

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE ED ELETTRICA E MATEMATICA APPLICATA



CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

WARPEX

Nome	Cognome	Matricola	E-Mail
Michele	Martino	0622702424	m.martino48@studenti.unisa.it
Francesco	Quagliuolo	0622702412	f.quagliuolo@studenti.unisa.it
Emanuele	Relmi	0622702368	e.relmi@studenti.unisa.it
Benito	Senese	0622702425	b.senese1@studenti.unisa.it

ANNO ACCADEMICO 2023/2024

INDICE

1 User Stories	4
1.1 US1 - Apertura Cancello	4
1.2 US2 - Chiusura Cancello	4
1.3 US3 - Regolazione Tempo Chiusura Automatica	5
1.4 US4 - Regolazione Tempo Lavoro	5
1.5 US5 - Riapertura Automatica con Rilevazione Ostacolo	6
1.6 US6 - Gestione Sicura del Cancello in Presenza di Ostacoli	6
1.7 US7 - Gestione del sensore di chiusura per determinare lo stato del cancello .	7
1.8 US8 - Errore in caso di malfunzionamento del sensore di chiusura	7
1.9 US9 - Avvio Chiusura Cancello senza Sensore Attivo	8
1.10 US10 - Indicazione del Cancello in Movimento	8
1.11 US11 - Indicazione di Errore di Chiusura	9
1.12 US12 - Indicazione di Ostacolo	9
1.13 US13 - Indicazione Cancello Chiuso	9
1.14 US14 - Indicazione Cancello Aperto	10
2 Use Cases	11
2.1 Apertura Cancello [US1-US10-US14]	11
2.2 Chiusura Cancello [US2-US10-US13]	12
2.3 Regolazione Tempo di Chiusura Automatica [US3]	13
2.4 Regolazione Tempo di Lavoro [US4]	13
2.5 Riapertura Automatica con Rilevazione Ostacolo [US5-US14]	14
2.6 Gestione Richieste in Presenza di Ostacoli [US6-US12]	14
2.7 Determinazione e Comunicazione Stato Cancello [US7-US13]	15

INDICE

2.8 Gestione dello Stato di Errore [US8-US11]	16
2.9 Chiusura automatica all'accensione [US9]	17
2.10 General Use Case	18
3 Activity Diagrams	19
3.1 Apertura e Chiusura Cancello [Scenario 1]	19
3.2 Regolazioni [Scenario 2]	21
3.3 Gestione Stato e Ostacoli [Scenario 3]	23
3.4 Stato di Errore [Scenario 4]	25
3.5 Chiusura Automatica all'accensione [Scenario 5]	27
4 State Diagram	29
4.1 Input, Output e Stati	29
4.2 Logica di Funzionamento	31
4.2.1 Stato <i>Inattivo</i>	31
4.2.2 Macrostate <i>Chiusura</i>	32
4.2.3 Macrostate <i>Errore</i>	32
4.2.4 Macrostate <i>Apertura</i>	33
4.2.5 Macrostate <i>Ostacolo</i>	34
4.2.6 Macrostate <i>Regolazioni</i>	34
5 Modellazione del Sistema nell'ambiente Simulink	36
5.1 Gate Chart	36
5.1.1 Variabili	37
5.1.2 Stati e Funzionamento	37
5.2 Tuning Charts	40
5.2.1 Variabili	40
5.2.2 Stati e Funzionamento	40
5.3 Obstacle Chart	41
5.3.1 Variabili	41
5.3.2 Stati e Funzionamento	42
5.4 Buttons Charts	44
5.4.1 Variabili	44
5.4.2 Stati e Funzionamento	44
5.5 Final Chart	46
5.6 Inputs & Outputs	47

INDICE

6 Test Simulink	48
6.1 Test Apertura e Chiusura Cancello	48
6.2 Test Errore Cancello	49
6.3 Test Rilevazione Ostacolo se Aperto	50
6.4 Test Rilevazione Ostacolo se In Chiusura	51
6.5 Test Rilevazione Ostacolo se Chiuso	52
6.6 Regolazione T_C e T_L	53
6.7 Regolazione T_C e T_L non nello stato Chiuso	55
6.8 Reset del timer durante rilevazione Ostacolo	56
7 Implementazione	58
7.1 Realizzazione del circuito	58
7.2 Codice Sorgente	59
Indice delle Figure	61

CAPITOLO 1

USER STORIES

1.1 US1 - Apertura Cancello

Descrizione

Come utente,
voglio premere il pulsante B1 quando il cancello è chiuso o in chiusura,
al fine di avviare la fase di apertura del cancello.

Criterio di Accettazione

Dato che il cancello è chiuso o in fase di chiusura,
quando premo il pulsante B1,
allora il cancello deve iniziare la fase di apertura.

1.2 US2 - Chiusura Cancello

Descrizione

Come utente,
voglio premere il pulsante B1 quando il cancello è in apertura o aperto,
al fine di avviare la fase di chiusura del cancello.

1. USER STORIES

Criterio di Accettazione

Dato che il cancello è aperto o in fase di apertura,
quando premo il pulsante B1,
allora il cancello deve iniziare la fase di chiusura.

1.3 US3 - Regolazione Tempo Chiusura Automatica

Descrizione

Come utente,
voglio regolare il tempo di chiusura automatica del cancello premendo il pulsante B2 quando il cancello è chiuso,
al fine di impostare dopo quanto tempo dall'apertura il cancello deve richiudersi.

Criterio di Accettazione #1

Dato che il cancello è chiuso,
quando premo il pulsante B2,
se il tempo di chiusura automatica è inferiore a 120 secondi,
allora il tempo di chiusura automatica aumenta di 10 secondi.

Criterio di Accettazione #2

Dato che il cancello è chiuso,
quando premo il pulsante B2,
se il tempo di chiusura automatica è a 120 secondi,
allora il tempo di chiusura automatica ritorna a 10 secondi.

1.4 US4 - Regolazione Tempo Lavoro

Descrizione

Come utente,
voglio regolare la durata delle fasi di apertura e chiusura del cancello premendo il pulsante B3 quando il cancello è chiuso,
al fine di impostare la durata delle fasi di apertura e chiusura del cancello.

1. USER STORIES

Criterio di Accettazione #1

Dato che il cancello è chiuso,
quando premo il pulsante B3,
se il tempo di lavoro è inferiore a 120 secondi,
allora il tempo di lavoro aumenta di 10 secondi.

Criterio di Accettazione #2

Dato che il cancello è chiuso,
quando premo il pulsante B3,
se il tempo di lavoro è 120 secondi,
allora il tempo di lavoro ritorna a 10 secondi.

1.5 US5 - Riapertura Automatica con Rilevazione Ostacolo

Descrizione

Come utente,
voglio che il cancello si riapra automaticamente se viene rilevata la presenza di un ostacolo durante la fase di chiusura,
in modo da evitare danni al cancello e garantire la sicurezza delle persone e degli oggetti presenti.

Criterio di Accettazione

Dato che il cancello è in fase di chiusura,
quando il sensore di presenza (P1) rileva un ostacolo,
allora il cancello si riapre automaticamente.

1.6 US6 - Gestione Sicura del Cancello in Presenza di Ostacoli

Descrizione

Come utente,
voglio che il dispositivo ignori le richieste di apertura o chiusura del cancello quando il sensore di presenza è attivo,
in modo da prevenire movimenti non sicuri del cancello in presenza di ostacoli o persone.

1. USER STORIES

Criterio di Accettazione #1

Dato che il sensore di presenza (P1) è attivo,
quando c'è una richiesta di apertura o chiusura del cancello,
allora il dispositivo non esegue l'azione richiesta.

Criterio di Accettazione #2

Dato il sensore di presenza (P1),
quando esso non rileva più alcun ostacolo,
allora il dispositivo è nuovamente pronto a ricevere e gestire le richieste di apertura o chiusura del cancello..

1.7 US7 - Gestione del sensore di chiusura per determinare lo stato del cancello

Descrizione

Come utente,
voglio che il dispositivo utilizzi il sensore di presenza (P2) come sensore di chiusura del cancello
in modo da determinare affidabilmente lo stato del cancello.

Criterio di Accettazione

Dato che il cancello è in fase di chiusura,
quando il sensore di presenza (P2) è attivo,
allora il cancello si considera chiuso completamente.

1.8 US8 - Errore in caso di malfunzionamento del sensore di chiusura

Descrizione

Come utente
voglio che il dispositivo entri in uno stato di errore se il sensore di chiusura (P2) non si attiva dopo il tempo di lavoro previsto durante la fase di chiusura del cancello,
in modo da essere avvisato in caso di malfunzionamento del sensore.

1. USER STORIES

Criterio di Accettazione

Dato che è in corso la fase di chiusura del cancello,
quando il sensore di chiusura (P2) non si attiva entro il tempo di lavoro previsto,
allora il dispositivo entra in uno stato di errore.

1.9 US9 - Avvio Chiusura Cancello senza Sensore Attivo

Descrizione

Come utente,
voglio che il dispositivo avvii la procedura di chiusura del cancello quando viene acceso, se il sensore di chiusura (P2) e il sensore di presenza (P1) non sono attivi,
in modo da garantire la chiusura corretta del cancello all'accensione.

Criterio di Accettazione

Dato che il dispositivo è acceso,
quando il sensore di chiusura (P2) e il sensore di presenza (P1) non sono attivi,
allora viene avviata la procedura di chiusura del cancello.

1.10 US10 - Indicazione del Cancello in Movimento

Descrizione

Come utente,
voglio che il LED giallo lampeggi mentre il cancello è in apertura o in chiusura,
al fine di avere una conferma visiva dello stato di movimento.

Criterio di Accettazione

Dato che il cancello è in fase di apertura o chiusura,
quando il cancello si muove,
allora il LED giallo lampeggia con una frequenza di 0,5 Hz.

1.11 US11 - Indicazione di Errore di Chiusura

Descrizione

Come utente,
voglio che il LED rosso si accenda se il cancello non si chiude entro 10 secondi dal completamento del tempo di lavoro,
al fine di essere notificato di uno stato di errore.

Criterio di Accettazione

Dato che il cancello è in fase di chiusura,
quando il cancello non si chiude entro 10 secondi dal completamento del tempo di lavoro,
allora il LED rosso si accende per notificare lo stato di errore.

1.12 US12 - Indicazione di Ostacolo

Descrizione

Come utente,
voglio che il LED verde lampeggi se un ostacolo è presente davanti al sensore P1 quando si richiede l'apertura o la chiusura,
al fine di essere notificato della presenza di un ostacolo.

Criterio di Accettazione

Dato che viene richiesta l'apertura o la chiusura del cancello,
quando un ostacolo è presente davanti al sensore P1,
allora il LED verde lampeggia con una frequenza di 1 Hz per 30 secondi.

1.13 US13 - Indicazione Cancello Chiuso

Descrizione

Come utente,
voglio che tutti i LED siano spenti quando il cancello è chiuso,
al fine di avere una conferma visiva che il cancello è completamente chiuso.

Criterio di Accettazione

Dato che la procedura di chiusura del cancello è attiva,
quando il cancello è completamente chiuso,
allora tutti i LED sono spenti.

1.14 US14 - Indicazione Cancello Aperto

Descrizione

Come utente,
voglio che tutti i LED siano accesi senza lampeggiare quando il cancello è aperto,
al fine di avere una conferma visiva che il cancello è completamente aperto.

Criterio di Accettazione

Dato che la procedura di apertura del cancello è attiva,
quando il cancello è completamente aperto,
allora tutti i LED sono accesi senza lampeggiare.

CAPITOLO 2

USE CASES

Per poter rappresentare le user stories sopra descritte, utilizziamo gli Use Case Diagrams, dei diagrammi che rappresentano le interazioni tra gli utenti e il sistema. In questo scenario, l'attore principale è l'utente che interagisce con il sistema del cancello automatico attraverso i pulsanti B1, B2 e B3.

2.1 Apertura Cancello [US1-US10-US14]

L'utente ha la possibilità, in prossimità del cancello, di richiederne l'apertura premendo il pulsante B1. Quando il sistema rileva che il pulsante B1 è stato premuto e il cancello è nelle condizioni specificate (chiuso o in chiusura), avvia il processo di apertura del cancello. Il dispositivo inoltre fornisce un feedback visivo ottenuto dall'attivazione di un segnale luminoso, identificato dal lampeggiare di un LED giallo con una frequenza 0.5 Hz. Il dispositivo, inoltre, permette di verificare la completa apertura del cancello tramite l'accensione di tutti i LED: giallo, rosso e verde (figura 2.1).

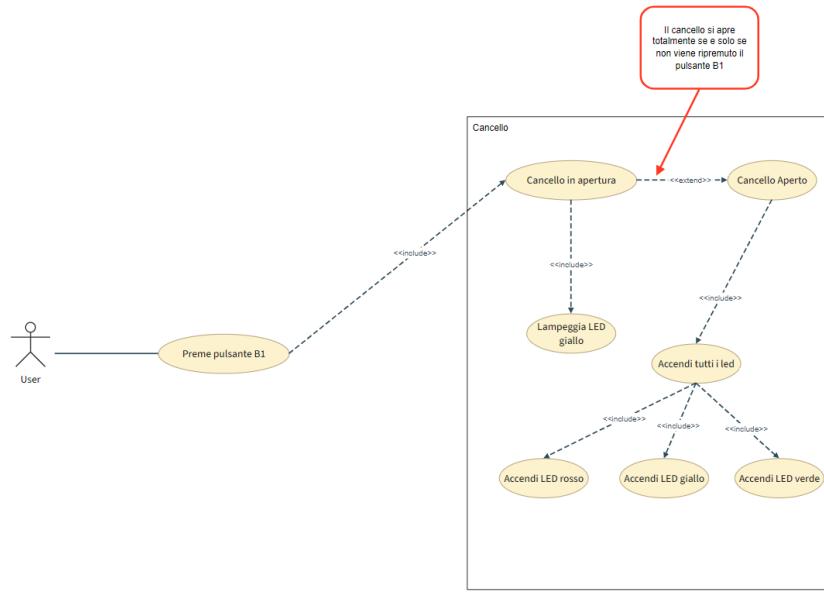


Figure 2.1: Apertura Cancello

2.2 Chiusura Cancello [US2-US10-US13]

L'utente ha la possibilità, in prossimità del cancello, di richiederne la chiusura premendo il pulsante B1. Quando il sistema rileva che il pulsante B1 è stato premuto e il cancello è nelle condizioni specificate (aperto o in apertura), avvia il processo di chiusura del cancello. Il dispositivo inoltre fornisce un feedback visivo ottenuto dall'attivazione di un segnale luminoso, identificato dal lampeggiare di un LED giallo con una frequenza 0.5 Hz. Il dispositivo, inoltre, permette di verificare la completa chiusura del cancello tramite lo spegnimento di tutti i LED (figura 2.2).

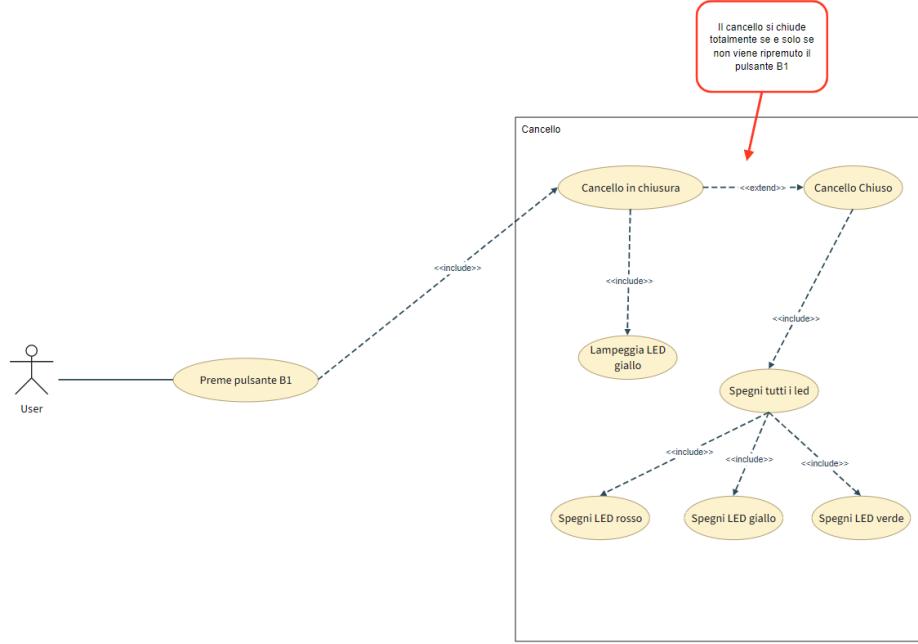


Figure 2.2: Chiusura Cancello

2.3 Regolazione Tempo di Chiusura Automatica [US3]

L'utente ha la possibilità di regolare il tempo di chiusura automatica del cancello andando a determinare quanto esso debba rimanere aperto prima di chiudersi automaticamente. Quando il cancello è chiuso, l'utente preme il pulsante B2 per effettuare la regolazione. Se il tempo di chiusura automatica è inferiore a 120 secondi, ogni pressione del pulsante B2 aumenta il tempo di 10 secondi. Se il tempo è già a 120 secondi, premendo nuovamente B2 il tempo viene riportato a 10 secondi (figura 2.3).

2.4 Regolazione Tempo di Lavoro [US4]

L'utente ha la possibilità di richiedere la regolazione della durata delle fasi di apertura e chiusura del cancello premendo l'apposito pulsante (B3) solo quando il cancello è chiuso. Questa azione è essenziale per impostare la durata delle due fasi del cancello. Ogni pressione del pulsante incrementa la durata di 10 secondi e, se il tempo di lavoro è al suo massimo (120 secondi), esso ritorna a 10 secondi (figura 2.3).

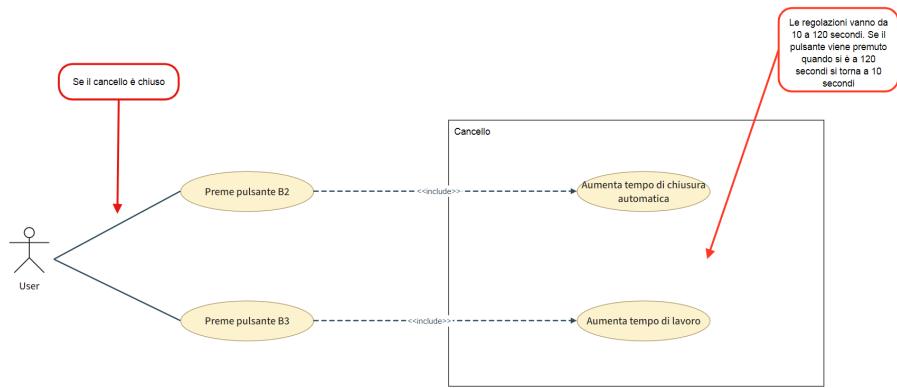


Figure 2.3: Regolazioni

2.5 Riapertura Automatica con Rilevazione Ostacolo [US5-US14]

Il dispositivo, in caso di rilevamento, tramite il sensore di presenza (P1), di un ostacolo durante la fase di chiusura, può effettuare la riapertura automatica del cancello. Questa azione è essenziale per evitare danni al cancello e garantire la sicurezza delle persone e degli oggetti presenti. Il dispositivo fornisce un feedback in caso di apertura completa del cancello, dato dall'accensione di tutti i LED (figura 2.4).

2.6 Gestione Richieste in Presenza di Ostacoli [US6-US12]

Il dispositivo, nel caso in cui il sensore di presenza (P1) sia attivo, ignora le richieste di apertura o chiusura del cancello. Questa azione è essenziale per prevenire movimenti non sicuri del cancello in presenza di ostacoli o persone. Il dispositivo fornisce un feedback visivo in caso di presenza di un ostacolo, dato dal lampeggio del LED verde con frequenza di 1 Hz per un tempo di 30 secondi o finché l'ostacolo non viene più rilevato (figura 2.4).

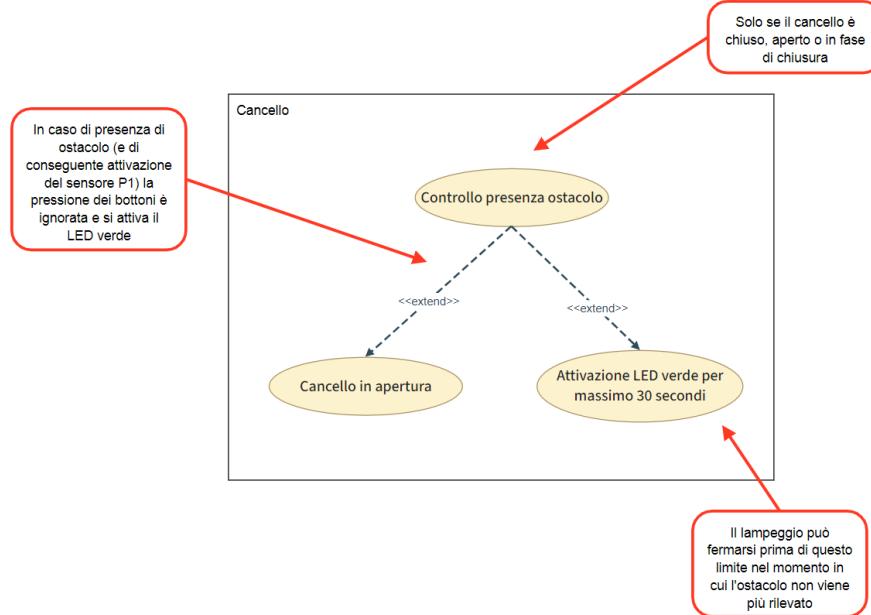


Figure 2.4: Controllo Ostacolo e Gestione Richieste

2.7 Determinazione e Comunicazione Stato Cancello [US7-US13]

Il dispositivo, tramite il sensore di presenza (P2), decreta lo stato di chiusura completa del cancello. Questa azione è essenziale per determinare correttamente lo stato del cancello che si considera chiuso quando il sensore è attivo (figura 2.5). Il dispositivo fornisce un feedback visivo in caso il cancello risulti completamente chiuso, dato dallo spegnimento di tutti i LED.

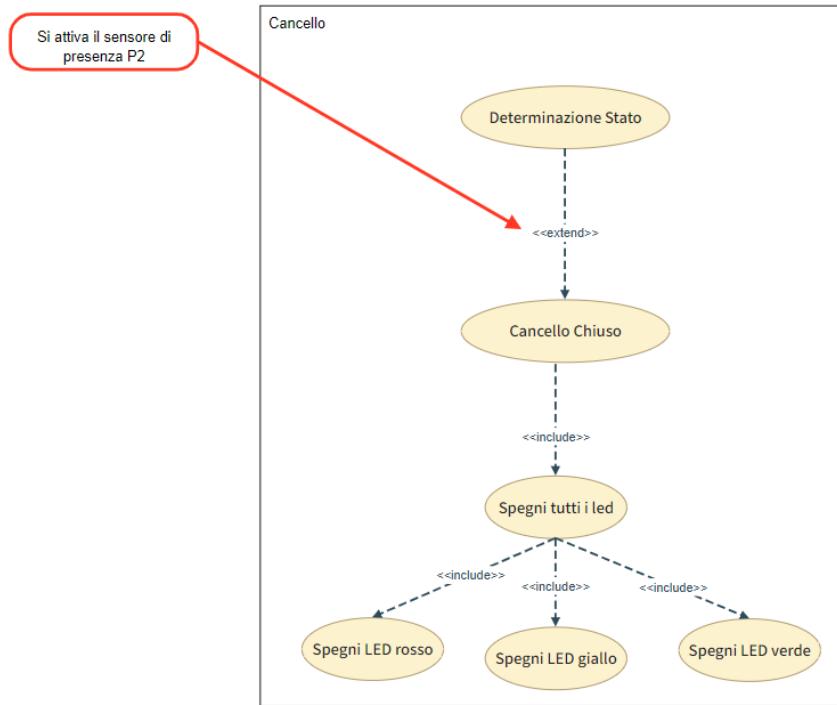


Figure 2.5: Determinazione Stato

2.8 Gestione dello Stato di Errore [US8-US11]

Il dispositivo entra in uno stato di errore nel caso in cui il sensore di presenza (P2) non si attivi dopo il tempo di lavoro previsto durante la fase di chiusura del cancello. Questa azione è essenziale per far sì che l'utente venga avvisato in caso di malfunzionamento del sensore. Il dispositivo fornisce un feedback visivo dello stato di errore, dato dall'accensione del LED rosso nel caso in cui il cancello non si chiuda entro 10 secondi dal completamento del tempo di lavoro (figura 2.6).

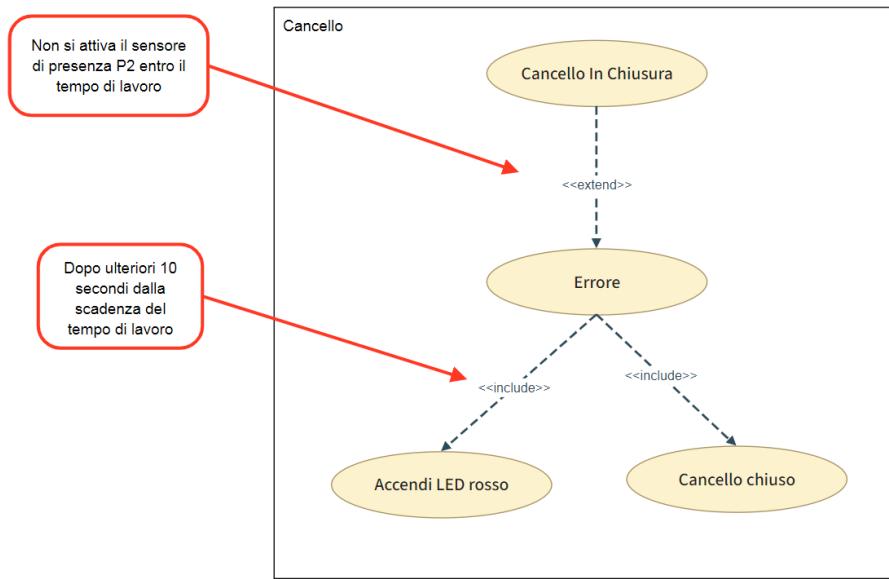


Figure 2.6: Stato di Errore

2.9 Chiusura automatica all'accensione [US9]

L'utente ha la possibilità di richiedere che il dispositivo avvii la procedura di chiusura del cancello automatico quando il dispositivo viene acceso per la prima volta, solo se i due sensori di presenza P1 e P2 non sono attivi. Questa azione è essenziale per garantire la corretta chiusura del cancello all'accensione del dispositivo.



Figure 2.7: Chiusura Automatica

2. USE CASES

2.10 General Use Case

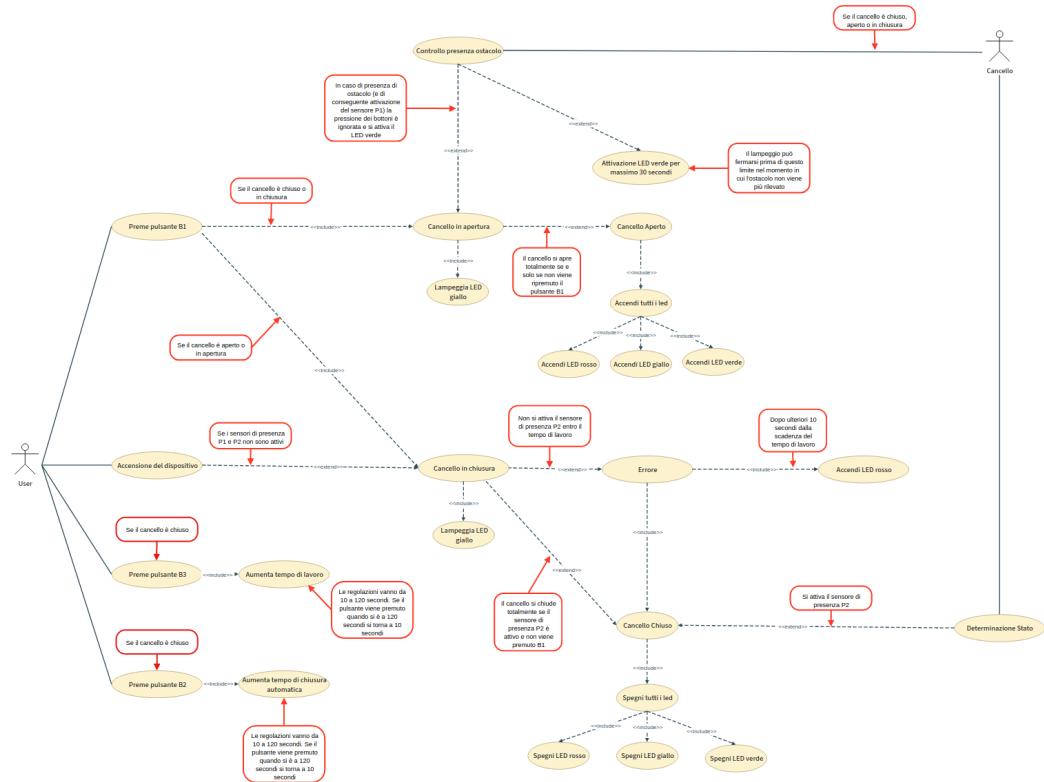


Figure 2.8: Use Cases Generale

CAPITOLO 3

ACTIVITY DIAGRAMS

3.1 Apertura e Chiusura Cancello [Scenario 1]

Questo scenario, illustrato in figura 3.1, descrive in sequenza le azioni che l'utente compie per aprire e chiudere il cancello, dalle fasi iniziali di richiesta tramite il pulsante B1, fino al feedback visivo che conferma l'operazione completata. Di seguito vengono presentati i flussi di azioni associati allo scenario corrente.

Apertura del Cancello:

1. L'utente decide di aprire il cancello e si avvicina ad esso;
2. Per avviare il processo di apertura, l'utente preme il pulsante B1;
3. Il sistema rileva che il pulsante B1 è stato premuto;
4. Il sistema verifica che il cancello sia chiuso o in chiusura;
5. Il sistema avvia il processo di apertura del cancello;
6. Durante l'apertura, il dispositivo fornisce un feedback visivo attivando il lampeggiamento del LED giallo con frequenza 0.5 Hz;
7. Una volta completata l'apertura, tutti i LED (giallo, rosso e verde) si accendono staticamente per indicare la completa apertura del cancello.

Chiusura del Cancello:

3. ACTIVITY DIAGRAMS

1. L'utente decide di chiudere il cancello e si avvicina ad esso;
2. Per avviare il processo di chiusura, l'utente preme il pulsante B1;
3. Il sistema rileva che il pulsante B1 è stato premuto;
4. Il sistema verifica che il cancello sia aperto o in apertura;
5. Il sistema avvia il processo di chiusura del cancello;
6. Durante la chiusura, il dispositivo fornisce un feedback visivo attivando il lampeggiamento del LED giallo con frequenza 0.5 Hz;
7. Una volta completata la chiusura, tutti i LED (giallo, rosso e verde) si spengono per indicare la completa chiusura del cancello.

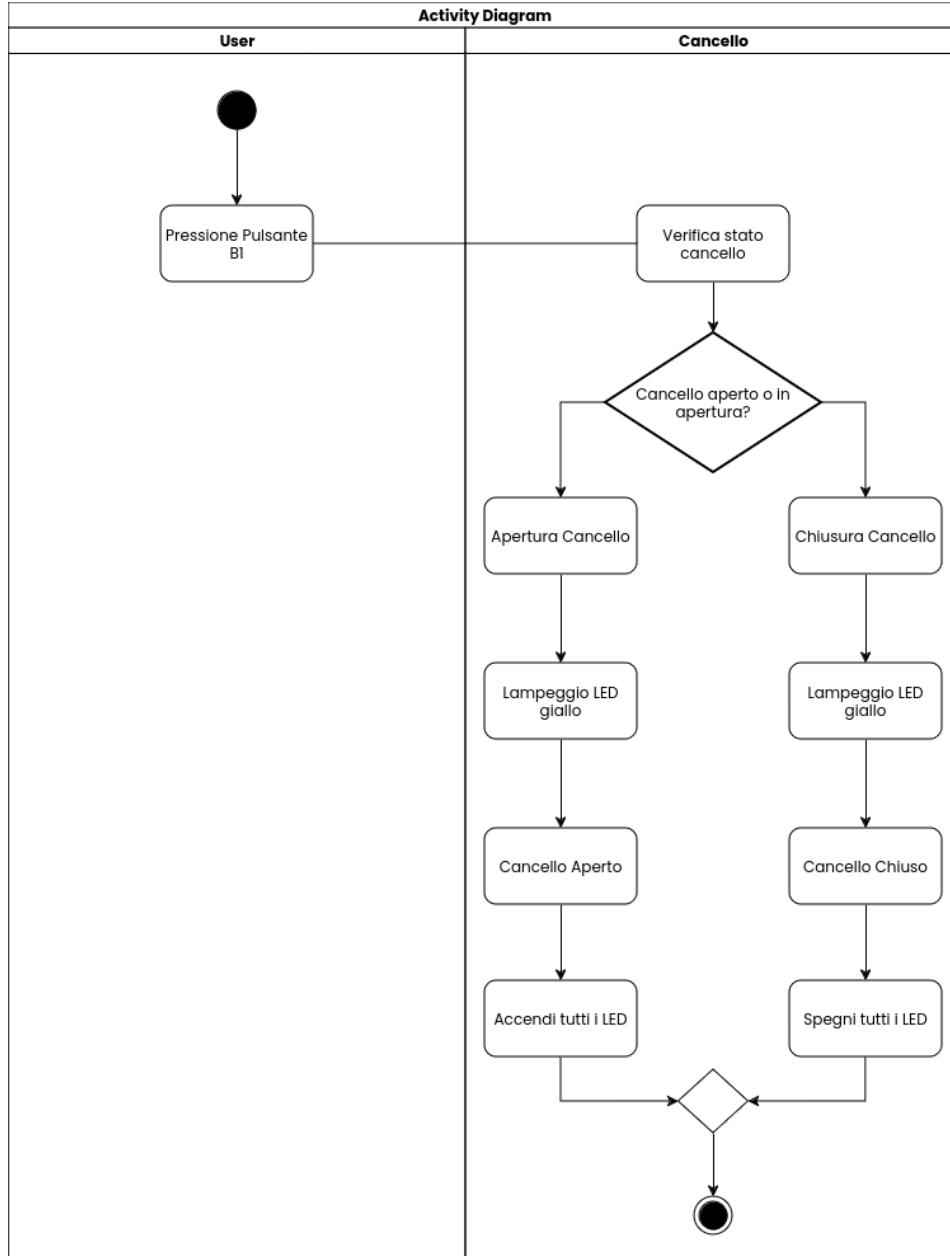


Figure 3.1: Scenario 1

3.2 Regolazioni [Scenario 2]

Questo scenario è suddiviso in due parti, entrambe ricostruite in figura 3.2, poiché i diagrammi di entrambe le parti sono identici sotto il profilo funzionale, è stato preferito l'inserimento di un singolo diagramma esemplificativo per entrambi.

La prima parte dello scenario descrive sequenzialmente le azioni che l'utente compie per regolare il tempo di chiusura automatica del cancello. Di seguito viene presentato il

3. ACTIVITY DIAGRAMS

flusso di azioni associato alla sezione corrente.

Regolazione Tempo di Chiusura Automatica:

1. L'utente decide di regolare il tempo di chiusura automatica del cancello;
2. L'utente si avvicina al cancello chiuso;
3. Per avviare il processo di regolazione, l'utente preme il pulsante B2;
4. Il sistema rileva che il pulsante B2 è stato premuto;
5. Il tempo di chiusura automatica viene così gestito:
 - (a) Se il tempo di chiusura automatica è inferiore a 120 secondi, ogni pressione del pulsante B2 aumenta il tempo di 10 secondi;
 - (b) Se il tempo di chiusura automatica è già a 120 secondi, premendo nuovamente B2 il tempo viene riportato a 10 secondi.

La seconda parte descrive, invece, le azioni che l'utente compie per regolare la durata delle fasi di apertura e chiusura del cancello.

Regolazione Tempo di Lavoro:

1. L'utente decide di regolare la durata delle fasi di apertura e chiusura del cancello;
2. L'utente si avvicina al cancello chiuso;
3. Per avviare il processo di regolazione, l'utente preme il pulsante B3;
4. Il sistema rileva che il pulsante B3 è stato premuto;
5. Il tempo per la regolazione delle fasi viene così gestito:
 - (a) Se il tempo di chiusura automatica è inferiore a 120 secondi, ogni pressione del pulsante B3 aumenta il tempo di 10 secondi;
 - (b) Se il tempo di chiusura automatica è già a 120 secondi, premendo nuovamente B3 il tempo viene riportato a 10 secondi.

3. ACTIVITY DIAGRAMS

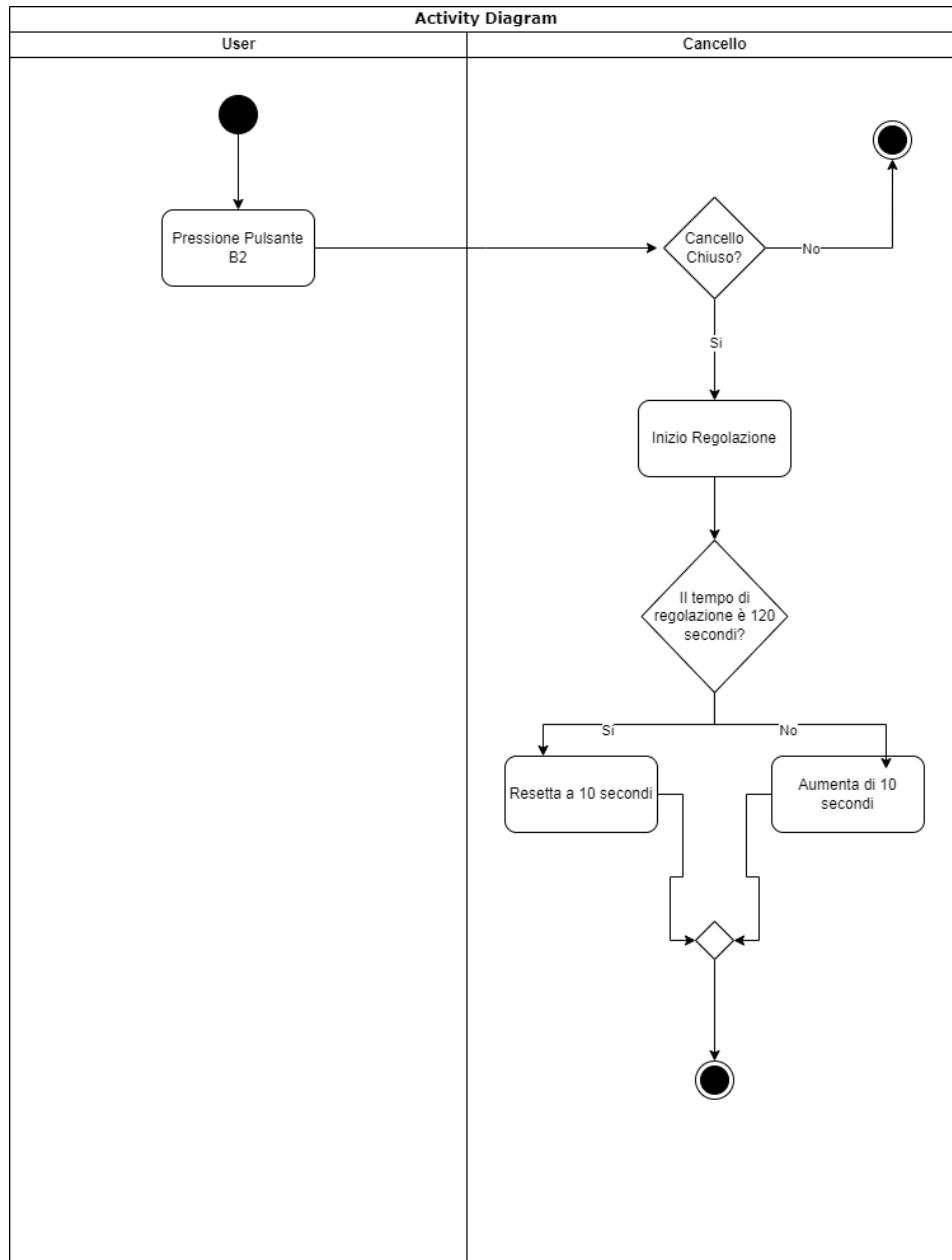


Figure 3.2: Scenario 2

3.3 Gestione Stato e Ostacoli [Scenario 3]

Lo scenario attuale, ritratto in figura 3.3, è anch'esso suddiviso in due sezioni differenti.

La prima parte descrive in modo consecutivo le azioni relative alla riapertura automatica del cancello, attuata dal dispositivo, in presenza di un ostacolo. Di seguito è presentato il flusso di azioni associato allo scenario corrente.

3. ACTIVITY DIAGRAMS

Riapertura Automatica con Rilevazione Ostacolo:

1. Il sistema rileva un ostacolo tramite il sensore di presenza (P1) durante la fase di chiusura del cancello;
2. Il sistema avvia la riapertura automatica del cancello per evitare danni e garantire la sicurezza;
3. Il dispositivo fornisce un feedback visivo in caso di apertura completa del cancello, accendendo tutti i LED (giallo, rosso e verde);
4. Se il sistema non rileva alcun ostacolo procederà con la chiusura del cancello e l'attivazione del sensore di presenza P2.

La seconda parte descrive, invece, le azioni compiute dal sistema per gestire l'apertura o la chiusura del cancello in presenza di ostacoli. Di seguito viene presentato il flusso di azioni associato allo scenario corrente.

Gestione Richieste in Presenza di Ostacoli:

1. Il sistema rileva la presenza di un ostacolo tramite il sensore P1;
2. Il sistema ignora le richieste di apertura o chiusura del cancello per prevenire movimenti non sicuri;
3. Il dispositivo fornisce un feedback visivo della presenza di un ostacolo, facendo lampeggiare il LED verde con una frequenza di 1 Hz per un massimo di 30 secondi o finché l'ostacolo non viene più rilevato.

3. ACTIVITY DIAGRAMS

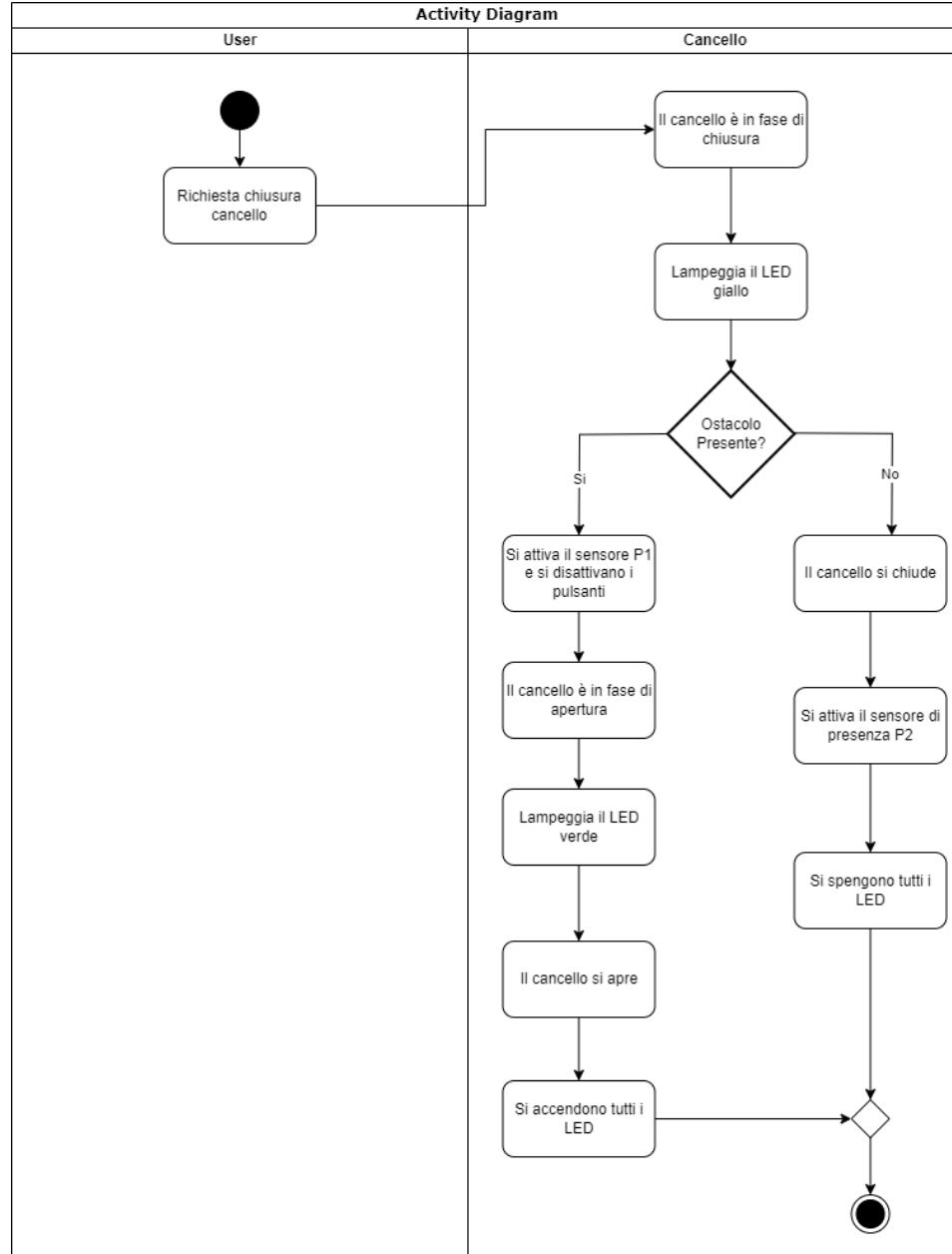


Figure 3.3: Scenario 3

3.4 Stato di Errore [Scenario 4]

Questo scenario, riprodotto in figura 3.4, descrive in modo seriale le azioni che portano il dispositivo in uno stato di errore. Di seguito viene presentato il flusso di azioni associato allo scenario corrente.

Stato di Errore con Rilevazione Malfunzionamento del Sensore:

1. L'utente richiede la chiusura del cancello tramite il pulsante B1;

3. ACTIVITY DIAGRAMS

2. Il sistema rileva che il sensore di presenza (P2) non si è attivato dopo il tempo di lavoro previsto durante la fase di chiusura del cancello;
3. Il sistema entra in uno stato di errore per avvisare l'utente;
4. Il dispositivo fornisce un feedback visivo dello stato di errore accendendo il LED rosso se il cancello non si chiude entro 10 secondi dalla scadenza del tempo di lavoro.

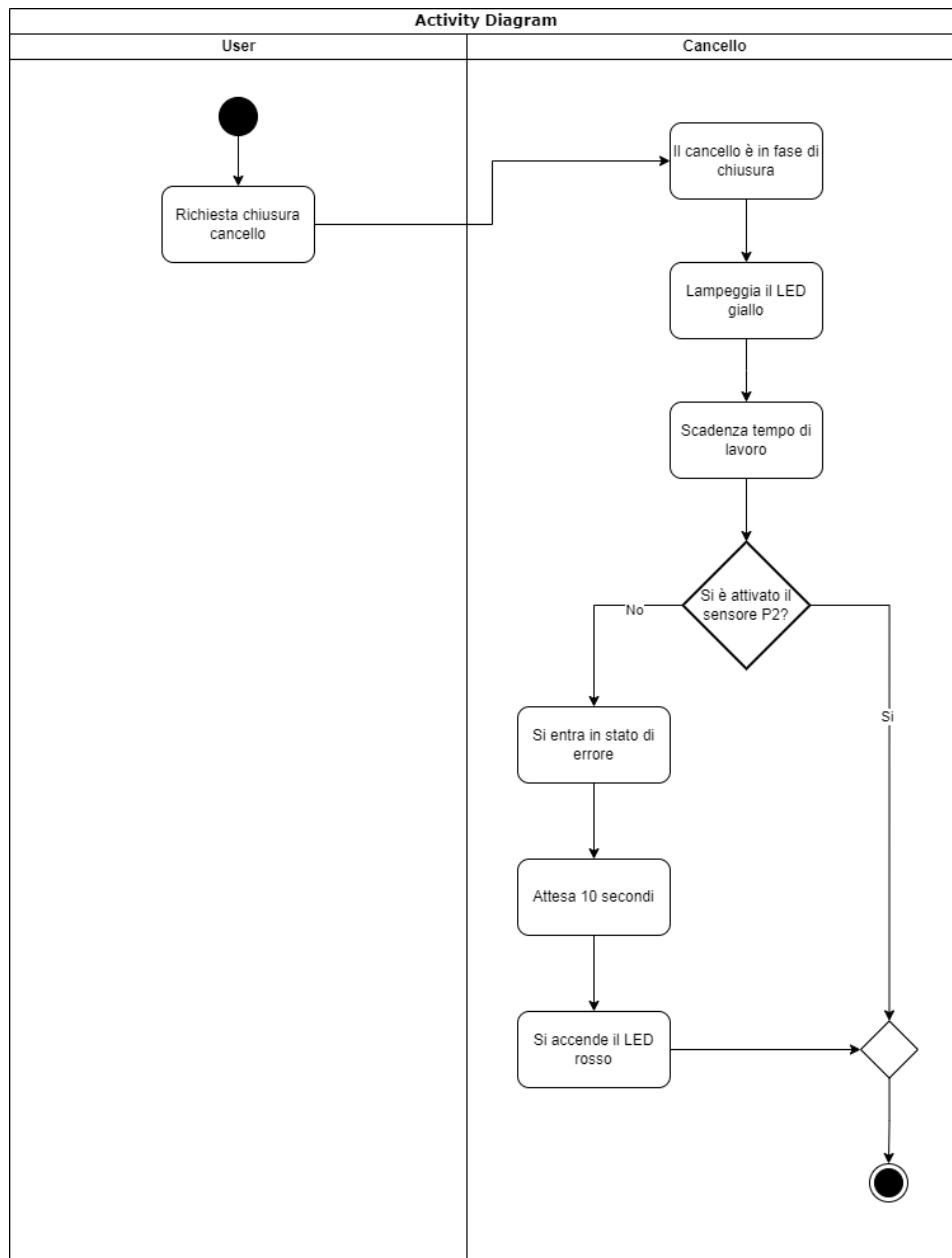


Figure 3.4: Scenario 4

3.5 Chiusura Automatica all'accensione [Scenario 5]

Questo scenario, rappresentato in figura 3.5, descrive in successione le azioni necessarie per la chiusura automatica del cancello all'accensione del dispositivo. Nell'elenco sottostante è presentato il flusso di azioni associato allo scenario corrente.

Chiusura Automatica all'Accensione del Dispositivo:

1. L'utente accende il dispositivo per la prima volta;
2. Il sistema verifica che i sensori di presenza P1 e P2 non siano attivi;
3. Il sistema avvia la procedura di chiusura del cancello;
4. Il dispositivo garantisce la corretta chiusura del cancello all'accensione.

3. ACTIVITY DIAGRAMS

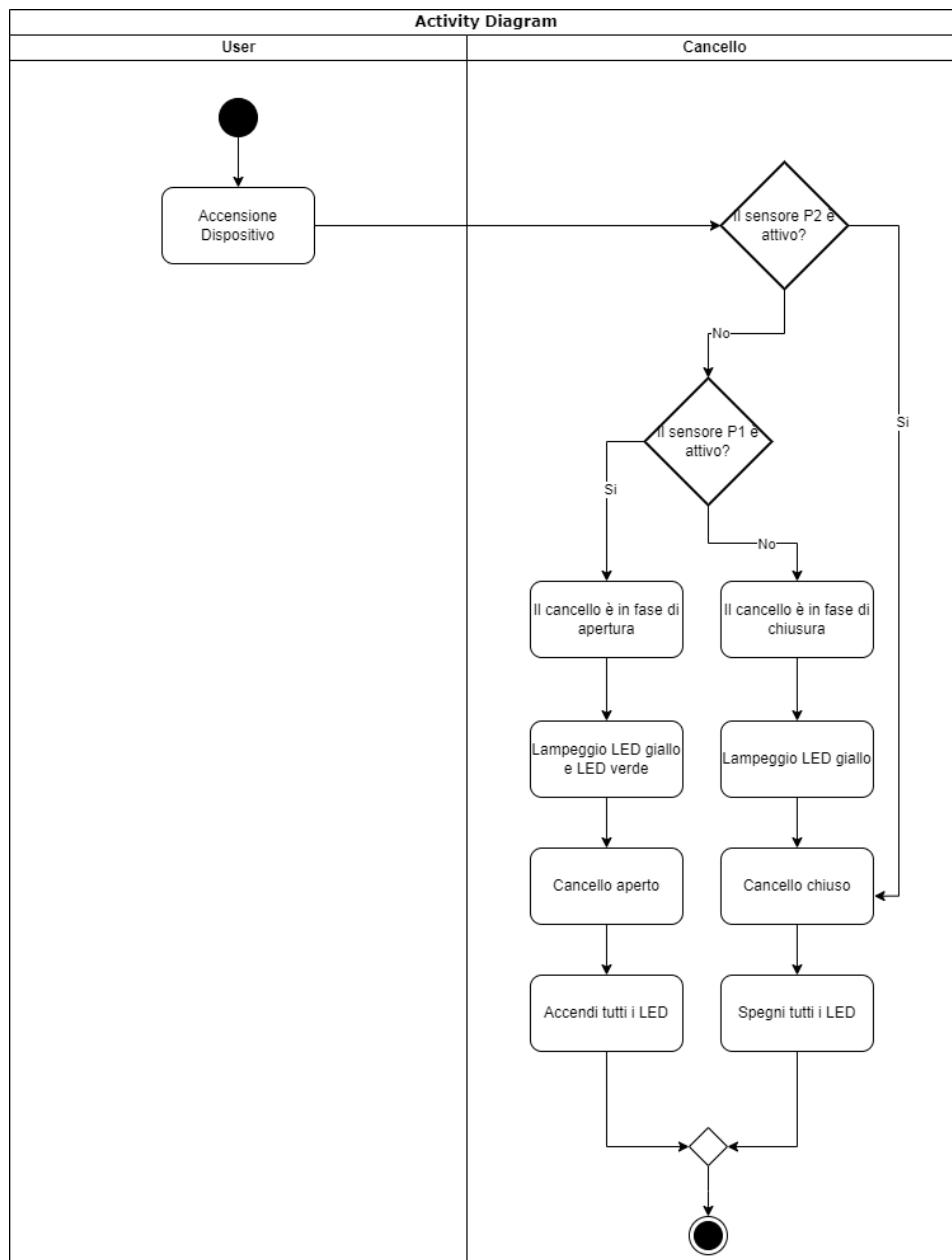


Figure 3.5: Scenario 5

CAPITOLO 4

STATE DIAGRAM

4.1 Input, Output e Stati

Lo **state diagram** è una rappresentazione grafica che mostra i differenti stati in cui un sistema può trovarsi e le transizioni tra questi ultimi, in funzione di variabili esterne.

Gli input del diagramma principale sono:

- **B1**: bottone per la chiusura e apertura del cancello
- **B2**: bottone per la regolazione del tempo di chiusura automatica del cancello
- **B3**: bottone per la regolazione del tempo di lavoro del cancello
- **P1**: sensore di presenza per la rilevazione di ostacoli
- **P2**: sensore di presenza per la rilevazione della chiusura completa del cancello
- **T_L**: timer relativo al tempo di lavoro
- **T_C**: timer relativo al tempo di chiusura automatica

Gli output del diagramma sono:

- **Led Green**
- **Led Yellow**
- **Led Red**

Gli stati del diagramma sono:

- **Inattivo:** è lo stato iniziale del sistema, disponibile appena dopo l'accensione, durante il quale viene controllata l'attivazione dei due sensori di presenza **P1** e **P2** per garantire una corretta chiusura del cancello dopo l'accensione dello stesso;
- **Chiusura:** macrostato esterno che raggruppa al suo interno:
 - **In Chiusura:** raffigura lo stato secondo cui il cancello si sta chiudendo, durante il quale si controlla l'eventuale presenza di ostacoli. Si controlla anche che il cancello si chiuda entro il tempo di lavoro (T_L) prestabilito;
 - **Chiuso:** rappresenta lo stato per cui il cancello risulta completamente chiuso;
 - **Errore:** macrostato relativo alla gestione degli errori, suddiviso in:
 - * **In Errore:** simboleggia lo stato di errore che si verifica nel momento in cui il cancello risulta non chiuso dopo il tempo prestabilito (T_L);
 - * **LED Rosso Errore:** simboleggia lo stato in cui il dispositivo entra quando il cancello risulta non chiuso dopo il tempo prestabilito (T_L) più 10 secondi di attesa. È connotato dall'accensione del LED rosso.
- **Apertura:** macrostato contenente i seguenti stati:
 - **In Apertura:** delinea lo stato di apertura in corso del cancello;
 - **Aperto:** rappresenta lo stato in cui il cancello risulta completamente aperto;
- **Ostacolo:** macrostato relativo alla gestione degli ostacoli. Gli stati interni sono:
 - **Fermo:** rappresenta lo stato di attesa in cui si trova il cancello prima dell'attivazione del pulsante **B1** e/o dopo lo scadere dei 30 secondi di *blinking* del LED verde o dopo la rimozione dell'ostacolo;
 - **Ostacolo presente:** descrive lo stato durante il quale il cancello è aperto/chiuso/inattivo ma è stata rilevata la presenza di un ostacolo dal sensore **P1** e, dunque, non vi è possibilità di movimento;
- **Regolazioni:** macrostato contenente gli stati addetti alla regolazione dei tempi di chiusura (T_C) e di lavoro (T_L). Contiene:
 - **Tempo Lavoro:** imposta il tempo di lavoro iniziale ad una quantità pari a 10 secondi;
 - **Aggiunta Tempo Lavoro:** incrementa il tempo di lavoro di 10 secondi ad ogni interazione fino al raggiungimento dei 120 secondi, dopo i quali riparte da 10;
 - **Tempo Chiusura:** imposta il tempo di chiusura iniziale ad una quantità pari a 10 secondi;

- **Aggiunta Tempo Chiusura:** incrementa il tempo di chiusura di 10 secondi ad ogni interazione fino al raggiungimento dei 120 secondi, dopo i quali riparte da 10.

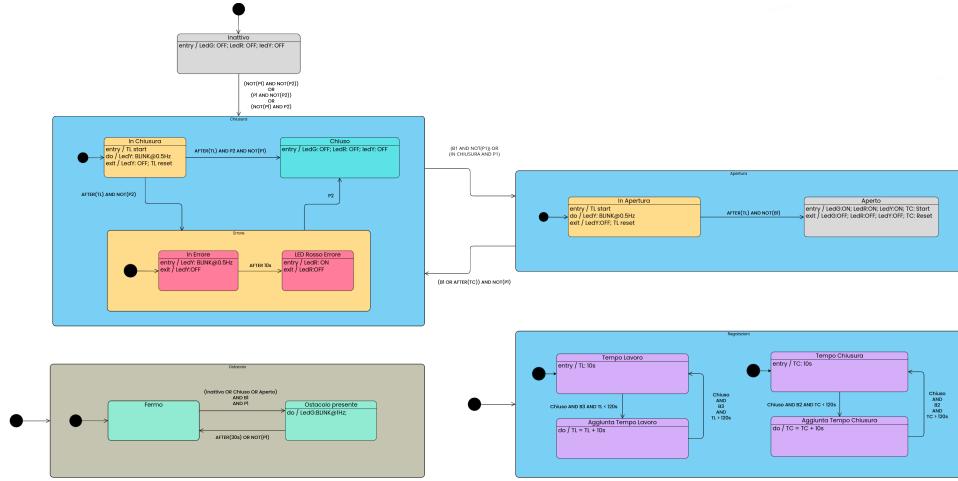


Figure 4.1: State Diagram

4.2 Logica di Funzionamento

4.2.1 Stato Inattivo

All'avvio, il sistema si trova nello stato iniziale **Inattivo**, durante il quale si controlla lo stato dei due sensori di presenza **P1** e **P2**, al fine di ritornare allo stato iniziale desiderato di **Chiuso**.

Le transizioni in uscita da esso prevedono, per progettazione, il passaggio nel macrostato *Chiusura* e sono:

- se entrambi i sensori sono disattivati si procede alla chiusura passando allo stato **In Chiusura**;
- se è attivo il sensore di presenza **P1**, ma non il sensore **P2**, si passa per lo stato **In Chiusura** per poi giungere subito allo stato **In Apertura**, contenuto nel macrostato *Apertura*;
- se non è attivo il sensore di presenza **P1**, ma lo è **P2**, si giunge allo stato **Chiuso** per segnalarne la corretta chiusura;

4.2.2 Macrostate *Chiusura*

All'interno del macrostato in esame sono presenti diverse strutture, in particolare due stati interni (**In Chiusura** e **Chiuso**) e un macrostato sottostante **Errore** (quest'ultimo analizzato nel paragrafo 4.2.3).

La transizione in uscita da questo macrostato è la seguente:

- *Chiusura* → *Apertura*: se l'utente preme il bottone **B1** senza che vi sia la rilevazione di ostacoli dal sensore **P1** o se viene rilevato un ostacolo senza la pressione del pulsante mentre si è nello stato **In Chiusura**, si passa al macrostato **Apertura**;

Di seguito sono trattati gli stati interni.

Stato *In Chiusura*

In questo stato viene innanzitutto iniziato il conteggio del timer relativo al tempo di lavoro (**T_L**). Durante la persistenza dello stato, viene inoltre attivato il *blink* del LED giallo ad una frequenza pari a 0.5Hz per segnalare il movimento del cancello. All'uscita dallo stato, il LED giallo viene spento e il timer di **T_L** viene resettato.

Le transizioni in uscita da questo stato sono:

- *In Chiusura* → *Chiuso*: se termina il tempo di lavoro (**T_L**) e si attiva il sensore di presenza **P2** e non vi è rilevazione di ostacoli dal sensore **P1**, allora si passa allo stato **Chiuso**;
- *In Chiusura* → *Errore*: se dovesse scadere il tempo di lavoro (**T_L**), ma senza alcuna attivazione del sensore di presenza **P2**, si passerebbe al macrostato di **Errore**.

Stato *Chiuso*

Questo stato interno rappresenta lo stato di chiusura avvenuta del cancello, rilevata dall'attivazione del sensore **P2**. Non vi sono transizioni in uscita e l'unica azione effettuata da questo stato è quella di spegnere tutti i LED.

4.2.3 Macrostate *Errore*

Nel macrostato attuale, contenuto nel macrostato esterno **Chiusura**, ci si occupa della gestione dello stato di errore, nello specifico sono presenti due stati interni (**In Errore** e **LED Rosso Errore**)

L'unica transizione in uscita è:

- *Errore* → *Chiuso*: se il sensore **P2** rileva l'avvenuta chiusura del cancello è possibile uscire dallo stato di errore.

Di seguito sono trattati gli stati interni.

Stato *In Errore*

In questo stato è ancora presente il *blinking* del LED giallo ad una frequenza di 0.5Hz, il quale viene spento soltanto all'uscita. È possibile entrare nel poc'anzi citato stato nel momento in cui scade la durata del tempo di lavoro (T_L) e non vi è ancora alcuna segnalazione da parte del sensore di chiusura **P2**.

La transizione in uscita è:

- *In Errore* → **LED Rosso Errore**: se dopo la scadenza del tempo di lavoro (T_L) e una successiva attesa di 10 secondi, non è stata ancora segnalata la chiusura dal sensore **P2**, allora si entra nello stato **LED Rosso Errore**.

Stato *LED Rosso Errore*

Quando si entra in questo stato avviene l'accensione del LED rosso, a indicare che sono trascorsi 10 secondi dalla scadenza del tempo di lavoro (T_L) e non è ancora pervenuta nessuna segnalazione da parte del sensore di chiusura (**P2**). Non sono presenti transizioni in uscita da questo stato.

4.2.4 Macrostate *Apertura*

Nel macrostato di cui si sta trattando ora, è possibile osservare l'esistenza di due stati interni (**In Apertura** e **Aperto**).

L'unica transizione in uscita è:

- *Apertura* → **Chiusura**: se l'utente preme il bottone **B1** o se scade il tempo di chiusura (T_C) senza che vi sia la rilevazione di un ostacolo da parte del sensore **P1**, allora si entra nel macrostato **Chiusura**.

Di seguito sono trattati gli stati interni.

Stato *In Apertura*

In questo stato viene dapprima avviato il conteggio del timer relativo al tempo di lavoro (T_L). Durante la persistenza dello stato, viene inoltre attivato il *blink* del LED giallo ad una frequenza pari a 0.5Hz per segnalare il movimento del cancello. All'uscita dallo stato, viene spento il LED giallo e si resetta il timer T_L .

La transizione in uscita da questo stato è la seguente:

- *In Apertura* → **Aperto**: se dovesse scadere il tempo di lavoro (T_L) senza alcuna pressione del bottone **B1**, si entrerebbe nello stato **Aperto**.

Stato *Aperto*

Nello stato in essere vengono accesi staticamente tutti i LED per segnalare la corretta apertura del cancello ed inoltre viene attivato anche il timer relativo alla chiusura (T_C). Inoltre, non sono presenti transizioni in uscita dallo stato, ma in uscita dallo stesso vengono spenti tutti i LED e il timer del tempo di chiusura (T_C) viene resettato.

4.2.5 Macrostate *Ostacolo*

In questo macrostato ci sono due stati interni (**Fermo** e **Ostacolo**), oltretutto non sono presenti transizioni in uscita.

Di seguito sono trattati gli stati interni.

Stato *Fermo*

Questo stato indica che il sistema attende la pressione del pulsante **B1** da parte dell'utente. Ci si trova in questo stato quando o il cancello è inattivo/chiuso/aperto o dopo che sono trascorsi 30 secondi di lampeggio del LED verde o dopo la rimozione dell'ostacolo dalla fotocellula **P1**.

La transizione in uscita è:

- *Fermo* → *Ostacolo presente*: se il dispositivo si trova nello stato *Inattivo* o *Chiuso* o *Aperto* e l'utente preme il bottone **B1** in presenza di ostacolo, rilevato dal sensore **P1**, si passa allo stato **Ostacolo presente**.

Stato *Ostacolo presente*

Questo stato si occupa della gestione del LED verde, attivandone il *blinking* ad una frequenza di 1Hz fintantoché l'ostacolo è presente dinanzi alla fotocellula **P1** per un massimo di 30 secondi.

La transizione in uscita è:

- *Ostacolo presente* → *Fermo*: si passa allo stato **Fermo** una volta passati i 30 secondi di *blinking* del LED verde o nel momento in cui l'ostacolo non viene più rilevato.

4.2.6 Macrostate *Regolazioni*

In questo macrostato l'utente può regolare sia il tempo di chiusura (T_C) che il tempo di lavoro (T_L) tramite, rispettivamente, la pressione dei bottoni **B2** e **B3**, modificandone la durata in un range che spazia dai 10 secondi ai 120 secondi. Non vi sono transizioni in uscita e la pressione dei due bottoni **B2** e **B3** è ignorata in tutti gli altri stati.

Di seguito sono analizzati gli stati interni.

Tempo di Chiusura

- **Stato *Tempo Chiusura***

L'azione principale di questo stato è impostare il tempo di chiusura (T_C) a 10 secondi.

La transizione in uscita è:

- *Tempo Chiusura* → *Aggiunta Tempo Chiusura*: se il cancello si trova nello stato **Chiuso**, l'utente preme il bottone **B2** e il tempo di chiusura (T_C) attuale è inferiore a 120 secondi, allora si entra nello stato **Aggiunta Tempo Chiusura**.

- **Stato *Aggiunta Tempo Chiusura***

In questo stato si incrementa, ad ogni iterazione, la durata del tempo di chiusura (T_C) di 10 secondi. La transizione in uscita è:

- *Aggiunta Tempo Chiusura* → *Tempo Chiusura*: se il cancello è **Chiuso**, l'utente preme il bottone **B2** e il tempo di chiusura (T_C) attuale è uguale a 120 secondi, anziché aumentare, esso viene ripristinato a 10 secondi entrando nello stato **Tempo Chiusura**.

Tempo di Lavoro

- **Stato *Tempo Lavoro***

L'azione principale di questo stato è impostare il tempo di lavoro (T_L) a 10 secondi.

La transizione in uscita è:

- *Tempo Lavoro* → *Aggiunta Tempo Lavoro*: se il cancello si trova nello stato **Chiuso**, l'utente preme il bottone **B3** e il tempo di lavoro (T_L) attuale è inferiore a 120 secondi, allora si entra nello stato **Aggiunta Tempo Chiusura**.

- **Stato *Aggiunta Tempo Lavoro***

In questo stato si incrementa, ad ogni iterazione, la durata del tempo di lavoro (T_L) di 10 secondi. La transizione in uscita è:

- *Aggiunta Tempo Lavoro* → *Tempo Lavoro*: se il cancello si trova nello stato **Chiuso**, l'utente preme il bottone **B3** e il tempo di lavoro (T_L) attuale è uguale a 120 secondi, allora si entra nello stato **Tempo Lavoro**.

CAPITOLO 5

MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

In questo capitolo esploreremo l'ambiente Simulink di MATLAB, uno strumento avanzato per la modellazione, la simulazione e l'analisi di sistemi dinamici. Simulink consente di creare modelli grafici dei sistemi utilizzando un'interfaccia intuitiva a blocchi, facilitando la rappresentazione e la comprensione delle interazioni tra i vari componenti del sistema. Attraverso esempi pratici e casi di studio, illustreremo come costruire modelli accurati, simulare il comportamento del sistema e analizzare i risultati per ottimizzare le prestazioni. Particolare attenzione sarà dedicata a Stateflow, un modulo di Simulink specializzato nella gestione di logiche basate su stati, condizioni decisionali e comportamenti che cambiano nel tempo.

5.1 Gate Chart

In questa sezione esamineremo l'implementazione del modulo principale **Cancello**, che si occupa della movimentazione dello stesso, dell'attivazione e disattivazione dei LED, e della gestione dello stato di errore.

5. MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

5.1.1 Variabili

Le variabili presenti all'interno del macrostato sono le seguenti:

Tipo	Nome	Valore	Descrizione
local_data	Var_Inattivo	ON/OFF	Variabile utilizzata per segnalare che lo stato Inattivo è attualmente attivo
local_data	Var_Chiuso	ON/OFF	Variabile utilizzata per segnalare che lo stato Chiuso è attualmente attivo
local_data	Var_Aperto	ON/OFF	Variabile utilizzata per segnalare che lo stato Aperto è attualmente attivo
local_data	Var_In_Chiusura	ON/OFF	Variabile utilizzata per segnalare che lo stato In Chiusura è attualmente attivo
output_data	LedG	ON/OFF	Variabile utilizzata per l'attivazione del LED verde in output
output_data	LedR	ON/OFF	Variabile utilizzata per l'attivazione del LED Rosso in output
output_data	LedY	ON/OFF	Variabile utilizzata per l'attivazione del LED Giallo in output
input_data	P1	ON/OFF	Variabile utilizzata per la segnalazione di un ostacolo
input_data	P2	ON/OFF	Variabile utilizzata per la segnalazione della chiusura del cancello
constant_data	Blink_Period	2	Costante utilizzata per la durata del blink del LED Giallo
local_data	T_L	[10, 120]	Contatore del tempo di lavoro
local_data	T_C	[10, 120]	Contatore del tempo di chiusura
constant_data	ErrDur	10	Costante utilizzata per la durata del lampeggio del LED rosso
local_event	buttonpressed1	TRUE/FALSE	Evento per la pressione del pulsante B1

Table 5.1: Tabella variabili Cancello

5.1.2 Stati e Funzionamento

Il macrostato **Cancello** presenta i seguenti stati e macrostati:

- **Inattivo**
- **Chiusura**

5. MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

- In Chiusura
 - * Blink_COff
 - * Blink_COn
- Chiuso
- Errore
 - * In errore
 - Blink_EOff
 - Blink_EOn
 - * Led_Rosso_Errorre
- Apertura
 - In Apertura
 - * Blink_AOff
 - * Blink_AOn
 - Aperto
 - Aperto_Con_Ostacolo

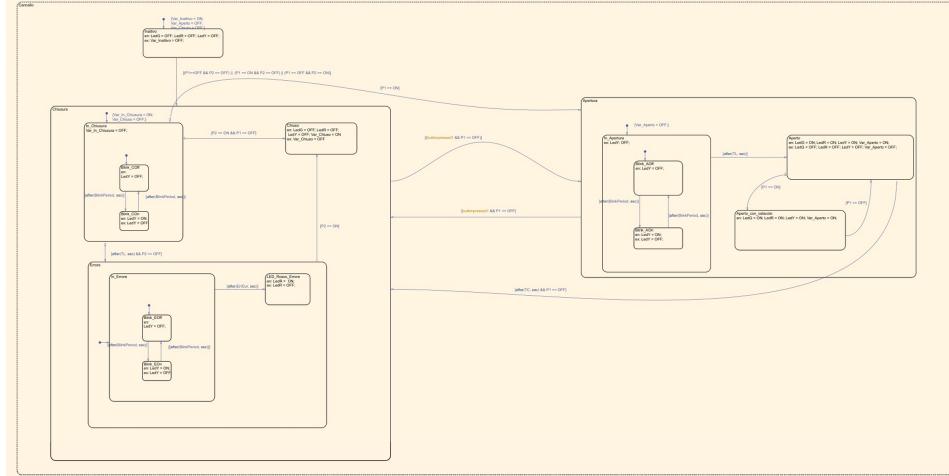


Figure 5.1: Stato Cancello

Proseguiamo ora descrivendo il funzionamento dell'intero macrostato **Cancello**

1. All'avvio del sistema lo stato è impostato come **Inattivo**, inizializza le tre variabili *Var_Inattivo*, *Var_Aperto* e *Var_Chiuso* e si occupa di spegnere tutti i LED. Quando si esce da codesto stato, si pone *Var_Inattivo* ad OFF. La condizione di uscita prevede che si vada nello stato **Chiusura** se e solo se i due sensori di presenza non sono contemporaneamente e inizialmente attivi.

5. MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

2. All'interno del macrostato **Chiusura**, lo stato iniziale è **In_Chiusura**, in cui viene inizializzata la variabile *Var_In_Chiusura* ad ON, mentre la variabile *Var_Chiuso* viene posta ad OFF. Inoltre, nel macrostato **In_Chiusura**, avviene il blink del LED giallo grazie all'alternarsi degli stati sottostanti con periodo di 2 secondi.

Le transizioni in uscita da questo stato sono:

- Se si attiva il sensore di presenza P2 di chiusura del cancello, ed il sensore P1 risulta spento (non rilevando dunque ostacoli), si entra nello stato **Chiuso**, nel quale vengono spenti tutti i LED e viene attivata la variabile *Var_Chiuso*
 - Se scade il timer riguardante il tempo di lavoro T_L senza conseguente attivazione del sensore di presenza P2, si entra nello stato **Errore**, nel quale si accende il LED rosso se e solo se vi si permane per ulteriori 10 secondi. Si esce da questo stato solo se ritorna attivo il sensore di presenza P2
 - Se invece, si attiva il sensore di presenza P1 che rileva un ostacolo durante la chiusura, si entra nel macrostato **Apertura**
3. All'interno del macrostato **Apertura**, lo stato iniziale è quello **In_Apertura**, nel quale viene effettuato il blink del LED giallo grazie all'alternarsi degli stati sottostanti con periodo di 2 secondi. Da questo stato si esce solo quando scade il tempo di lavoro T_L per poi passare allo stato **Aperto**.
 4. In questo stato vengono accesi tutti i LED per segnalare la corretta apertura del cancello e viene posta ad ON la variabile *Var_Aperto*. In uscita da questo stato invece, i LED verranno spenti ponendo ad OFF la variabile *Var_Aperto*.

Le possibili transizioni per uscire da questo stato sono:

- Se viene rilevato un ostacolo si entra nello stato **Aperto_Con_Ostacolo** nel quale vengono riaccesi tutti i LED e l'unica transizione possibile per uscire da questo stato è quando non viene più rilevato. Questa funzionalità è stata implementata per azzerare il timer di chiusura ogni volta che un ostacolo viene rilevato
 - Se scade il tempo di chiusura e non viene rilevato alcun ostacolo, allora si entra nel macrostato **Chiusura**, il quale avvia il processo di chiusura del cancello
5. Sia nel macrostato **Chiusura** che **Apertura**, la transizione di uscita è identica, nella quale viene premuto il pulsante B1 e non vi sono ostacoli rilevati dal sensore di presenza P1.

5.2 Tuning Charts

In questa sezione analizzeremo l'implementazione dei due moduli utilizzati per le regolazioni, **Regolazione_Tempo_Chiusura** e **Regolazione_Tempo_Lavoro**. Essi permettono di regolare le due variabili T_C e T_L attraverso la pressione dei pulsanti **B2** e **B3** rispettivamente.

5.2.1 Variabili

Le variabili presenti all'interno dei due macrostati sono le seguenti:

Tipo	Nome	Valore	Descrizione
input_data	T_L	[10, 120]	Contatore del tempo di lavoro
input_data	Var_Chiuso	ON/OFF	Variabile utilizzata per segnalare che lo stato Chiuso è attualmente attivo
local_event	buttonpressed3	TRUE/FALSE	Evento per la pressione del pulsante B3
input_data	T_C	[10, 120]	Contatore del tempo di chiusura
local_event	buttonpressed2	TRUE/FALSE	Evento per la pressione del pulsante B2

Table 5.2: Tabella variabili Tuning

5.2.2 Stati e Funzionamento

I due macrostati presentano i seguenti stati al loro interno:

- **Regolazione_Tempo_Lavoro**
 - *Tempo_Lavoro*
 - *Aggiunta_Tempo_Lavoro*

- **Regolazione_Tempo_Chiusura**
 - *Tempo_Chiusura*
 - *Aggiunta_Tempo_Chiusura*

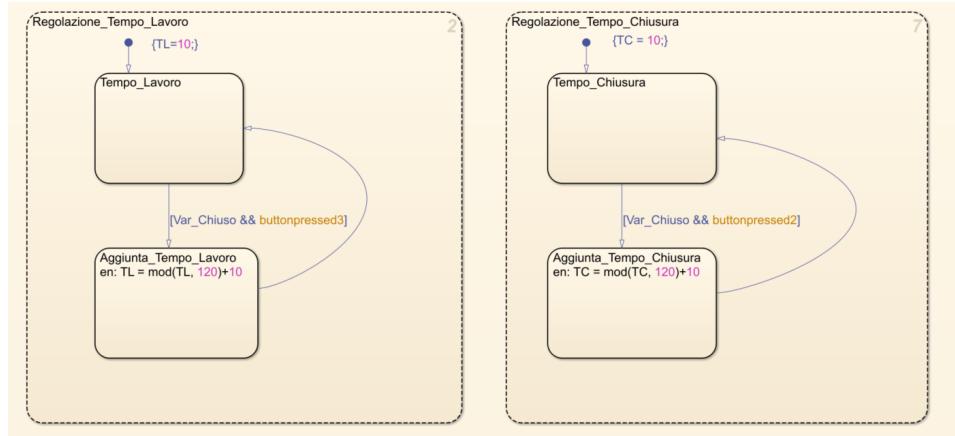


Figure 5.2: Stati Regolazioni

Proseguiamo ora analizzando il funzionamento di un singolo macrostato (**Regolazione_Tempo_Chiusura**), essendo l'altro identico per costruzione:

1. Si parte dallo stato **Tempo_Chiusura** nel quale la variabile T_C viene inizializzata al valore 10 (secondi).
2. In seguito, se ci troviamo nello stato **Chiuso** e parallelamente viene rilevata la pressione del pulsante **B2** allora, ci spostiamo nello stato **Aggiunta_Tempo_Chiusura** nel quale la variabile T_C viene incrementata di 10 (secondi).
3. Una volta avvenuto l'incremento, si ritorna nello stato iniziale **Tempo_Chiusura** nel quale si resta in attesa delle condizioni tali per cui si possono effettuare gli ulteriori incrementi.

5.3 Obstacle Chart

In questa sezione analizzeremo l'implementazione del macrostato **Ostacolo**, il quale consente la gestione del sistema in caso di rilevazione dello stesso. Esso permette il lampeggio del LED verde in output, nel caso il sensore di presenza **P1** si attivi.

5.3.1 Variabili

Le variabili all'interno del macrostato sono le seguenti:

5. MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

Tipo	Nome	Valore	Descrizione
constant_data	GreenDur	30	Costante utilizzata per la durata del lampeggio del LED verde
input_data	Var_Chiuso	ON/OFF	Variabile utilizzata per segnalare che lo stato Chiuso è attualmente attivo
input_data	Var_Inattivo	ON/OFF	Variabile utilizzata per segnalare che lo stato Inattivo è attualmente attivo
input_data	Var_Aperto	ON/OFF	Variabile utilizzata per segnalare che lo stato Aperto è attualmente attivo
local_event	buttonpressed1	TRUE/FALSE	Evento per la pressione del pulsante B1
constant_data	BlinkPeriod2	1	Costante utilizzata per la durata del blink del LED verde
input_data	P1	ON/OFF	Variabile utilizzata per la segnalazione di un ostacolo
output_data	LedG	ON/OFF	Variabile utilizzata per l'attivazione del LED verde in output

Table 5.3: Tabella variabili Ostacolo

5.3.2 Stati e Funzionamento

Il macrostato **Ostacolo** presenta i seguenti stati al suo interno:

- **Fermo**
- **Ostacolo_Presente**
- **Blink_OOff**
- **Blink_OOn**
- **Aperto_Con_Ostacolo_Led**

5. MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

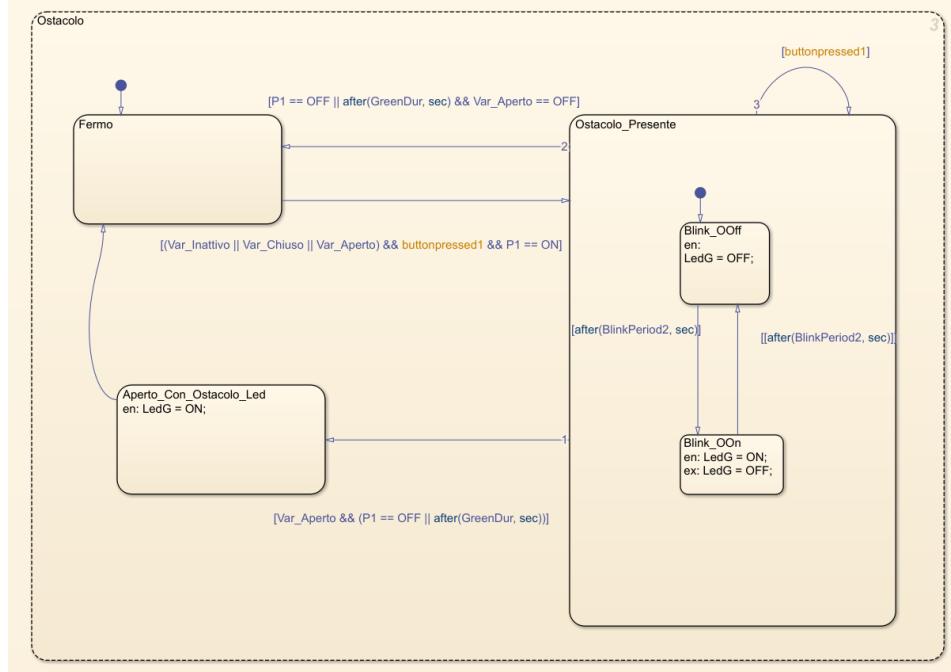


Figure 5.3: Stato Ostacolo

Proseguiamo ora descrivendo il funzionamento dell'intero macrostato **Ostacolo**:

1. All'attivazione del sistema, si entra nello stato **Fermo** (che rappresenta uno stato di *Idle*) dal quale si esce solo quando ci si trova o nello stato **Inattivo** o nello stato **Chiuso** o nello stato **Aperto** e al contempo il sensore di presenza **P1** è attivo con annessa rilevazione di pressione del pulsante **B1**.
2. La transizione sopra citata, ci porta nello stato **Ostacolo_Presente** nel quale viene effettuato il *blink* del LED verde. Nel dettaglio, lo stato iniziale del blink è **Blink_OOff** il quale si occupa di spegnere il LED verde, mentre lo stato **Blink_OOn** si occupa di accenderlo. La transizione tra i due stati avviene in accordo alla scadenza di un timer, di durata pari a **Blink_Period2**, ovvero 1 secondo.
3. Dall'attuale stato, si esce quando si verifica una delle seguenti condizioni:
 - Non viene più rilevato l'ostacolo poiché il sensore di presenza P1 si disattiva, oppure poiché scade il timer associato al lampeggio del LED. In aggiunta a questi requisiti **Var_Aperto** deve essere uguale ad OFF. In questo caso, ritorniamo nello stato **Fermo**.
 - Non viene più rilevato l'ostacolo poiché il sensore di presenza P1 si disattiva, oppure poiché scade il timer associato al lampeggio del LED, ma ci troviamo parallelamente nello stato **Aperto**, dunque, il nuovo stato diventa

Aperto_Con_Ostacolo_Led nel quale viene riattivato il LED verde dopo il lampeggio. Infine si ritorna nello stato **Fermo**.

- Si preme il pulsante B1 e si ritorna nello stato **Ostacolo_Presente** in modo da resettare il timer di 30 secondi relativo al lampeggio del LED verde.

5.4 Buttons Charts

In questa sezione analizzeremo l'implementazione dei macrostati **Button**. Essi permettono di rilevare quando il pulsante è stato premuto consentendo l'attivazione dell'evento. Vedremo prima l'implementazione del macrostato **Button1** e successivamente di **Button2** e **Button3** che sono identici per costruzione.

5.4.1 Variabili

Le variabili all'interno del macrostato **Button1** sono le seguenti:

Tipo	Nome	Valore	Descrizione
input_data	B1	ON/OFF	Variabile utilizzata per segnalare la pressione del pulsante
local_event	buttonpressed1	TRUE/FALSE	Evento per la pressione del pulsante B1

Table 5.4: Tabella variabili Button1

Le variabili all'interno dei macrostati **Button2** e **Button3** sono le seguenti:

Tipo	Nome	Valore	Descrizione
input_data	B2	ON/OFF	Variabile utilizzata per segnalare la pressione del pulsante
local_event	buttonpressed2	TRUE/FALSE	Evento per la pressione del pulsante B2
input_data	B3	ON/OFF	Variabile utilizzata per segnalare la pressione del pulsante
local_event	buttonpressed3	TRUE/FALSE	Evento per la pressione del pulsante B3

Table 5.5: Tabella variabili Button2 e Button3

5.4.2 Stati e Funzionamento

Proseguiamo ora descrivendo inizialmente il funzionamento del macrostato **Button1** e successivamente dei restanti. Tutti i pulsanti presentano un'implementazione su fronte di discesa.

5. MODELLAZIONE DEL SISTEMA NELL'AMBIENTE SIMULINK

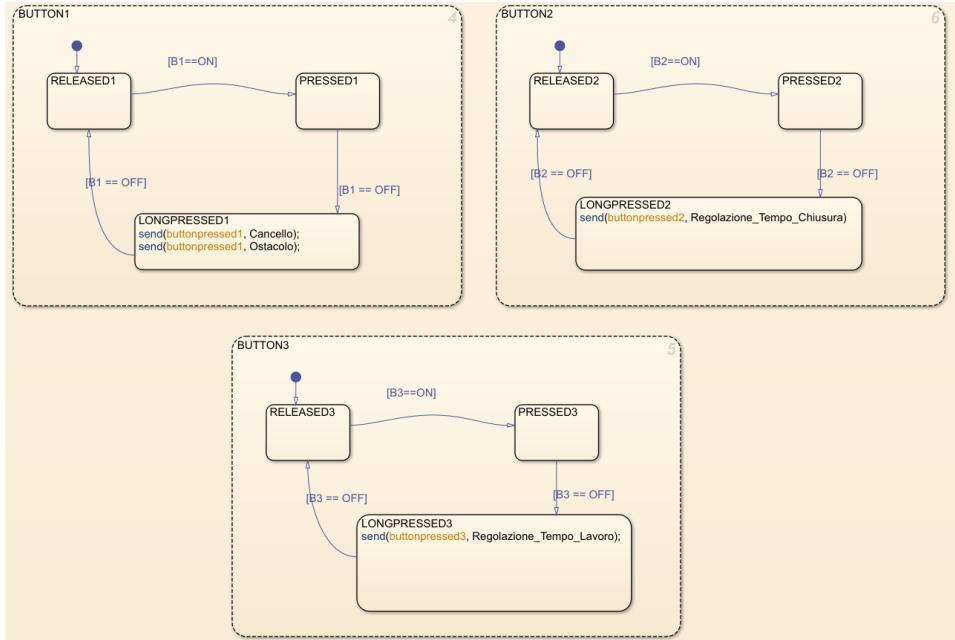


Figure 5.4: Stati Buttons

Button1:

1. Si entra nello stato **Released1**, il quale sottolinea il fatto che all'accensione del sistema il pulsante è rilasciato.
2. Alla pressione del pulsante B1 si entra nello stato **Pressed1**, per segnalare che il pulsante è stato premuto. Questi due stati non presentano azioni in entrata o in uscita, ma sono utilizzati solo per rappresentare l'interazione con l'utente.
3. Quando verrà rilasciato il pulsante, ovvero quando $B1 == OFF$, si entra nello stato **Longpressed1**, il quale si occupa di inviare l'evento *buttonpressed1* ai due macrostati **Cancello** e **Ostacolo** per segnalare la corretta pressione del pulsante.

Button2 e Button3:

1. Si entra nello stato **Released2/Released3**, il quale sottolinea che all'accensione del sistema il pulsante viene rilasciato.
2. Alla pressione del pulsante B2/B3 si entra nello stato **Pressed2/Pressed3**, per indicare che il pulsante è stato premuto. Questi due stati non posseggono azioni in entrata o in uscita, ma segnalano solo l'interazione con l'utente.
3. Quando verrà rilasciato il pulsante, ovvero quando $B2/B3 == OFF$, si entra nello stato **Longpressed2/Longpressed3**, il quale si occupa di inviare l'evento *buttonpressed2/buttonpressed3* ai due macrostati **Regolazione_Tempo_Chiusura** o **Regolazione_Tempo_Lavoro** e segnalarne la corretta pressione.

5.5 Final Chart

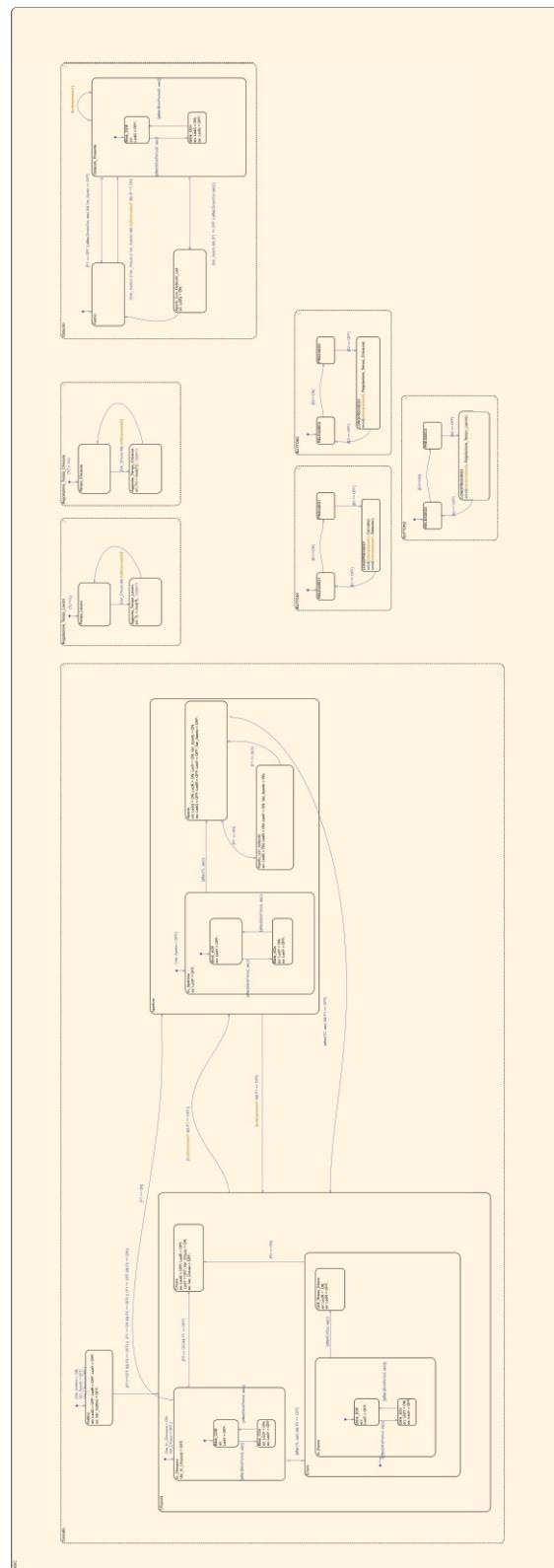


Figure 5.5: Final Chart

Il grafico finale in figura 5.5 rappresenta il diagramma a stati nella sua interezza. Vi sono 6 macrostati che vengono eseguiti in parallelo grazie alla presenza del macrostato **LOGIC** esterno avente decomposizione **AND**.

5.6 Inputs & Outputs

Il grafico in figura 5.6 mostra il diagramma precedente con i relativi input e output, ovvero 5 pulsanti (rispettivamente B1, B2, B3, P1, P2) e 3 LED (LedG, LedY, LedR).

I pulsanti relativi ai sensori di presenza P1 e P2 sono degli switch collegati a delle costanti che vengono poste ad 1 quando il sensore si attiva e a 0 quando il sensore di disattiva. Anche i bottoni sono stati collegati a delle costanti ed essi sono configurati in modalità *Momentary*, dunque il loro valore è normalmente 0, ma diventa 1 solo quando viene premuto.

Per i LED sono stati utilizzati dei blocchi *Out* collegati a blocchi *Lamp* di colore verde, rosso e giallo per segnalarne la corretta accensione.



Figure 5.6: Input e Output

CAPITOLO 6

TEST SIMULINK

In questo capitolo ci concentreremo su verifica e validazione del diagramma a stati attraverso l'utilizzo di Simulink Test in Simulink. Essi permettono di isolare e testare singoli componenti del modello, garantendo che ogni parte del sistema funzioni correttamente. Esploreremo le tecniche per creare, configurare ed eseguire test dettagliati e analizzare i risultati per identificare e correggere eventuali errori. Questo processo è fondamentale per assicurare l'affidabilità e l'accuratezza del modello complessivo, consentendo di simulare scenari diversi e verificare il comportamento del sistema in condizioni variabili.

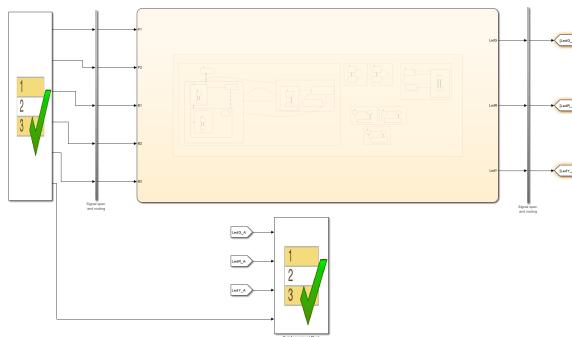


Figure 6.1: Esempio Test Harness

6.1 Test Apertura e Chiusura Cancello

Di seguito è riportato il test relativo alla verifica della corretta apertura e chiusura del cancello quando l'utente preme il pulsante B1. In particolare, si effettua la verifica del corretto lampeggio del LED giallo in fase di verifica, della totale accensione di tutti i LED

6. TEST SIMULINK

in fase di apertura e del totale spegnimento di questi ultimi in fase di chiusura.

Step	Transition	Next Step	Description
Int %> initialize data outputs. P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	: true	GateClose	▼ Situazione iniziale in cui il cancello non è chiuso e non c'è presente nessun ostacolo
GateClose P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 0;	: true	ButtonPressed	▼ Iostizziamo che il cancello si chiude poiché il sensore di presenza P2 si è attivato
ButtonPressed P1 = 0; P2 = 1; B1 = 1; B2 = 0; B3 = 0;	: true	ButtonReleased	▼ L'utente preme il bottone B1 per aprire il cancello
ButtonReleased P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	: after(T_L , sec)	GateOpen	▼ L'utente rilascia il pulsante per mandare il cancello in fase di apertura
GateOpen P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	: true	ButtonPressed1	▼ Il cancello si considera Aperto dopo T_L secondi.
ButtonPressed1 P1 = 0; P2 = 0; B1 = 1; B2 = 0; B3 = 0;	: true	ButtonReleased1	▼ L'utente preme il bottone B1 per aprire il cancello.
ButtonReleased1 P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	: after(T_L , sec)	GateClosed	▼ L'utente rilascia il pulsante per mandare il cancello in fase di chiusura
GateClosed P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 0;			Il cancello è chiuso se P2 è attivo.

Figure 6.2: Test Sequence

Step	Transition	Next Step	Description
B1 Run			
— VerifyInit when TSActiveStepIn == TSActiveStepEnum.Int verify(~Led0 & ~LedR & ~LedY);			
; VerifyGateClose when TSActiveStepIn == TSActiveStepEnum.GateClose verify(~Led0 & ~LedR & ~LedY);	Add step after - Add sub-step - □		
; VerifyButtonPressed when TSActiveStepIn == TSActiveStepEnum.ButtonPressed verify(~Led0 & ~LedR & ~LedY);			
B1 VerifyButtonReleased when TSActiveStepIn == TSActiveStepEnum.ButtonReleased			
— VerifyButtonReleased_1 verify(~Led0 & ~LedR & ~LedY);	: after(2, sec)	VerifyB... ▼	
VerifyButtonReleased_2 verify(~Led0 & ~LedR & ~LedY);	: after(2, sec)	VerifyB... ▼	
VerifyGateOpen when TSActiveStepIn == TSActiveStepEnum.GateOpen verify(Led0 & LedR & LedY);			
VerifyButtonPressed1 when TSActiveStepIn == TSActiveStepEnum.ButtonPressed1 verify(Led0 & LedR & LedY);			
B1 VerifyButtonReleased1 when TSActiveStepIn == TSActiveStepEnum.ButtonReleased1			
— VerifyButtonReleased_1 verify(~Led0 & ~LedR & ~LedY);	: after(2, sec)	VerifyB... ▼	
VerifyButtonReleased_2 verify(~Led0 & ~LedR & ~LedY);	: after(2, sec)	VerifyB... ▼	
VerifyGateClosed when TSActiveStepIn == TSActiveStepEnum.GateClosed verify(~Led0 & ~LedR & ~LedY);			
step_1_2			

Figure 6.3: Test Assessment

6.2 Test Errore Cancello

Si riporta il test relativo alla verifica dello stato di errore del cancello qualora il sensore di presenza P2 non si attivi dopo T_L secondi durante la fase di chiusura. Si ricorda che, dopo aver atteso tale tempo, il dispositivo attende ulteriori 10 secondi prima di accendere il LED rosso e segnalare la definitiva permanenza nello stato di errore.

6. TEST SIMULINK

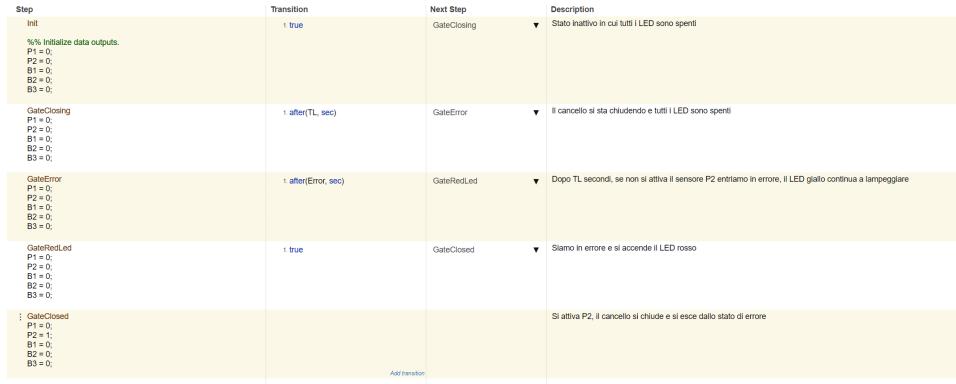


Figure 6.4: Test Sequence



Figure 6.5: Test Assessment

6.3 Test Rilevazione Ostacolo se Aperto

Si riporta il test relativo alla rilevazione di un ostacolo quando il cancello si trova nello stato **Aperto**. Si ricorda che in caso di rilevazione positiva, mentre il cancello si sposta nello stato **Aperto con Ostacolo**, il macrostato **Ostacolo** permette il lampeggio del LED verde per un periodo di 30 secondi se l'utente tenta di chiudere il cancello con il sensore P1 attivo.

6. TEST SIMULINK

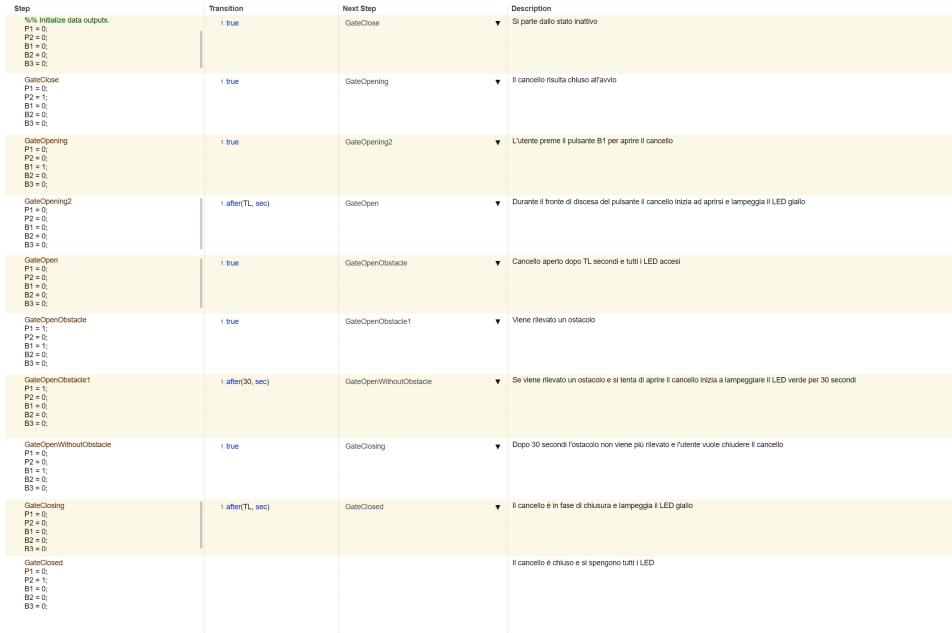


Figure 6.6: Test Sequence

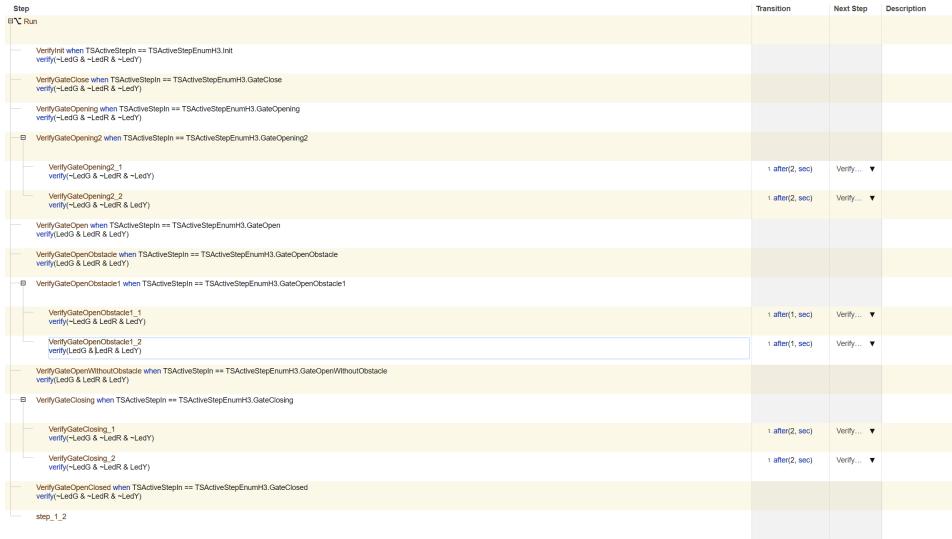


Figure 6.7: Test Assessment

6.4 Test Rilevazione Ostacolo se In Chiusura

Si riporta il test relativo alla rilevazione di un ostacolo quando il cancello è in fase di chiusura. In particolare, il cancello si sposta in fase di apertura continuando il lampeggio del LED giallo per poi aprirsi completamente. In seguito, il test prevede l'attesa del tempo di chiusura automatica T_C per poi chiudersi definitivamente.

6. TEST SIMULINK

Step	Transition	Next Step	Description
Init %% Initialize data outputs. P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	: true	GateClosing	Stato iniziale in cui tutti i LED sono spenti
GateClosing P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	: true	GateObstacle	Cancello in chiusura in cui il LED giallo lampeggia ogni 2 secondi
GateObstacle P1 = 1; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	: after(TL, sec)	GateOpened	Viene rilevato un ostacolo in fase di chiusura e si entra in fase di apertura in cui lampeggi il LED giallo
GateOpened P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	: after(TC, sec)	GateAutomaticClosing	Cancello aperto con tutti i LED accesi
GateAutomaticClosing P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	: after(TL, sec)	GateClosed	Cancello in chiusura con il LED giallo che lampeggia
GateClosed P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 0;			Cancello chiuso con tutti i LED spenti

Figure 6.8: Test Sequence

Step	Transition	Next Step	Description
B[C Run			
VerifyInit when TSActiveStepIn == TSActiveStepEnumH4.Int verify(~LedG & ~LedR & ~LedY)			
B VerifyGateClosing when TSActiveStepIn == TSActiveStepEnumH4.GateClosing			
VerifyGateClosing_1 verify(~LedG & ~LedR & ~LedY)	: after(2, sec)	VerifyGateClosing_2	
VerifyGateClosing_2 verify(~LedG & ~LedR & LedY)	: after(2, sec)	VerifyGateClosing_1	
B VerifyGateObstacle when TSActiveStepIn == TSActiveStepEnumH4.GateClosing			
VerifyGateObstacle_1 verify(~LedG & ~LedR & ~LedY)	: after(2, sec)	VerifyGateObstacle_2	
VerifyGateObstacle_2 verify(~LedG & ~LedR & LedY)	: after(2, sec)	VerifyGateObstacle_1	
B VerifyGateOpened when TSActiveStepIn == TSActiveStepEnumH4.GateOpened verify(LedG & LedR & LedY)			
B VerifyGateAutomaticClosing when TSActiveStepIn == TSActiveStepEnumH4.GateAutomaticClosing			
VerifyGateAutomaticClosing_1 verify(~LedG & ~LedR & ~LedY)	: after(2, sec)	VerifyGateAutomaticClosing_2	
VerifyGateAutomaticClosing_2 verify(~LedG & ~LedR & LedY)	: after(2, sec)	VerifyGateAutomaticClosing_1	
VerifyGateClosed when TSActiveStepIn == TSActiveStepEnumH4.GateClosed verify(~LedG & ~LedR & ~LedY)			
step_1_2			

Figure 6.9: Test Assessment

6.5 Test Rilevazione Ostacolo se Chiuso

Si riporta il test relativo alla rilevazione di un ostacolo quando il cancello è chiuso. In particolare, il cancello non si apre se l'utente preme il pulsante B1, ma inizia il lampeggio del LED verde per 30 secondi, per poi spegnersi insieme agli altri due LED.

6. TEST SIMULINK

Step	Transition	Next Step	Description
Init %% Initialize data outputs. P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i: true	GateClosing	Stato iniziale in cui i LED sono tutti spenti
GateClosing P1 = 0; P2 = 0; B1 = 1; B2 = 0; B3 = 0;	i: after(tL, sec)	GateClosed	Cancello in chiusura con lampeggio del LED giallo
GateClosed P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 0;	i: true	GateClosedWithObstacle	Cancello chiuso con tutti i LED spenti
GateClosedWithObstacle P1 = 1; P2 = 1; B1 = 1; B2 = 0; B3 = 0;	i: true	GateClosedWithObstacle1	L'utente preme il pulsante B1 per aprire il cancello ma non può poiché vi è un ostacolo
GateClosedWithObstacle1 P1 = 1; P2 = 1; B1 = 0; B2 = 0; B3 = 0;	i: after(30, sec)	GateClosedWithoutObstacle	Il cancello non si apre poiché è presente l'ostacolo, dunque inizia il lampeggio del LED verde per 30 secondi
GateClosedWithoutObstacle P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 0;			Il cancello è chiuso senza ostacolo e i LED sono spenti

Figure 6.10: Test Sequence

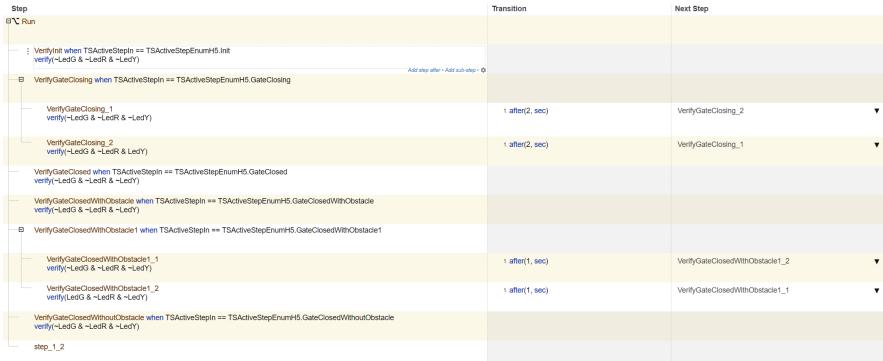


Figure 6.11: Test Assessment

6.6 Regolazione T_C e T_L

Si riporta il test relativo alla regolazione del tempo di lavoro e del tempo di chiusura. In particolare, il cancello si chiude inizialmente con T_L pari a 10 secondi. In seguito, l'utente preme il pulsante B2 e B3 per aumentare sia T_L che T_C di 10 secondi. Dopodiché, il cancello viene portato in fase di apertura (e successivamente chiusura) con il nuovo tempo di lavoro.

6. TEST SIMULINK

Step	Transition	Next Step	Description
Init % Initialize data outputs. P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i true	GateClosing	Stato iniziale in cui il cancello è nello stato inattivo e tutti i LED sono spenti
GateClosing P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i after(TL_10, sec)	GateClosed	Il cancello è nello stato in chiusura, dunque lampeggia il LED giallo
GateClosed P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 0;	i true	Tuning10to20_Closing	Il cancello è chiuso dopo TL secondi e tutti i LED sono spenti
Tuning10to20_Closing P1 = 0; P2 = 1; B1 = 0; B2 = 1; B3 = 0;	i true	Tuning10to20_Closing_1	
Tuning10to20_Closing_1 P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 0;	i true	Tuning10to20_Working	L'utente preme il pulsante B2 per aumentare il tempo di chiusura di 10 secondi
Tuning10to20_Working P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 1;	i true	Tuning10to20_Working_1	
Tuning10to20_Working_1 P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 0;	i true	GateOpening	L'utente preme il pulsante B3 per aumentare il tempo di lavoro di 10 secondi
GateOpening P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i true	GateOpening1	
GateOpening1 P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i after(TL_20, sec)	GateOpened	L'utente apre il cancello premendo il pulsante B1
GateOpened P1 = 0; P2 = 0; B1 = 1; B2 = 0; B3 = 0;	i after(TC_20, sec)	GateClosing1	Il cancello è aperto e tutti i LED sono accesi
GateClosing1 P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i after(TL_20, sec)	GateClosed1	
GateClosed1 P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;			

Figure 6.12: Test Sequence

Step	Transition	Next Step	Description
E ^x Run			
VerifyInit when TSActiveStepIn == TSActiveStepEnumH7.Init verify!~LedG & ~LedR & ~LedY			
VerifyGateClosing when TSActiveStepIn == TSActiveStepEnumH7.GateClosing VerifyGateClosing_1 verify!~LedG & ~LedR & ~LedY	i after(2, sec)	Verify...	
VerifyGateClosing_2 verify!~LedG & ~LedR & LedY	i after(2, sec)	Verify...	
VerifyGateClosed when TSActiveStepIn == TSActiveStepEnumH7.GateClosed verify!~LedG & ~LedR & ~LedY			
VerifyTun when TSActiveStepIn == TSActiveStepEnumH7.Tuning10to20_Closing verify!~LedG & ~LedR & ~LedY			
VerifyTun when TSActiveStepIn == TSActiveStepEnumH7.Tuning10to20_Closing_1 verify!~LedG & ~LedR & ~LedY			
VerifyTun when TSActiveStepIn == TSActiveStepEnumH7.Tuning10to20_Working verify!~LedG & ~LedR & ~LedY			
VerifyTun when TSActiveStepIn == TSActiveStepEnumH7.Tuning10to20_Working_1 verify!~LedG & ~LedR & ~LedY			
VerifyGateOpening when TSActiveStepIn == TSActiveStepEnumH7.GateOpening verify!~LedG & ~LedR & ~LedY			
VerifyGateOpening1 when TSActiveStepIn == TSActiveStepEnumH7.GateOpening1 VerifyGateOpening_1 verify!~LedG & ~LedR & ~LedY	i after(2, sec)	Verify...	
VerifyGateOpening_2 verify!~LedG & ~LedR & ~LedY	i after(2, sec)	Verify...	
VerifyGateOpened when TSActiveStepIn == TSActiveStepEnumH7.GateOpened verify!~LedG & ~LedR & ~LedY			
VerifyGateClosing1 when TSActiveStepIn == TSActiveStepEnumH7.GateClosing1 VerifyGateClosing_1 verify!~LedG & ~LedR & ~LedY	i after(2, sec)	Verify...	
VerifyGateClosing_2 verify!~LedG & ~LedR & ~LedY	i after(2, sec)	Verify...	
VerifyGateClosed when TSActiveStepIn == TSActiveStepEnumH7.GateClosed1 VerifyGateClosed_1			

Figure 6.13: Test Assessment

6. TEST SIMULINK

6.7 Regolazione T_C e T_L non nello stato Chiuso

Si riporta il test relativo al fallimento della regolazione del tempo di lavoro e del tempo di chiusura mentre il cancello non si trova nello stato **Chiuso**. In particolare, l'utente prova a regolare T_C e T_L premendo i pulsanti **B2** e **B3** mentre il cancello si trova nello stato **Aperto_Con_Ostacolo**, dunque la regolazione non avviene con successo e le due variabili precedentemente citate non cambiano il loro valore.

Step	Transition	Next Step	Description
Int %% initialize data outputs. P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i true	GateClosed	Il cancello è nello stato iniziale con tutti i LED spenti
GateClosed P1 = 0; P2 = 0; B1 = 1; B2 = 0; B3 = 0;	i true	GateOpening	Il cancello è chiuso con tutti i LED spenti.
GateOpening P1 = 0; P2 = 0; B1 = 1; B2 = 0; B3 = 0;	i true	GateOpening1	L'utente preme il pulsante B1 per aprire il cancello
GateOpening1 P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i after(TL, sec)	GateOpenedObstacle	Il cancello è in fase di apertura e lampeggia il LED giallo.
GateOpenedObstacle P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i true	GateOpenedObstacleTuning	Viene rilevato un ostacolo e si entra nello stato Aperto con Ostacolo
GateOpenedObstacleTuning P1 = 1; P2 = 0; B1 = 0; B2 = 1; B3 = 1;	i true	GateOpenedObstacleTuning1	L'utente prova a regolare il tempo di chiusura e tempo di lavoro senza successo poiché non ci troviamo nello stato Chiuso
GateOpenedObstacleTuning1 P1 = 1; P2 = 0; B1 = 0; B2 = 1; B3 = 0;	i after(TC, sec)	GateClosing	Si attende il tempo di chiusura non modificato prima di chiudere
GateClosing P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i after(TL, sec)	GateClosed1	Inizia la fase di chiusura e lampeggia il LED giallo
GateClosed1 P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;			Il cancello è chiuso e tutti i LED sono spenti

Figure 6.14: Test Sequence

Step	Transition	Next Step	Description
B1 Run			
VerifyInit when TSActiveStepIn == TSActiveStepEnumH.B_Initial			
VerifyGateClosed when TSActiveStepIn == TSActiveStepEnumH.B_GateClosed			
VerifyOpening when TSActiveStepIn == TSActiveStepEnumH.B_GateOpening			
VerifyOpening1 when TSActiveStepIn == TSActiveStepEnumH.B_GateOpening1	i after(2, sec)	Verify...	
VerifyOpening1_1 when TSActiveStepIn == TSActiveStepEnumH.B_GateOpening1	i after(2, sec)	Verify...	
VerifyOpening1_2 when TSActiveStepIn == TSActiveStepEnumH.B_GateOpening1	i after(2, sec)	Verify...	
VerifyOpenedOb when TSActiveStepIn == TSActiveStepEnumH.B_GateOpenedObstacle			
VerifyOpenedOB when TSActiveStepIn == TSActiveStepEnumH.B_GateOpenedObstacleTuning			
VerifyOpenedOB1 when TSActiveStepIn == TSActiveStepEnumH.B_GateOpenedObstacleTuning1			
VerifyClosing when TSActiveStepIn == TSActiveStepEnumH.B_GateClosing			
VerifyClosingT1 when TSActiveStepIn == TSActiveStepEnumH.B_GateClosing	i after(2, sec)	Verify...	
VerifyClosingT2 when TSActiveStepIn == TSActiveStepEnumH.B_GateClosing	i after(2, sec)	Verify...	
VerifyGateClosed1 when TSActiveStepIn == TSActiveStepEnumH.B_GateClosed1			
step_1_2			

Figure 6.15: Test Assessment

6.8 Reset del timer durante rilevazione Ostacolo

Si riporta il test relativo al reset del timer relativo al lampeggio del LED verde se l'utente preme ulteriormente il pulsante B1. In particolare, se ci si trova nello stato **Aperto_Con_Ostacolo**, se l'utente interagisce col pulsante B1 per chiudere il cancello, il LED verde inizia a lampeggiare per 30 secondi senza alcuna movimentazione del cancello. Il test dimostra che, qualora l'utente dovesse premere di nuovo il pulsante di chiusura (ad esempio dopo soli 5 secondi), il timer si resetta ed il conteggio riparte da 30 secondi.

Step	Transition	Next Step	Description
Init %% Initialize data outputs. P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i true	GateClose	▼ Cancello nello stato inattivo, tutti i LED sono spenti
GateClose P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 0;	i true	GateOpening	▼ Cancello nello stato Chiuso, tutti i LED sono nuovamente spenti
GateOpening P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i true	GateOpening1	▼ L'utente preme il pulsante B1 per aprire il cancello
GateOpening1 P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i after(TL_sec)	GateOpened	▼ Il cancello è in fase di Apertura, lampeggia il LED giallo ogni 2 secondi
GateOpened P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i true	GateOpenedWithObstacle	▼ Il cancello è aperto e tutti i LED sono accesi
GateOpenedWithObstacle P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i true	GateObstacleOpening	▼ Il cancello è nello stato Aperto con Ostacolo in cui è stato rilevato un ostacolo tramite il sensore di presenza P1
GateObstacleOpening P1 = 1; P2 = 0; B1 = 1; B2 = 0; B3 = 0;	i true	GateObstacleOpening1	▼ L'utente prova a chiudere il cancello premendo il pulsante B1
GateObstacleOpening1 P1 = 1; P2 = 0; B1 = 1; B2 = 0; B3 = 0;	i after(5_sec)	GateObstacleOpening2	▼ Il cancello non si chiude poiché è presente un ostacolo, dunque lampeggia il LED verde per 30 secondi
i GateObstacleOpening2 P1 = 1; P2 = 0; B1 = 1; B2 = 0; B3 = 0;	i true	GateObstacleOpening3	▼ Prima della scadenza del timer, dopo appena 5 secondi, l'utente ripreme il pulsante B1 per chiudere il cancello
i GateObstacleOpening3 P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i after(30_sec)	GateOpened1	▼ Il cancello non si chiude poiché è ancora presente l'ostacolo, dunque riparte il timer di 30 secondi relativo al lampeggio del LED verde
i GateOpened1 P1 = 0; P2 = 0; B1 = 0; B2 = 0; B3 = 0;	i after(TC_sec)	GateClosing	▼ Dopo 30 secondi, il cancello ritorna nello stato Aperto
i GateClosing P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 0;	i after(TL_sec)	GateClosed	▼ Dopo TC secondi, il cancello è in fase di chiusura e lampeggia il LED giallo
GateClosed P1 = 0; P2 = 1; B1 = 0; B2 = 0; B3 = 0;			Dopo TL secondi il cancello è chiuso e tutti i LED sono spenti

Figure 6.16: Test Sequence

6. TEST SIMULINK

Step	Transition	Next Step	Description
Run			
— VerifyInit when TSActiveStepIn == TSActiveStepEnumH9.Int verify(~LedG & ~LedR & ~LedY)			
— VerifyGateClose when TSActiveStepIn == TSActiveStepEnumH9.GateClose verify(~LedG & ~LedR & ~LedY)			
— VerifyGateOpening when TSActiveStepIn == TSActiveStepEnumH9.GateOpening verify(~LedG & ~LedR & ~LedY)			
VerifyGateOpening1 when TSActiveStepIn == TSActiveStepEnumH9.GateOpening1			
— VerifyGateOpening_1 verify(~LedG & ~LedR & ~LedY)	1 after(2, sec)	VerifyGateOpening1_2	▼
— VerifyGateOpening_2 verify(~LedG & ~LedR & LedY)	1 after(2, sec)	VerifyGateOpening1_1	▼
— VerifyGateOpened when TSActiveStepIn == TSActiveStepEnumH9.GateOpened verify(LedG & LedR & LedY)			
VerifyGateOpenedWithObstacle when TSActiveStepIn == TSActiveStepEnumH9.GateOpenedWithObstacle verify(LedG & LedR & LedY)			
VerifyObcOpening when TSActiveStepIn == TSActiveStepEnumH9.GateObstacleOpening verify(LedG & LedR & LedY)			
VerifyObcOpening1 when TSActiveStepIn == TSActiveStepEnumH9.GateObstacleOpening1			
— VerifyObcOpening_1 verify(~LedG & LedR & LedY)	1 after(1, sec)	VerifyObcOpening1_2	▼
— VerifyObcOpening_2 verify(LedG & LedR & LedY)	1 after(1, sec)	VerifyObcOpening1_1	▼
— VerifyObcOpening2 when TSActiveStepIn == TSActiveStepEnumH9.GateObstacleOpening2 verify(LedG & LedR & LedY)			
— VerifyObcOpening3 when TSActiveStepIn == TSActiveStepEnumH9.GateObstacleOpening3 verify(LedG & LedR & LedY)			
— VerifyObcOpening3_1 verify(~LedG & LedR & LedY)	1 after(1, sec)	VerifyObcOpening3_2	▼
— VerifyObcOpening3_2 verify(LedG & LedR & LedY)	1 after(1, sec)	VerifyObcOpening3_1	▼
— VerifyGateOpened when TSActiveStepIn == TSActiveStepEnumH9.GateOpened1 verify(LedG & LedR & LedY)			
VerifyGateClosing when TSActiveStepIn == TSActiveStepEnumH9.GateClosing			
— VerifyGateClosing_1 verify(~LedG & ~LedR & ~LedY)	1 after(2, sec)	VerifyGateClosing_2	▼
— VerifyGateClosing_2 verify(~LedG & ~LedR & LedY)	1 after(2, sec)	VerifyGateClosing_1	▼
— VerifyGateClosed when TSActiveStepIn == TSActiveStepEnumH9.GateClosed verify(~LedG & ~LedR & ~LedY)			
step_1_2			

Figure 6.17: Test Assessment

CAPITOLO 7

IMPLEMENTAZIONE

In questo capitolo si presenterà il codice generato da Matlab, esportato tramite l'add-on Embedded Coder ed il successivo caricamento del firmware sulla scheda STM32 Nucleo G474RE.

7.1 Realizzazione del circuito

I componenti utilizzati per realizzare il circuito sono i seguenti:

Componente	Pin
B1	PC8
B2	PC6
B3	PC5
P1	PC2
P2	PC3
LedG	PB5
LedY	PB4
LedR	PB3

Table 7.1: Tabella variabili Tuning

In figura 7.1 è presente il circuito realizzato sul software Fritzing.

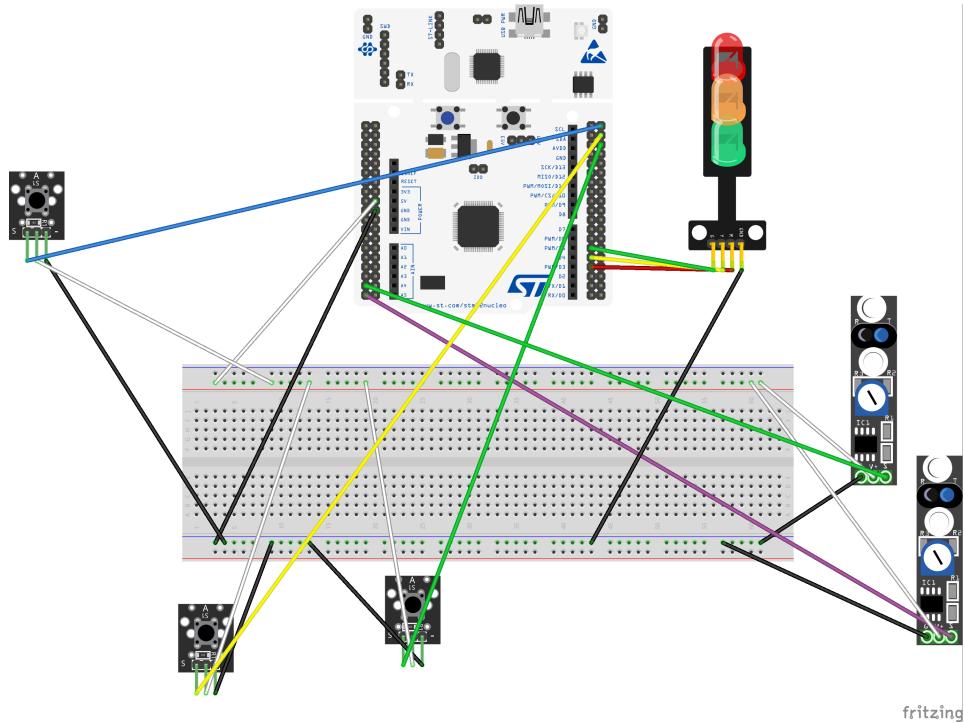


Figure 7.1: Circuito su Fritzing

7.2 Codice Sorgente

Per generare il codice è stato utilizzato l'add-on **Embedded Coder**, il quale ha permesso la creazione di tutte le funzioni e variabili necessarie per realizzare l'automa. La creazione del codice sorgente tramite Embedded Coder è una metodologia avanzata che consente di generare codice C e C++ ottimizzato direttamente da modelli Simulink e Stateflow. Questo processo automatizzato non solo accelera lo sviluppo del software embedded, ma riduce anche la probabilità di errori manuali. Embedded Coder offre strumenti potenti per la configurazione del codice generato, permettendo di personalizzare le proprietà e le specifiche per soddisfare requisiti specifici dell'applicazione. Inoltre, supporta l'integrazione con ambienti di sviluppo integrato (IDE) e toolchain per microcontrollori, facilitando il deployment su hardware target.

Il codice generato è stato importato nell'IDE **STM32CubeIDE**, dove è stato creato il file main per collegare i componenti fisici della scheda con le variabili create da MATLAB. Questo processo garantisce una sinergia ottimale tra il codice generato automaticamente e l'hardware, assicurando che tutti i sensori, attuatori e altri dispositivi siano correttamente interfacciati e operativi.

7. IMPLEMENTAZIONE

```
#include "main.h"
#include "gpio.h"
#include "Chart.h"

void SystemClock_Config(void);

static void Chart_read_inputs(){
    rtU.B1 = HAL_GPIO_ReadPin(B1_GPIO_Port, B1_Pin);
    rtU.B2 = HAL_GPIO_ReadPin(B2_GPIO_Port, B2_Pin);
    rtU.B3 = HAL_GPIO_ReadPin(B3_GPIO_Port, B3_Pin);
    rtU.P1 = !HAL_GPIO_ReadPin(P1_GPIO_Port, P1_Pin);
    rtU.P2 = !HAL_GPIO_ReadPin(P2_GPIO_Port, P2_Pin);
}

static void Chart_write_outputs(){
    HAL_GPIO_WritePin(LedG_GPIO_Port, LedG_Pin, rtY.LedG);
    HAL_GPIO_WritePin(LedY_GPIO_Port, LedY_Pin, rtY.LedY);
    HAL_GPIO_WritePin(LedR_GPIO_Port, LedR_Pin, rtY.LedR);
}

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    Chart_initialize();
    while (1)
    {
        uint32_t elapsed, start;
        start = HAL_GetTick();
        Chart_read_inputs();
        Chart_step();
        Chart_write_outputs();
        elapsed = HAL_GetTick() - start;
        HAL_Delay(100-elapsed);
    }
}
```

Listing 1: Main

INDICE DELLE FIGURE

2.1	Apertura Cancello	12
2.2	Chiusura Cancello	13
2.3	Regolazioni	14
2.4	Controllo Ostacolo e Gestione Richieste	15
2.5	Determinazione Stato	16
2.6	Stato di Errore	17
2.7	Chiusura Automatica	17
2.8	Use Cases Generale	18
3.1	Scenario 1	21
3.2	Scenario 2	23
3.3	Scenario 3	25
3.4	Scenario 4	26
3.5	Scenario 5	28
4.1	State Diagram	31
5.1	Stato Cancello	38
5.2	Stati Regolazioni	41
5.3	Stato Ostacolo	43
5.4	Stati Buttons	45
5.5	Final Chart	46
5.6	Input e Output	47
6.1	Esempio Test Harness	48
6.2	Test Sequence	49

INDICE DELLE FIGURE

6.3	Test Assessment	49
6.4	Test Sequence	50
6.5	Test Assessment	50
6.6	Test Sequence	51
6.7	Test Assessment	51
6.8	Test Sequence	52
6.9	Test Assessment	52
6.10	Test Sequence	53
6.11	Test Assessment	53
6.12	Test Sequence	54
6.13	Test Assessment	54
6.14	Test Sequence	55
6.15	Test Assessment	55
6.16	Test Sequence	56
6.17	Test Assessment	57
7.1	Circuito su Fritzing	59