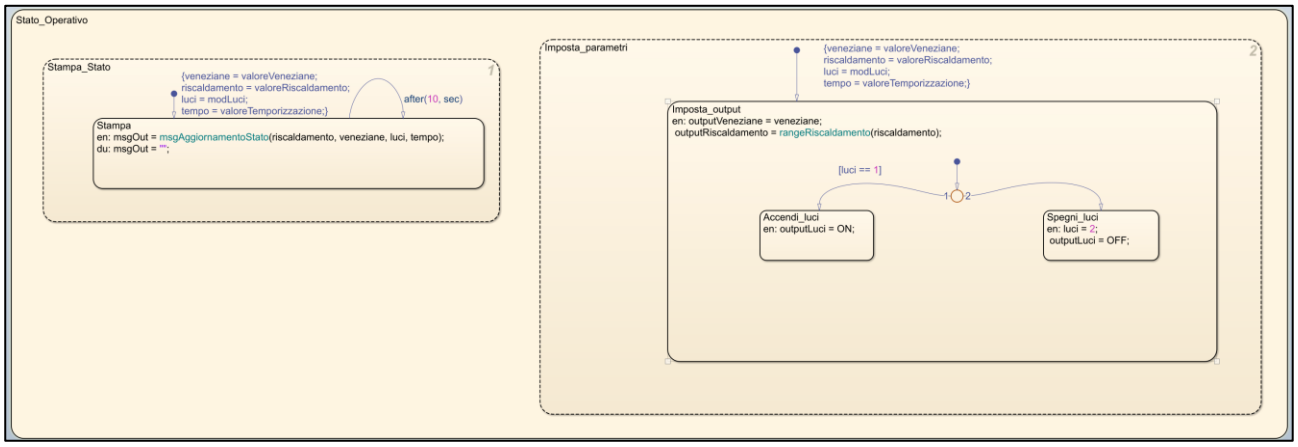
Questo subchart implementa la logica del sistema nella gestione dei parametri relativi alle funzionalità. Esso è composto da due subchart chiamati Stato_Operativo e Stato_di_configurazione, i quali verranno analizzati successivamente nel dettaglio. Ora verranno analizzate unicamente le transizioni relative al passaggio da uno stato all'altro.

Transizioni:

- **Transizione di default**: come si evince dal subchart, la transizione di default giunge ad una giunzione che consente di proseguire su due percorsi diversi. Per effettuare la scelta viene presa in considerazione la variabile booleana configurazioneValida, che viene posta a false durante l'inizializzazione e poi a true quando viene inserita una prima configurazione valida. Questa variabile e questa transizione servono ad implementare il comportamento del sistema di entrata automatica nello stato di configurazione in assenza di una configurazione valida (primo avvio del sistema). Infatti, se la variabile è posta a false, quindi non è presente alcuna configurazione valida, la transizione di default giungerà al subchart Stato_di_configurazione. Mentre, se la variabile è posta a true, quindi è presente una configurazione valida, la transizione di default giungerà al subchart Stato_Operativo.
- **Passaggio da Stato_Operativo a Stato_di_Configurazione**: questa transizione viene effettuata quando viene rilevato un fronte di salita sul segnale di input button_config, relativo alla pressione del pulsante CONFIG, configurato in modalità pull-down, da parte dell'utente.
- Le ulteriori transizioni che portano dallo Stato_di_configurazione allo Stato_Operativo partono da degli stati interni al subchart Stato_di_configurazione, quindi, di conseguenza, verranno analizzate successivamente, quando si parlerà del relativo subchart.

3.7.4. Stato_Operativo

Visualizzazione subchart:



Descrizione:

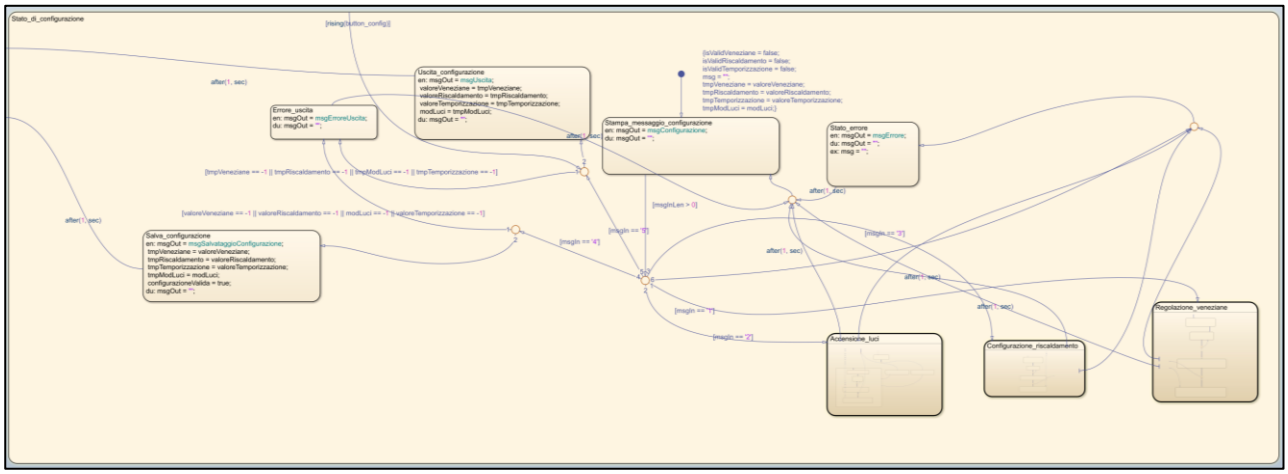
Questo subchart implementa i comportamenti che il sistema deve avere quando è attivo e non in modalità di configurazione. Esso è composto da due stati principali messi in parallelo tra loro, chiamati Stampa_Stato e Imposta_parametri, entrambi composti a loro volta da altri stati. Ora verranno analizzati nel dettaglio i singoli stati e le transizioni.

Stati:

- **Stampa_Stato**: è lo stato che si occupa di stampare periodicamente lo stato del sistema sull'intefaccia UART. Più in particoalre, a fare ciò è il suo unico stato interno chiamato Stampa che si occupa di settare la variabile di output per la stampa su terminale tramite l'utilizzo della funzione msgAggiornamentoStato. La transizione di default, presente sullo stato Stampa, serve a settare le variabili utilizzate per stampare lo stato del sistema uguali a quelle appartenenti alla configurazione corrente. Questo avviene solo all'ingresso nello stato poiché poi, successivamente, quelle variabili verranno modificate unicamente tramite i pulsanti. Se invece viene inserita di nuovo una nuova configurazione, verrà invocata di nuovo la transizione di default, perché si passerà da stato di configurazione a stato operativo. La transizione after(10, sec) sta ad indicare che il messaggio si aggiorna ogni 10 secondi.
- **Imposta_parametri**: è lo stato che si occupa di settare correttamente gli output dopo il salvataggio di una configurazione. Al suo interno ha uno stato principale, Imposta_output, all'interno del quale vengono impostati tutti gli output relativi alle varie funzionalità del sistema. Per impostare l'output relativo al riscaldamento viene utilizzata la funzione rangeRiscaldamento, mentre per le veneziane viene utilizzato direttamente il valore corrispondente all'angolo. Infine, per settare l'output delle luci, vengono utilizzati altri due stati, uno corrispondente all'accensione delle luci, uno allo spegnimento. La scelta di quale stato abilitare viene fatta tramite il parametro relativo alle luci, posto sulla transizione (luci == 1 equivale a luci accese, altrimenti spente). Come per lo stato Stampa, anche lo stato Imposta_output ha una transizione di default dove vengono settate le variabili locali da fornire agli output uguali alle variabili corrispondenti ai valori salvati nell'ultima configurazione valida.

3.7.5. Stato_di_configurazione

Visualizzazione subchart:



Descrizione:

Questo subchart implementa l'intero comportamento del sistema in relazione allo stato di configurazione. Si parte dallo stato Stampa_messaggio_configurazione e poi, in base alle interazioni dell'utente, si raggiungono gli altri stati del subchart. Lo scopo principale, dopo essere entrati in questo subchart, è quello di fornire al sistema una nuova configurazione valida. Ora procederemo all'analisi delle singole componenti al suo interno.

Stati:

- **Stampa_messaggio_configurazione:** è lo stato iniziale di questo subchart e serve a settare la variabile di output, relativa alla stampa su terminale UART, uguale ad un messaggio generato dalla funzione msgConfigurazione. La transizione di default che porta a questo stato serve ad inizializzare le variabili di appoggio booleane e la variabile msg a valori di default, mentre le variabili temporanee vengono settate uguali alle variabili relative alla configurazione presente precedente all'ingresso nello stato. Ciò serve in caso di uscita dallo stato di configurazione senza effettuare il salvataggio, caso in cui devono essere reimpostati i valori precedenti conservati nelle variabili temporanee. Queste variabili vengono utilizzate anche per controllare se precedentemente non fosse presente una configurazione valida. Infatti, in quel caso, tutte le variabili temporanee risulterebbero uguali a -1 e l'utente sarebbe impossibilitato ad uscire dallo stato di configurazione.
- **Stato_errore:** questo stato serve ad inviare in output un messaggio di errore per il terminale UART, attraverso l'utilizzo della funzione MATLAB msgErrore.
- **Accensione_luci**: verrà analizzato successivamente come subchart.
- **Configurazione_riscaldamento**: verrà analizzato successivamente come subchart.
- **Regolazione_veneziane**: verrà analizzato successivamente come subchart.
- **Salva_configurazione**: è lo stato in cui si giunge quando l'utente ha scelto di salvare una configurazione. Questo stato imposta l'output per il terminale UART, contenente un messaggio, ottenuto tramite l'utilizzo della funzione msgSalvataggioConfigurazione, per comunicare all'utente la riuscita del salvataggio, e, inoltre, reimposta tutte le variabili temporanee uguali alle variabili relative alla configurazione da salvare (questo è stato fatto altrimenti si veniva a creare un bug in relazione al passaggio allo stato inattivo subito dopo il salvataggio della configurazione) e imposta la flag per la configurazione valida a true.

- **Uscita_configurazione:** è lo stato in cui si giunge quando l'utente ha scelto di uscire dalla modalità di configurazione senza salvare. Questo stato imposta l'output per il terminale UART, contenente un messaggio, ottenuto mediante la funzione msgUscita, per comunicare all'utente la riuscita dell'uscita e la reimpostazione dei parametri della configurazione precedente all'ingresso nella modalità. Questo avviene ponendo le variabili di configurazione uguali alle variabili temporanee inizializzate all'ingresso nella modalità di configurazione (transizione di default).
- **Errore_uscita**: questo stato serve a comunicare all'utente il fallimento dell'operazione di uscita dallo stato di configurazione, mostrando, tramite relativo output, un messaggio al terminale UART ottenuto dalla funzione msgErroreUscita.

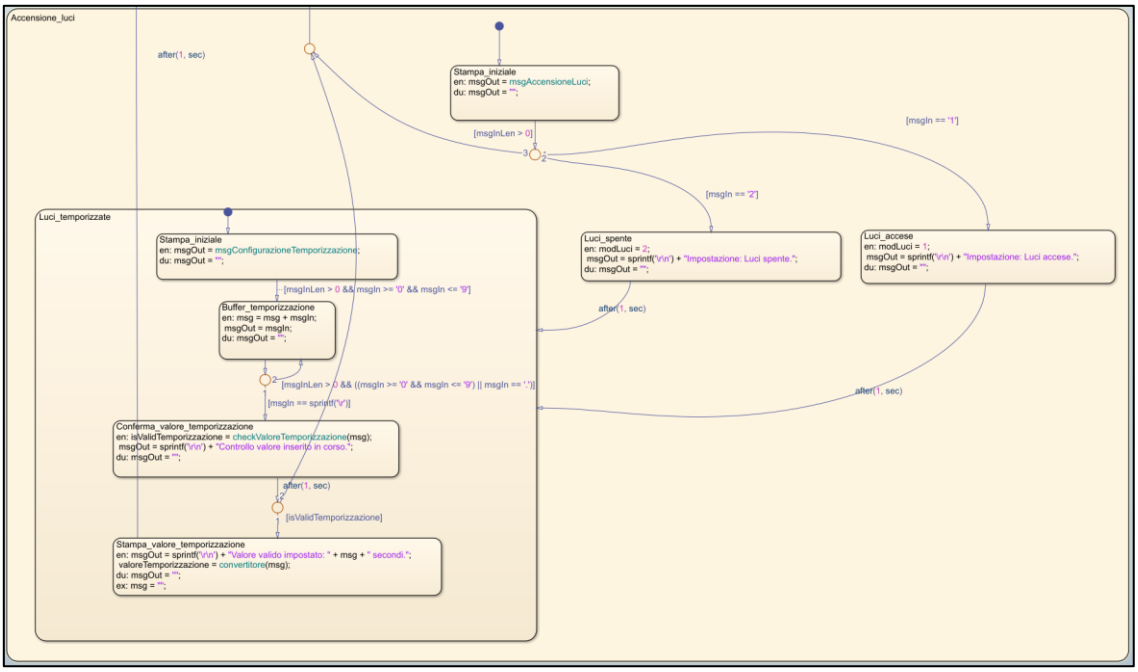
Transizioni:

- **Transizione dallo stato Stampa_messaggio_configurazione alla giunzione:** questa transizione viene attivata solo quando è valida la condizione msgInLen > 0, quindi solo quando il sistema si trova nello stato d'esame e rileva un messaggio in input dal terminale UART.
- **Percorsi in uscita dalla giunzione collegata allo stato Stampa_messaggio_configurazione**: i percorsi corrispondono alle varie scelte che possono essere state prese dall'utente tramite l'input che ha fornito dal terminale UART.
    - Se ha inserito '1' viene scelto il percorso che porta al subchart Regolazione_veneziane.
    - Se ha inserito '2' viene scelto il percorso che porta al subchart Accensione_luci.
    - Se ha scelto '3' viene scelto il percorso che porta al subchart Configurazione_riscaldamento.
    - Se ha scelto '4' viene scelto il percorso che porta a un'altra giunzione per la verifica relativa alla possibilità di effettuare un salvataggio.
    - Se ha scelto '5' viene scelto il percorso che porta a un'altra giunzione per la verifica relativa alla possibilità di effettuare un'uscita dalla modalità di configurazione.
    - Se l'utente inserisce input diversi da quelli elencati viene scelto il percorso che porta allo stato Stato_errore.
- **Giunzione relativa al salvataggio**: utilizzata per scegliere se poter effettuare un salvataggio oppure no. Ha due percorsi in uscita:
    - Se almeno uno dei parametri di configurazione ha valore -1, allora vuol dire che non è presente un parametro valido per ogni funzionalità del sistema e quindi è impossibile effettuare il salvataggio. Questo comporta che il percorso scelto è quello che porta allo stato Errore_uscita.
    - Se tutti i parametri hanno valori diversi da -1, allora vuol dire che è presente una configurazione valida e quindi può essere effettuato il salvataggio. Questo comporta che il percorso scelto è quello che porta allo stato Salva_configurazione.
- **Giunzione relativa all'uscita**: utilizzata per scegliere se poter effettuare l'uscita dallo stato di configurazione. Ha due percorsi in uscita:
    - Se almeno uno dei parametri tmp ha valore -1, allora vuol dire che la configurazione precedente all'ingresso nella modalità di configurazione non era valida (ciò avviene al primo avvio) e, di conseguenza, è impossibile uscire. Questo comporta che il percorso scelto è quello che porta allo stato Errore_uscita.
    - Se nessun parametro tmp ha valore -1, allora vuol dire che la configurazione precedente all'ingresso nella modalità di configurazione era valida e, di conseguenza, è possibile uscire. Questo comporta che il percorso scelto è quello che porta allo stato Uscita_configurazione.

- **Transizione che va dal subchart padre Stato_di_configurazione alla giunzione relativa all'uscita**: Questa transizione si attiva quando ci si trova nel subchart Stato_di_configurazione e viene rilevato un fronte di salita da parte del segnale input button_config, relativo alla pressione del pulsante CONFIG, impostato in modalità pull-down, da parte dell'utente. Questa interazione corrisponde all'azione di uscita che si può effettuare anche digitando '5' dal menù contestuale presente all'ingresso dello stato di configurazione. Date queste circostanze, la transizione giunge alla giunzione relativa all'uscita, da cui poi prosegue seguendo lo stesso comportamento descritto precedentemente per l'uscita.
- **Transizioni in uscita dagli stati Uscita_configurazione e Salva_configurazione**: queste due transizioni, che hanno come evento di attivazione il passaggio di un secondo di tempo, permettono il passaggio dal subchart Stato_di_configurazione al subchart Stato_Operativo.
- **Le transizioni rimanenti da analizzare** sono unicamente transizioni che hanno come evento di attivazione il passaggio di un secondo di tempo e permettono il passaggio dai relativi stati di partenza allo stato Stampa_messaggio_configurazione. Ciò avviene sia quando un'intera operazione di configurazione ha avuto successo, sia quando fallisce, dopo aver mostrato gli opportuni messaggi all'utente tramite interfaccia UART.

3.7.6. Accenzione_luci

Visualizzazione subchart:



Descrizione:

Questo subchart implementa il comportamento del sistema durante la fase di configurazione delle funzionalità relative alla gestione delle luci. In particolare, determina se le luci devono essere configurate in modalità accesa o spenta e gestisce anche l'impostazione della modalità di accensione temporizzata. Ora procederemo all'analisi delle singole componenti al suo interno.

Stati:

- **Stampa_iniziale**: è lo stato iniziale in cui si giunge appena si entra all'interno di questo subchart. Si occupa di comunicare all'utente, tramite interfaccia UART, le scelte che può effettuare per la configurazione delle luci. Ciò avviene impostando un messaggio, ottenuto tramite la funzione msgAccensioneLuci, nella variabile di output relativa al terminale UART.
- **Luci_accese**: è lo stato in cui viene configurata la modalità delle luci come luci accese. Si occupa anche di informare l'utente della riuscita dell'operazione, sempre tramite interfaccia UART.
- **Luci_spente**: è lo stato in cui viene configurata la modalità delle luci come luci spente. Si occupa anche di informare l'utente della riuscita dell'operazione, sempre tramite interfaccia UART.
- **Luci_temporizzate**: è lo stato che si occupa di gestire la modalità di accensione temporizzata. Comprende ulteriori stati/transizioni per gestire in maniera precisa il comportamento del sistema relativo a quest'aspetto.
  Stati:
  - **Stampa_iniziale**: è lo stato iniziale in cui si giunge appena si entra all'interno dello stato Luci_temporizzate. Si occupa di comunicare all'utente, tramite interfaccia UART, il range di valori validi che può inserire per configurare il periodo di tempo

relativo alla modalità di accensione temporizzata. Ciò avviene impostando un messaggio, ottenuto tramite la funzione msgConfigurazioneTemporizzazione, nella variabile di output relativa al terminale UART.

- **Buffer_temporizzazione**: questo stato ha la funzione di comporre il valore inserito dall'utente. È fondamentale, poiché l'interfaccia UART è configurata per inviare un solo carattere alla volta al modello. Ciò comporta una difficoltà nella gestione di valori composti da più caratteri. Lo stato risolve questo problema implementando il comportamento di un buffer: concatena progressivamente i singoli caratteri ricevuti dall'utente, tramite l'interfaccia UART, all'interno di una variabile di appoggio denominata "msg", fino a quando l'utente non preme il tasto "Enter" per confermare l'inserimento del valore completo. È importante precisare che, in questa fase, il valore è ancora trattato come una stringa; la conversione in formato numerico avverrà successivamente. Questo stato si occupa anche di mostrare sull'interfaccia UART il carattere digitato dall'utente, svolgendo anche il ruolo di echo server.
- **Conferma_valore_temporizzazione**: stato che si occupa di controllare la validità del valore inserito dall'utente tramite la funzione checkValoreTemporizzazione e di comunicare all'utente, tramite interfaccia UART, un messaggio di attesa.
- **Stampa_valore_temporizzazione**: questo stato si occupa di comunicare all'utente l'avvenuta configurazione del valore relativo al periodo di tempo da impostare. Inoltre, provvede alla conversione di tale valore da stringa a numero in virgola mobile (double), utilizzando la funzione convertitore. Infine, in uscita da questo stato, viene impostata la variabile msg a stringa vuota, per permettere un suo corretto riutilizzo successivamente.

Transizioni:

- **Transizione da Stampa_iniziale a Buffer_temporizzazione:** questa transizione ha come condizione di attivazione la ricezione di un nuovo carattere dal terminale UART, il quale deve obbligatoriamente essere un carattere compreso tra '0' e '9' inclusi. Questo permette all'utente di poter inserire solo caratteri relativi a valori numerici.
- **Giunzione posta tra Buffer_temporizzazione e Conferma_valore_temporizzazione:** in uscita dallo stato Buffer_temporizzazione si giunge direttamente a questa giunzione, che presenta due percorsi in uscita. La scelta dipende dal nuovo carattere inserito dall'utente:
  - Se esso corrisponde alla codifica della pressione del tasto "Enter", allora vuol dire che l'utente vuole confermare l'invio del valore finora inserito e quindi viene scelto il percorso che porta allo stato Conferma_valore_temporizzazione.
  - Se il valore ricevuto è diverso dalla codifica del tasto "Enter", ma rientra nel range dei valori accettabili precedentemente definito, con l'aggiunta del carattere '.' per la gestione di valori decimali, il sistema ritorna nello stato Buffer_temporizzazione, al fine di memorizzare il nuovo carattere nel buffer di composizione del valore finale.
- **Giunzione posta tra Conferma_valore_temporizzazione e Stampa_valore_temporizzazione:** in uscita dallo stato Conferma_valore_temporizzazione si giunge direttamente a questa giunzione dopo 1 secondo, la quale ha 2 percorsi in uscita. La scelta dipende dal controllo di validità del valore inserito dall'utente:
  - Se esso è valido vuol dire che l'operazione di configurazione è andata a buon fine e si procede verso lo stato Stampa_valore_temporizzazione.
  - Se esso non è valido vuol dire che l'operazione di configurazione è fallita e si procede verso lo stato di errore presente nel subchart

Stato_di_configurazione, padre di quello attuale. La transizione avviene dopo un secondo.

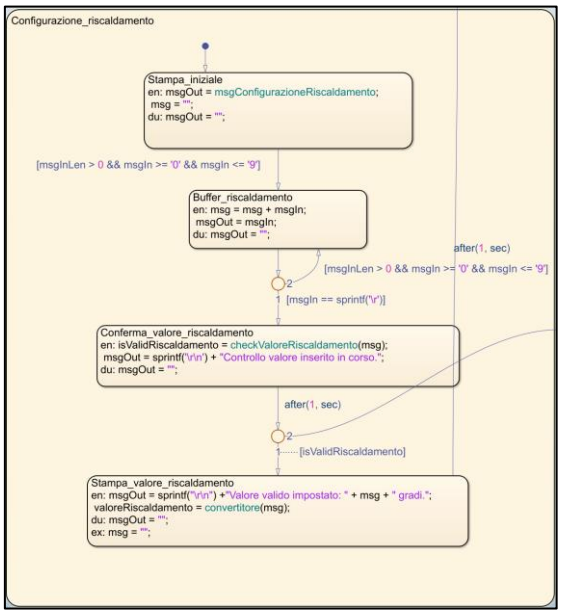
- **Transizione in uscita da Stampa_valore_temporizzazione**: questa transizione, dopo un secondo, porta il sistema nello stato Stampa_messaggio_configurazione presente nel subchart Stato_di_configurazione, padre di quello attuale. Ciò avviene per consentire all'utente di continuare a configurare altre funzionalità del sistema.

Transizioni:

- **Giunzione in uscita da Stampa_iniziale**: questa giunzione, posta in uscita dallo stato Stampa_iniziale, ha tre percorsi possibili in uscita, dipendenti dalla scelta effettuata dall'utente.
  - Se l'utente inserisce '1' indica l'intenzione di configurare le luci in modalità accese, di conseguenza viene scelto il percorso verso lo stato Luci_accese.
  - Se l'utente inserisce '2' indica l'intenzione di configurare le luci in modalità spente, di conseguenza viene scelto il percorso verso lo stato Luci_spente.
  - Se l'utente inserisce input diversi da quelli elencati viene scelto il percorso che porta allo stato Stato_errore presente nel subchart Stato_di_configurazione, padre di quello attuale.
- **Transizioni in uscita da Luci_accese e Luci_spente:** entrambe queste transizioni portano, dopo un secondo, allo stato Luci_temporizzate. Ciò permette di scegliere, dopo aver configurato correttamente la modalità delle luci, il valore relativo alla loro temporizzazione.

3.7.7. Configurazione_riscaldamento

Visualizzazione subchart:



Descrizione:

Questo subchart implementa il comportamento del sistema durante la fase di configurazione della funzionalità relativa alla gestione del riscaldamento. Ora procederemo all'analisi delle singole componenti al suo interno.

Stati:

- **Stampa_iniziale**: è lo stato iniziale in cui si giunge appena si entra all'interno di questo subchart. Si occupa di comunicare all'utente, tramite interfaccia UART, il range di validità dei valori che può inserire. Ciò avviene impostando un messaggio, ottenuto tramite la funzione msgConfigurazioneRiscaldamento, nella variabile di output relativa al terminale UART.
- **Buffer_riscaldamento**: questo stato ha la funzione di comporre il valore inserito dall'utente. È fondamentale, poiché l'interfaccia UART è configurata per inviare un solo carattere alla volta al modello. Ciò comporta una difficoltà nella gestione di valori composti da più caratteri. Lo stato risolve questo problema implementando il comportamento di un buffer: concatena progressivamente i singoli caratteri ricevuti dall'utente, tramite l'interfaccia UART, all'interno di una variabile di appoggio denominata "msg", fino a quando l'utente non preme il tasto "Enter" per confermare l'inserimento del valore completo. È importante precisare che, in questa fase, il valore è ancora trattato come una stringa; la conversione in formato numerico avverrà successivamente. Questo stato si occupa anche di mostrare sull'interfaccia UART il carattere digitato dall'utente, svolgendo anche il ruolo di echo server.
- **Conferma_valore_riscaldamento**: stato che si occupa di controllare la validità del valore inserito dall'utente tramite la funzione checkValoreRiscaldamento e di comunicare all'utente, tramite interfaccia UART, un messaggio di attesa.
- **Stampa_valore_riscaldamento**: questo stato si occupa di comunicare all'utente l'avvenuta configurazione del valore relativo al parametro del riscaldamento da impostare. Inoltre, provvede alla conversione di tale valore da stringa a numero in virgola mobile (double), utilizzando la funzione convertitore. Infine, in uscita da questo stato, viene impostata la variabile msg a una stringa vuota, per permettere un suo corretto riutilizzo successivamente.

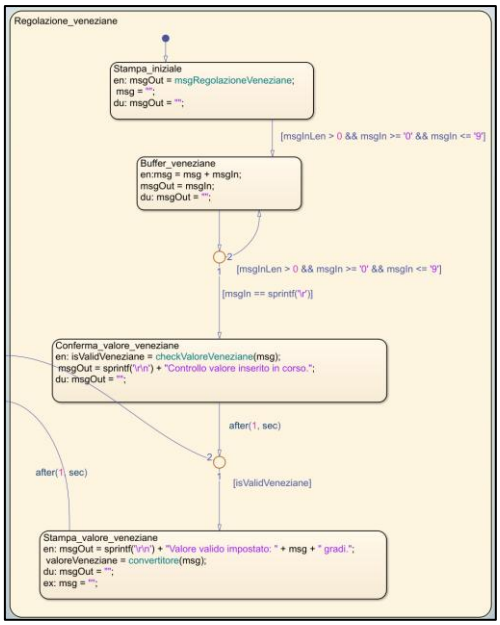
Transizioni:

- **Transizione da Stampa_iniziale a Buffer_riscaldamento:** questa transizione ha come condizione di attivazione la ricezione di un nuovo carattere dal terminale UART, il quale deve obbligatoriamente essere un carattere compreso tra '0' e '9' inclusi. Questo permette all'utente di poter inserire solo caratteri relativi a valori numerici.
- **Giunzione posta tra Buffer_riscaldamento e Conferma_valore_riscaldamento:** in uscita dallo stato Buffer_riscaldamento si giunge direttamente a questa giunzione, la quale ha 2 percorsi in uscita. La scelta dipende dal nuovo carattere inserito dall'utente:
    - Se esso corrisponde alla codifica della pressione del tasto "Enter", allora vuol dire che l'utente vuole confermare l'invio del valore finora inserito e quindi viene scelto il percorso che porta allo stato Conferma_valore_riscaldamento.
    - Se il valore ricevuto è diverso dalla codifica del tasto "Enter", ma rientra nel range dei valori accettabili precedentemente definito, il sistema ritorna nello stato Buffer_riscaldamento, al fine di memorizzare il nuovo carattere nel buffer di composizione del valore finale.

- **Giunzione posta tra Conferma_valore_riscaldamento e Stampa_valore_riscaldamento:** in uscita dallo stato Conferma_valore_riscaldamento si giunge direttamente a questa giunzione dopo 1 secondo, la quale ha 2 percorsi in uscita. La scelta dipende dal controllo di validità del valore inserito dall'utente:
    - Se esso è valido vuol dire che l'operazione di configurazione è andata a buon fine e si procede verso lo stato Stampa_valore_riscaldamento.
    - Se esso non è valido vuol dire che l'operazione di configurazione è fallita e si procede verso lo stato di errore presente nel subchart Stato_di_configurazione, padre di quello attuale. La transizione avviene dopo un secondo.
- **Transizione in uscita da Stampa_valore_riscaldamento**: questa transizione, dopo un secondo, porta il sistema nello stato Stampa_messaggio_configurazione presente nel subchart Stato_di_configurazione, padre di quello attuale. Ciò avviene per consentire all'utente di continuare a configurare altre funzionalità del sistema.

3.7.8. Regolazione_veneziane

Visualizzazione subchart:



Descrizione:

Questo subchart implementa il comportamento del sistema durante la fase di configurazione della funzionalità relativa alla gestione delle veneziane. Ora procederemo all'analisi delle singole componenti al suo interno.

Stati:

- **Stampa_iniziale**: è lo stato iniziale in cui si giunge appena si entra all'interno di questo subchart. Si occupa di comunicare all'utente, tramite interfaccia UART, il range di validità dei valori che può inserire. Ciò avviene impostando un messaggio, ottenuto tramite la funzione msgConfigurazioneVeneziane, nella variabile di output relativa al terminale UART.
- **Buffer_veneziane**: questo stato ha la funzione di comporre il valore inserito dall'utente. È fondamentale, poiché l'interfaccia UART è configurata per inviare un solo carattere alla

volta al modello. Ciò comporta una difficoltà nella gestione di valori composti da più caratteri. Lo stato risolve questo problema implementando il comportamento di un buffer: concatena progressivamente i singoli caratteri ricevuti dall'utente, tramite l'interfaccia UART, all'interno di una variabile di appoggio denominata "msg", fino a quando l'utente non preme il tasto "Enter" per confermare l'inserimento del valore completo. È importante precisare che, in questa fase, il valore è ancora trattato come una stringa; la conversione in formato numerico avverrà successivamente. Questo stato si occupa anche di mostrare sull'interfaccia UART il carattere digitato dall'utente, svolgendo anche il ruolo di echo server.

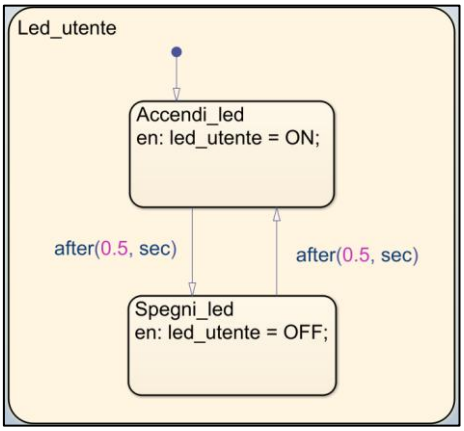
- **Conferma_valore_veneziane**: stato che si occupa di controllare la validità del valore inserito dall'utente tramite la funzione checkValoreVeneziane e di comunicare all'utente, tramite interfaccia UART, un messaggio di attesa.
- **Stampa_valore_veneziane**: questo stato si occupa di comunicare all'utente l'avvenuta configurazione del valore relativo al parametro delle veneziane da impostare. Inoltre, provvede alla conversione di tale valore da stringa a numero in virgola mobile (double), utilizzando la funzione convertitore. Infine, in uscita da questo stato, viene impostata la variabile msg a stringa vuota, per permettere un suo corretto riutilizzo successivamente.

Transizioni:

- **Transizione da Stampa_iniziale a Buffer_veneziane:** questa transizione ha come condizione di attivazione la ricezione di un nuovo carattere dal terminale UART, il quale deve obbligatoriamente essere un carattere compreso tra '0' e '9' inclusi. Questo permette all'utente di poter inserire solo caratteri relativi a valori numerici.
- **Giunzione posta tra Buffer_veneziane e Conferma_valore_veneziane:** in uscita dallo stato Buffer_veneziane si giunge direttamente a questa giunzione, la quale ha 2 percorsi in uscita. La scelta dipende dal nuovo carattere inserito dall'utente:
    - Se esso corrisponde alla codifica della pressione del tasto "Enter", allora vuol dire che l'utente vuole confermare l'invio del valore finora inserito e quindi viene scelto il percorso che porta allo stato Conferma_valore_veneziane.
    - Se il valore ricevuto è diverso dalla codifica del tasto "Enter", ma rientra nel range dei valori accettabili precedentemente definito, il sistema ritorna nello stato Buffer_veneziane, al fine di memorizzare il nuovo carattere nel buffer di composizione del valore finale.
- **Giunzione posta tra Conferma_valore_veneziane e Stampa_valore_veneziane:** in uscita dallo stato Conferma_valore_veneziane si giunge direttamente a questa giunzione dopo 1 secondo, la quale ha 2 percorsi in uscita. La scelta dipende dal controllo di validità del valore inserito dall'utente:
    - Se esso è valido vuol dire che l'operazione di configurazione è andata a buon fine e si procede verso lo stato Stampa_valore_veneziane.
    - Se esso non è valido vuol dire che l'operazione di configurazione è fallita e si procede verso lo stato di errore presente nel subchart Stato_di_configurazione, padre di quello attuale. La transizione avviene dopo un secondo.
- **Transizione in uscita da Stampa_valore_veneziane**: questa transizione, dopo un secondo, porta il sistema nello stato Stampa_messaggio_configurazione presente nel subchart Stato_di_configurazione, padre di quello attuale. Ciò avviene per consentire all'utente di continuare a configurare altre funzionalità del sistema.

3.7.9. Led_utente

Visualizzazione subchart:



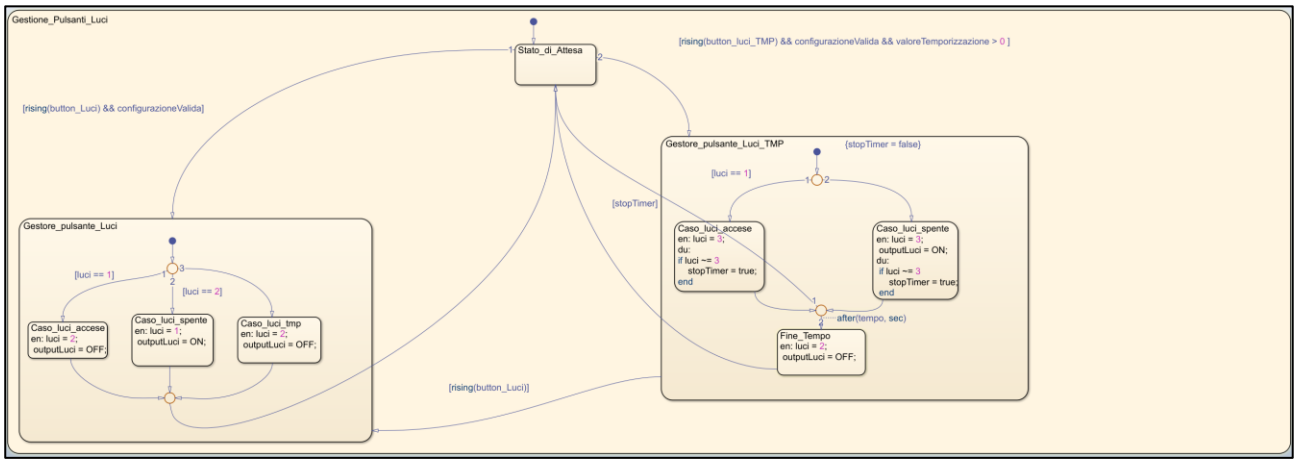
Descrizione:

Questo subchart implementa il lampeggio del led utente quando il dispositivo è attivo. È composto da due stati, Accendi_led e Spegni_led, che si devono alternare ogni 0.5 secondi (il led deve lampeggiare con una frequenza di 2 Hz).

- Stato Accendi_led: in questo stato viene impostato l'output relativo al led utente ad ON (1), che corrisponde all'accensione del led.
- Stato Spegni_led: in questo stato viene impostato l'output relativo al led utente ad OFF (0), che corrisponde allo spegnimento del led.
- Transizione da Accendi_led a Spegni_led e viceversa: il passaggio avviene dopo 0.5 secondi, grazie all'utilizzo dell'evento "after" per la transizione.

3.7.10. Gestione_Pulsanti_Luci

Visualizzazione subchart:



Descrizione:

Questo subchart implementa la gestione degli input relativi alla pressione dei pulsanti LUCI e LUCI TMP da parte dell'utente. Ora procederemo all'analisi delle singole componenti al suo interno.

Stati:

- **Stato_di_attesa:** è uno stato di attesa in cui il sistema non fa nulla, ma attende input dall'utente. In ingresso ad esso viene settata un variabile booleane per gestire se il timer deve essere stoppato o meno durante la sua esecuzione.
- **Gestore_pulsante_luci:** è lo stato che si occupa di gestire l'input derivante dalla pressione del pulsante LUCI da parte dell'utente. Esso è composto da ulteriori stati e transizioni.
  Stati:
  - **Caso_luci_accese**: stato che rappresenta il caso in cui il pulsante viene premuto e il sistema si trova nello stato con le luci accese. Di conseguenza, come da requisiti, le luci vengono spente e viene cambiato lo stato corrente relativo alle luci.
  - **Caso_luci_spente**: stato che rappresenta il caso in cui il pulsante viene premuto e il sistema si trova nello stato con le luci spente. Di conseguenza, come da requisiti, le luci vengono accese e viene cambiato lo stato corrente relativo alle luci.
  - **Caso_luci_tmp**: stato che rappresenta il caso in cui il pulsante viene premuto e il sistema si trova nello stato con le luci accese in modalità temporizzata. Di conseguenza, come da requisiti, le luci vengono spente anticipatamente e viene cambiato lo stato corrente relativo alle luci.

  Transizioni:
  - **Giunzione successiva alla transizione di default**: questa giunzione ha tre possibili percorsi in uscita che vengono scelti in base allo stato corrente delle luci.
    - Se luci == 1 vuol dire che sono attualmente accese, quindi si procede verso lo stato Caso_luci_accese.
    - Se luci == 2 vuol dire che sono attualmente spente, quindi si procede verso lo stato Caso_luci_spente.
    - Se luci diverso dai valori precedenti vuol dire che sono attualmente accese in modalità temporizzata, quindi si procede verso lo stato Caso_luci_tmp.
  - **Giunzione in uscita dallo stato Gestore_pulsante_luci**: questa giunzione è la giunzione a cui arrivano tutti e tre gli stati precedentemente elencati e riporta il sistema nello Stato_di_Attesa, in attesa di nuovi input dell'utente.
- **Gestore_pulsante_luci_tmp:** è lo stato che si occupa di gestire l'input derivante dalla pressione del pulsante LUCI da parte dell'utente. Esso è composto da ulteriori stati e transizioni.
  Stati:
  - **Caso_luci_accese**: stato che rappresenta il caso in cui il pulsante viene premuto e il sistema si trova nello stato con le luci accese. Di conseguenza, come da requisiti, le luci devono continuare a restare accese, ma solo per il tempo prestabilito dalla temporizzazione. Viene cambiato unicamente lo stato corrente delle luci. Durante l'esecuzione di questo stato, se il valore della modalità luci varia (viene salvata/resettata una configurazione) allora la variabile booleana stopTimer viene messa a true per stoppare l'esecuzione del timer.
  - **Caso_luci_spente**: stato che rappresenta il caso in cui il pulsante viene premuto e il sistema si trova nello stato con le luci spente. Di conseguenza, come da requisiti, le luci vengono accese in modalità temporizzata e viene cambiato lo stato corrente relativo alle luci. Durante l'esecuzione di questo stato, se il valore della modalità luci varia (viene salvata/resettata una configurazione) allora la variabile booleana stopTimer viene messa a true per stoppare l'esecuzione del timer.
  - **Fine_tempo**: stato che rappresenta la fine del periodo impostato per la temporizzazione. Ciò comporta lo spegnimento delle luci e il cambiamento dello stato corrente ad esse associato.

Transizioni:

- **Giunzione successiva alla transizione di default**: questa giunzione ha tre possibili percorsi in uscita che vengono scelti in base allo stato corrente delle luci.
  - Se luci == 1 vuol dire che sono attualmente accese, quindi si procede verso lo stato Caso_luci_accese.
  - Se luci != 1 vuol dire che sono attualmente spente, quindi si procede verso lo stato Caso_luci_spente.
- **Giunzione in ingresso allo stato Fine_tempo**: questa giunzione è la giunzione a cui giungono gli stati Caso_luci_accese e Caso_luci_spente. Ha due percorsi possibili:
  - Se, durante l'esecuzione dei due stati a monte in attesa dello scadee del timer, la variabile booleana stopTimer viene messa a true, allora il percorso scelto è quello diretto verso lo stato Stato_di_attesa, interrompendo l'esecuzione del timer.
  - In alternativa, se il timer giunge al termine, sceglie il percorso che va in ingresso allo stato Fine_tempo dopo un tot di secondi, stabilito dal parametro relativo al periodo della temporizzazione.
- **Transizione in uscita da Fine_tempo**: è una transizione che riporta il sistema nello Stato_di_attesa, in attesa di successivi input da parte dell'utente.

Transizioni:

- **Transizione da Stato_di_attesa a Gestore_pulsante_luci**: questa transizione ha come condizione di attivazione il rilevamento di un fronte di salita del segnale button_luci, corrispondente alla pressione del pulsante LUCI da parte dell'utente, e la presenza di una configurazione valida. Ciò consente di disabilitare il pulsante quando il dispositivo è attivo ma non è provvisto di una configurazione valida (primo avvio), mentre resta abilitato negli altri casi nei quali il dispositivo è attivo.
- **Transizione da Stato_di_attesa a Gestore_pulsante_luci_TMP**: questa transizione ha come condizione di attivazione il rilevamento di un fronte di salita del segnale button_luci_tmp, corrispondente alla pressione del pulsante LUCI TMP da parte dell'utente, e la presenza di una configurazione valida. Ciò consente di disabilitare il pulsante quando il dispositivo è attivo ma non è provvisto di una configurazione valida (primo avvio), mentre resta abilitato negli altri casi nei quali il dispositivo è attivo. Inoltre, il valore impostato come periodo della temporizzazione deve essere maggiore di 0, poiché, se avvenisse il contrario (periodo == 0) allora significherebbe che l'utente ha disattivato la temporizzazione, quindi il pulsante deve risultare disabilitato.
- **Transizione da Gestore_pulsante_luci a Gestore_pulsante_luci_TMP:** questa transizione ha come condizione di attivazione il rilevamento di un fronte di salita del segnale button_luci, corrispondente alla pressione del pulsante LUCI da parte dell'utente. Serve ad implementare il comportamento relativo alla pressione del pulsante LUCI durante l'accensione temporizzata delle luci del sistema.

4. Configurazione hardware

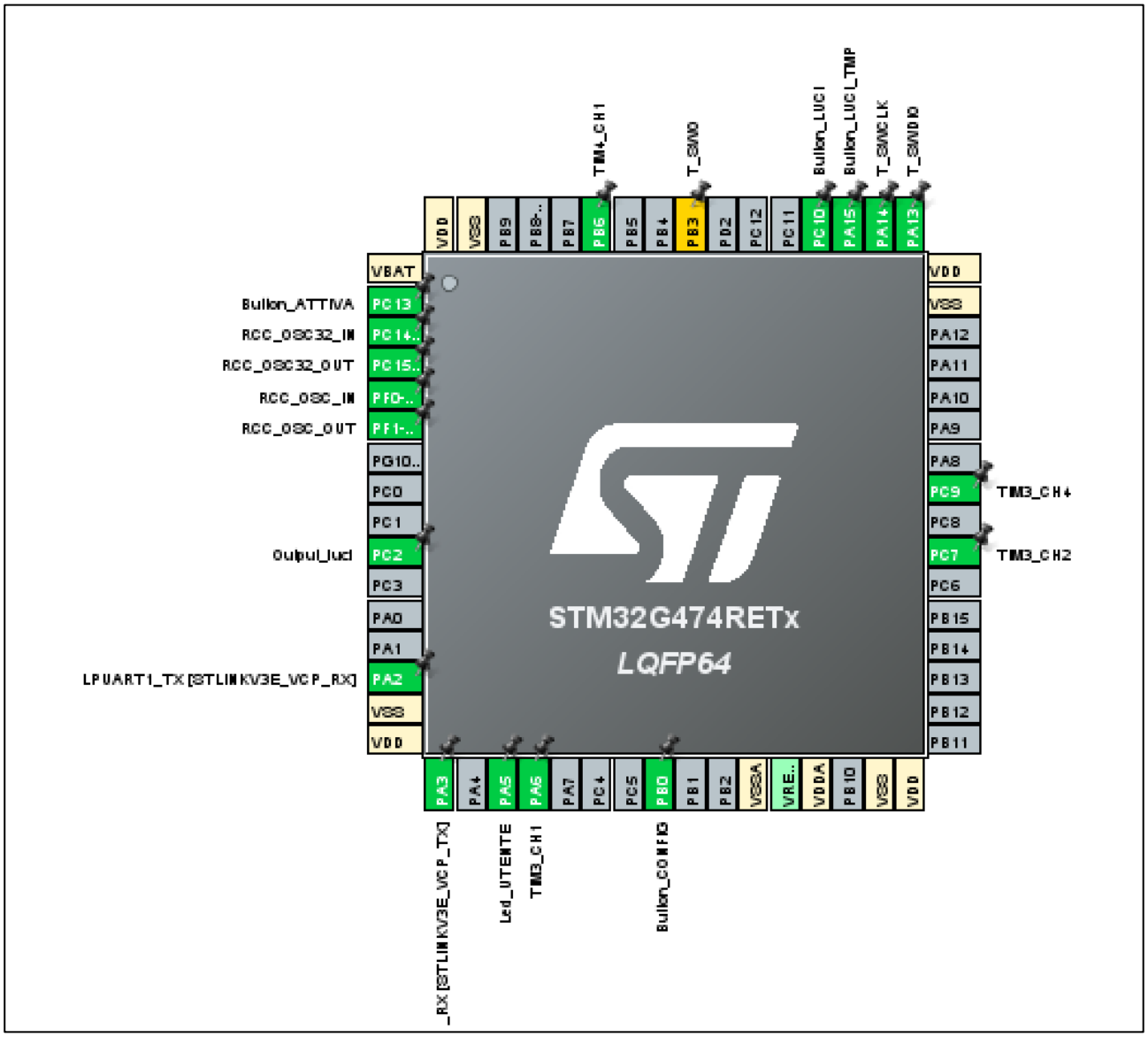
In questo paragrafo verranno descritte tutte le scelte relative alla configurazione hardware per poter eseguire il nostro sistema su un microprocessore STM32. In particolar modo, viene utilizzato il microprocessore STM32G474RE.

4.1. Pin setup

In questa sono riportati i pin da utilizzare per la connessione dei dispositivi di input e di output del sistema.

Dispositivo	**PIN**	**Nota**
LDG RGB colore rosso	PA6	TIM 3 PWM CH1
LDG RGB colore verde	PC7	TIM 3 PWM CH2
LDG RGB colore blu	PC9	TIM 3 PWM CH4
LED utente	PA5	
Controllo servomotore	PB6	TIM4 PWM CH1
Controllo relè	PC2	
Pulsante CONFIG	PB0	
Pulsante ATTIVA	PC13	User button
Pulsante LUCI	PC10	
Pulsante LUCI TMP	PA15	
RX	PA3	LPUART RX
TX	PA2	LPUART TX

Pin overview:



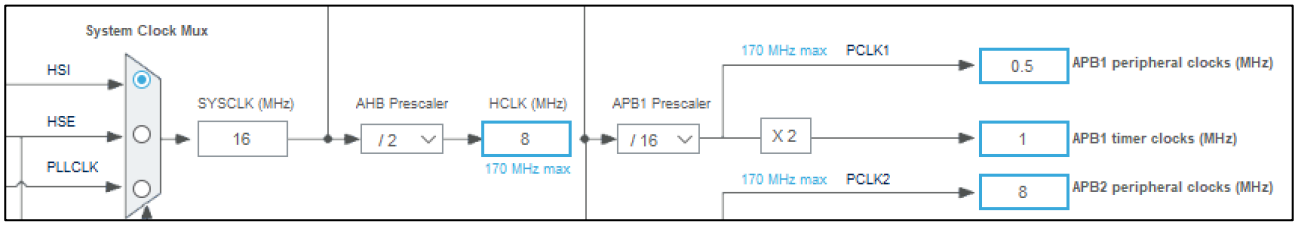
4.3. Configurazione file .ioc

Per permettere il funzionamento del sistema sul microprocessore, è necessario configurare un file specifico .ioc che si occupa dell'interfacciamento con le caratteristiche del microprocessore. Nel nostro caso il file in questione è chiamato "**Progetto_Gruppo21_IZ.ioc**", collocato nell'omonima cartella. Di seguito verranno presentate tutte le configurazioni effettuate.

4.3.1. Clock Configuration

La configurazione principale relativa al clock è il settaggio del clock principale di funzionamento del microprocessore a 16 MHz, tramite la modifica del parametro relativo al ***System Clock Multiplexer*** (impostato su HSI). Inoltre, si è scelto di impostare il clock di funzionamento del bus interno relativo ai timer di interesse (timer 3 e timer 4) a 1 Mhz. Il bus in questione è APB1 timer clocks. Gli altri valori si sono adattati in automatico a queste scelte effettuate.

Overview Clock Configuration:



4.3.2. GPIO Configuration

I pin relativi agli input e agli output digitali sono stati configurati nei seguenti modi:

- **Pin PA5:** pin relativo al led utente, configurato come pin di output mantenendo le configurazioni di default.
- **Pin PA15:** pin relativo al pulsante LUCI_TMP, configurato come pin di input, in modalità pull-down.
- **Pin PB0:** pin relativo al pulsante CONFIG, configurato come pin di input, in modalità pull-down.
- **Pin PC2:** pin relativo all'output delle luci, gestito tramite l'ausilio di un relè, configurato come pin di output mantenendo le configurazioni di default.
- **Pin PC10:** pin relativo al pulsante LUCI, configurato come pin di input, in modalità pull-down.
- **Pin PC13:** pin relativo al pulsante ATTIVA, configurato come pin di input, in modalità pull-down.

Overview Pin Configuration:

Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/Pull-do...	Maximum output sp...	Fast Mode	User Label	Modified
PA5	n/a	Low	Output Push Pull	No pull-up and no pu...	Low	n/a	Led_UTENTE	☑
PA15	n/a	n/a	Input mode	Pull-down	n/a	n/a	Button_LUCI_TMP	☑
PB0	n/a	n/a	Input mode	Pull-down	n/a	n/a	Button_CONFIG	☑
PC2	n/a	Low	Output Push Pull	No pull-up and no pu...	Low	n/a	Output_luci	☑
PC10	n/a	n/a	Input mode	Pull-down	n/a	n/a	Button_LUCI	☑
PC13	n/a	n/a	Input mode	Pull-down	n/a	n/a	Button_ATTIVA	☑

4.3.3. Timer Configuration

I timer utilizzati per l'implementazione di questo sistema sono due: Timer 3 e Timer 4. Del primo sono stati utilizzati i canali 1,2, e 4 mentre del secondo solo il canale 1. Di seguito verranno riportate le configurazioni specifiche.

- **TIMER 3**
  Il timer 3 viene utilizzato per la gestione del led RGB, per questo vengono utilizzati tre canali, uno per ogni pin del led da dover gestire. Tutti e 3 i canali sono configurati in modalità "PWM Generation" per generare un segnale PWM in ingresso ai pin del led RGB. In questo modo è possibile gestire le diverse colorazioni, in base ai diversi valori dei dutyCicle. Il timer ha ***Prescaler*** impostato a 0 e ***Counter Period*** impostato a 255. Ciò permette di impostare il pulse, utilizzando la proprietà counts del blocco appartenente alla libreria di supporto, esattamente ad un valore compreso tra 0 e 255, simulando il valore corrispettivo del colore nella scala RGB. Si tiene presente che il dutyCicle si ottiene tramite l'operazione $\frac{pulse}{period} * 100\%$. I primi due canali hanno il ***Pulse*** di default impostato a 0

mentre il terzo a 255, per simulare l'accensione del led RGB con un colore blu di default (ciò è superfluo perché ci pensa già il codice a settare questi parametri all'avvio).

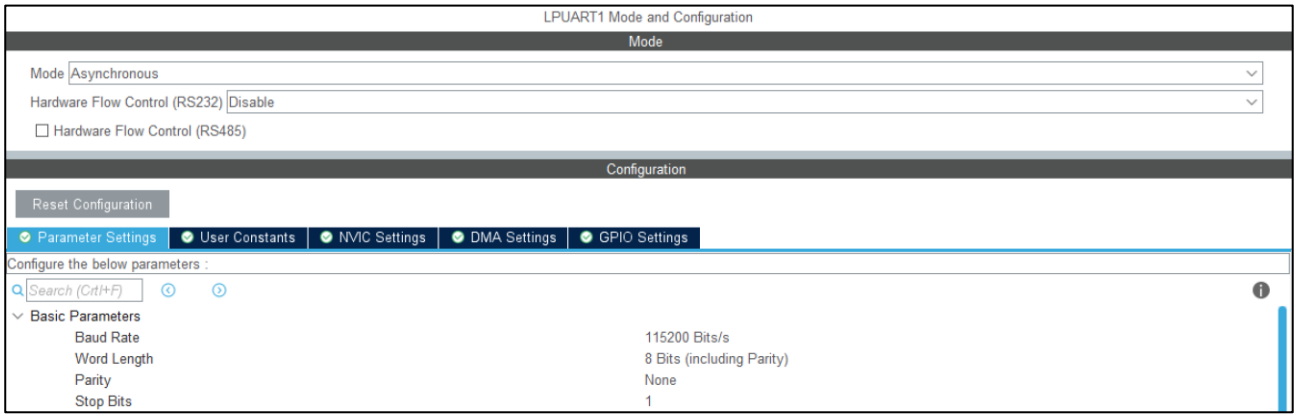
- **TIMER 4**
  Il timer 4 viene utilizzato per la gestione del servomotore. Viene utilizzato un unico canale che corrisponde al pin di input del servomotore per la gestione dell'angolo di posizionamento. Esso è configurato in modalità "PWM Generation" per generare un segnale PWM in ingresso al pin del servomotore. In questo modo è possibile gestire le diverse angolazioni, in base ai diversi valori del dutyCicle. Il timer ha ***Prescaler*** rimpostato a 0 e ***Counter Period*** impostato a 20.000. Per scelta implementativa abbiamo deciso di gestire il pulse come con l'altro timer, in modalità counts per quanto riguarda il blocco appartenente alla libreria di supporto, ma la gestione in questo caso era immediata anche in modalità percentage, andando a considerare direttamente il dutyCicle. Si tiene presente che il dutyCicle si ottiene tramite l'operazione $\frac{pulse}{period} * 100\%$. Basta impostare il dutyCicle al 2.5% del ***Counter Period*** (Pulse = 500) per impostare il servomotore a 0° oppure al 12.5% del ***Counter Period*** (Pulse = 2500) per impostare il servomotore a 180°. Tutte le altre angolazioni si ottengono tramite proporzione matematica. Di default, il canale abilitato ha il ***Pulse*** a 500 per simulare il servomotore settato a 0° (ciò è superfluo perché ci pensa già il codice a settare questo parametro all'avvio)

4.3.4. UART Configuration

Il cuore del sistema è l'interfaccia UART utilizzata per permettere all'utente di fornire input e visualizzare messaggi informativi relativi alle azioni che può effettuare. UART è un protocollo asincrono per la trasmissione dei dati, e per avere una visualizzazione di questi dati tramite terminale, viene utilizzata l'applicazione **PuTTY.** Nel nostro sistema, in particolare, viene usata la modalità **LPUART**, ovvero la "**Low Power UART**", che utilizza come **Pin RX** e **Pin TX** rispettivamente i pin **PA3** e **PA2**. Le configurazioni utilizzate sono:

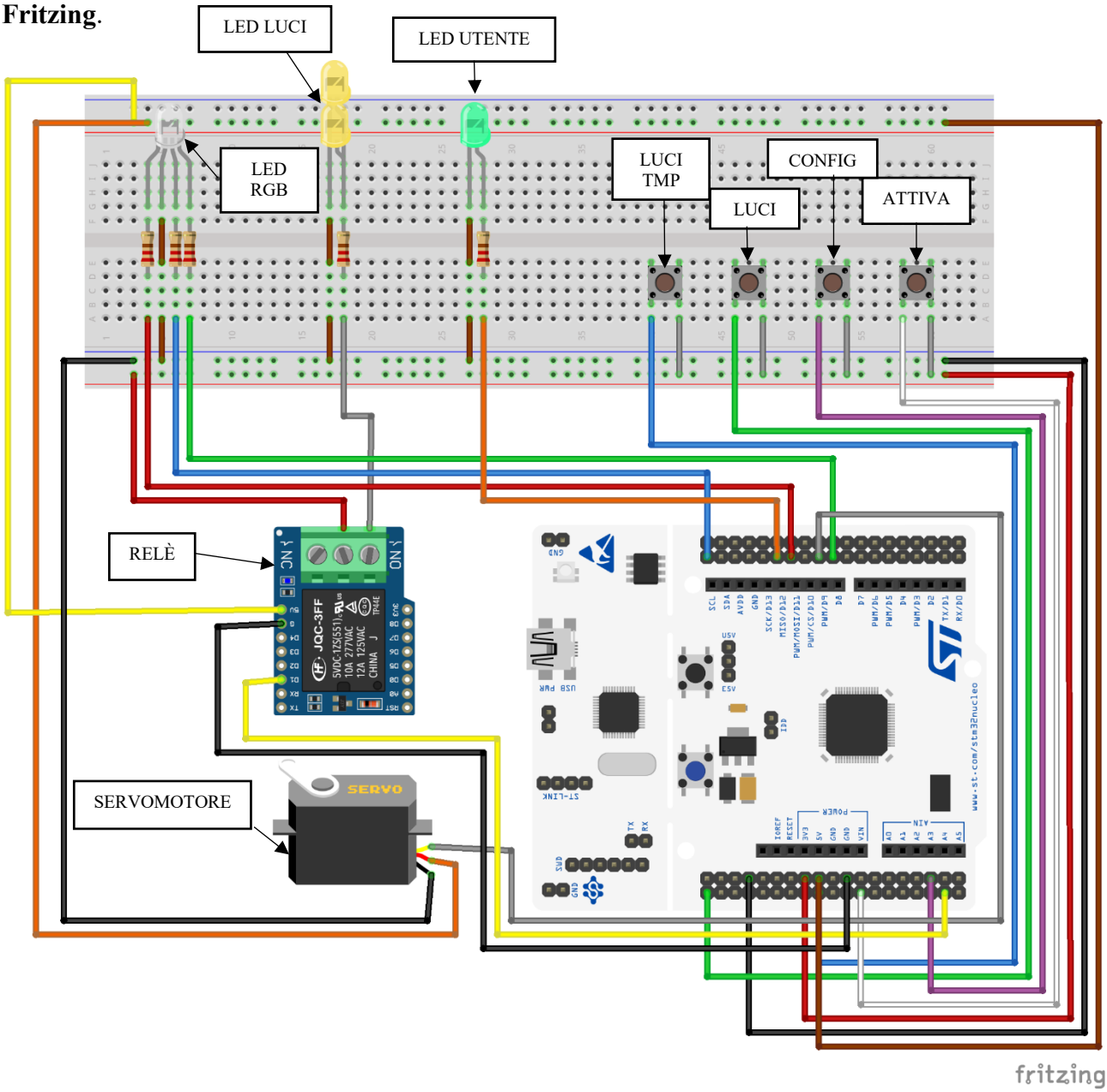
- Baud Rate a 115200.
- Buffer di trasmissione di 1 byte (8 bits).

Overview UART Configuration:



4.4. Overview configurazione circuito

La scheda contenente il microprocessore STM32 è stata collegata, tramite breadboard e jumper, a tutti i vari dispositivi per testare il corretto funzionamento del sistema implementato. Di seguito è mostrata la configurazione adottata, riprodotta virtualmente tramite l'utilizzo dell'applicazione **Fritzing**.



Considerazioni:

- I vari pulsanti inseriti rispettano la modalità con cui sono stati configurati, ovvero quella pull-down. I resistori relativi sono stati abilitati a bordo del microprocessore, mentre i collegamenti fisici sono stati effettuati correttamente, inserendo il pulsante tra l'alimentazione (VCC a 3.3V) e il relativo pin che gestisce l'input.
- I resistori, posizionati in serie ai rispettivi pin (anodi) dei vari led, hanno una resistenza pari a 220 Ohm, valore adeguato a consentire un corretto funzionamento dei led.
- Il relè mostrato in figura è di una tipologia differente da quello utilizzato realmente. Quello reale dispone in input solo di tre pin (GND, VCC e Switch), ma il funzionamento è lo

stesso. I pin per la gestione delle luci sono i pin COM (comune), NO (Normally Open) e NC (Normally Closed). Al pin COM è stata fornita l'alimentazione che si vuole pilotare (nel nostro caso una 3.3V), mentre il pin NO è stato collegato in serie al resistore e ai pin anodi dei led in parallelo corrispondenti alle luci (il pin NC non è stato utilizzato). Ciò consente di pilotare l'accensione e lo spegnimento dei led. Quando il relè non è sollecitato, quindi il pin Switch in input ha valore basso, allora all'interno del relè si crea un circuito tra i pin COM e NC, lasciando scollegato NO. Questo comporta che la tensione portata ai led relativi alle luci è pari a zero, ovvero i led restano spenti. Quando invece il relè è sollecitato, quindi il pin Switch in input ha valore alto, allora all'interno del relè si crea un circuito tra i pin COM e NO, lasciando scollegato NC. Questo comporta che la tensione portata ai led relativi alle luci è pari alla tensione posseduta dal pin COM, in questo caso 3.3V. Questo permette l'accensione dei led.

- Il led RGB presente su Fritzing ha una disposizione dei pin differente da quella del componente reale. Su Fritzing l'ordine del pin è: ANODO RED, CATODO COMUNE, ANODO BLU e poi ANODO GREEN. Mentre per il componente reale l'ordine dei pin è: ANODO RED, CATODO COMUNE, ANODO GREEN e poi ANODO BLU.

Configurazione reale: (Alcuni collegamenti potrebbero risultare differenti da quelli mostrati tramite la virtualizzazione di Fritzing)