

# Relazione Progetto Sistemi Embedded

Gruppo 1 Canale I-Z

06 June 2024



**Università degli Studi di Salerno**

Antonio	Sessa	0622702305	a.sessa108@studenti.unisa.it
Angelo	Molinario	0622702311	a.molinario3@studenti.unisa.it
Massimiliano	Ranauro	0622702373	m.ranauro2@studenti.unisa.it
Pietro	Martano	0622702402	p.martano@studenti.unisa.it

# **1 WP1**

## **1.1 Requisiti**

### **1.1.1 Requisiti nello stato cancello chiuso**

CLOSED.1 Quando il sensore di chiusura P2 è attivo significa che il cancello è chiuso.

CLOSED.2 Quando il cancello è chiuso, se premo il pulsante B1 e il sensore di presenza P1 non rileva alcun ostacolo, inizia la fase di apertura del cancello.

CLOSED.3 Quando il cancello è chiuso, se premo il pulsante B2, imposta il tempo di chiusura automatica al tempo di chiusura automatica precedente più dieci. Se il tempo di chiusura eccede 120, imposta a 10.

CLOSED.4 Quando il cancello è chiuso, se premo il pulsante B3, imposta il tempo di lavoro al tempo di lavoro precedente più dieci. Se il tempo di lavoro eccede 120, imposta a 10.

CLOSED.5 Quando il cancello è chiuso, e il sensore P1 risulta attivo, la pressione del pulsante B1 fa lampeggiare il Led VERDE per un tempo di 30 secondi con frequenza pari a 1 Hz.

CLOSED.6 Quando il cancello è chiuso, i Led VERDE, GIALLO e ROSSO del sistema sono spenti.

### **1.1.2 Requisiti nello stato cancello aperto**

OPEN.1 Quando il cancello è aperto, se premo il pulsante B1 e il sensore di presenza P1 non rileva alcun ostacolo, inizia la fase di chiusura del cancello

OPEN.2 Quando il cancello è aperto, e il sensore P1 risulta attivo, la pressione del pulsante B1 fa lampeggiare il Led VERDE per un tempo di 30 secondi con frequenza pari a 1 Hz.

OPEN.3 Quando il cancello è aperto, i Led VERDE, GIALLO e ROSSO del sistema sono accesi fissi.

OPEN.4 Se il cancello rimane aperto per un tempo pari al tempo di chiusura automatica, inizia la fase di chiusura del cancello.

### **1.1.3 Requisiti nella chiusura del cancello**

CLOSING.1 Quando il cancello è in chiusura, se premo il pulsante B1, inizia la fase di apertura del cancello.

CLOSING.2 Quando il cancello è in chiusura, se il sensore P1 rileva un ostacolo, inizia la fase di apertura del cancello.

CLOSING.3 Quando il cancello è in chiusura, se il tempo di lavoro è terminato e il sensore di chiusura P2 è inattivo, entro in uno stato di errore.

CLOSING.4 Quando il cancello è in chiusura, il Led GIALLO lampeggia con frequenza 0,5 Hz.

CLOSING.5 Quando accendo il sistema, se il sensore di chiusura P2 è inattivo, inizia la chiusura del cancello.

### **1.1.4 Requisiti nella apertura del cancello**

OPENING.1 Quando il cancello è in apertura, se premo il pulsante B1 e il sensore di presenza P1 non rileva alcun ostacolo, inizia la fase di chiusura del cancello.

OPENING.2 Quando il cancello è in apertura, il Led GIALLO lampeggia con frequenza 0,5 Hz.

### **1.1.5 Requisiti stato di errore**

ERR.1 Quando il cancello è in chiusura, se il tempo di lavoro è terminato e il sensore di chiusura P2 è inattivo, entro in uno stato di errore.

ERR.2 Quando mi trovo nello stato di errore da più di 10 secondi allora accendo il Led ROSSO.

ERR.3 Quando mi trovo nello stato di errore, se rilevo che il sensore di chiusura P2 è attivo, entro nello stato di cancello chiuso.

## 1.2 Diagrammi

### 1.2.1 Aggiornamento tempo di lavoro

Il seguente diagramma di attività illustra il modo in cui l'utente, interagendo con il pulsante B2, aggiorna il tempo di lavoro del cancello. Il tempo di lavoro può essere aggiornato solo nello stato di cancello chiuso, quindi alla pressione del pulsante B2 viene verificato lo stato del sensore P2. Qualora il sensore risultasse attivo (il che indica che il cancello è chiuso), il tempo di lavoro viene aggiornato secondo la formula:  $(\text{Old mod } 120) + 10$ .

Inoltre, esiste un meccanismo analogo per il pulsante B3, che modifica il tempo di chiusura automatica.

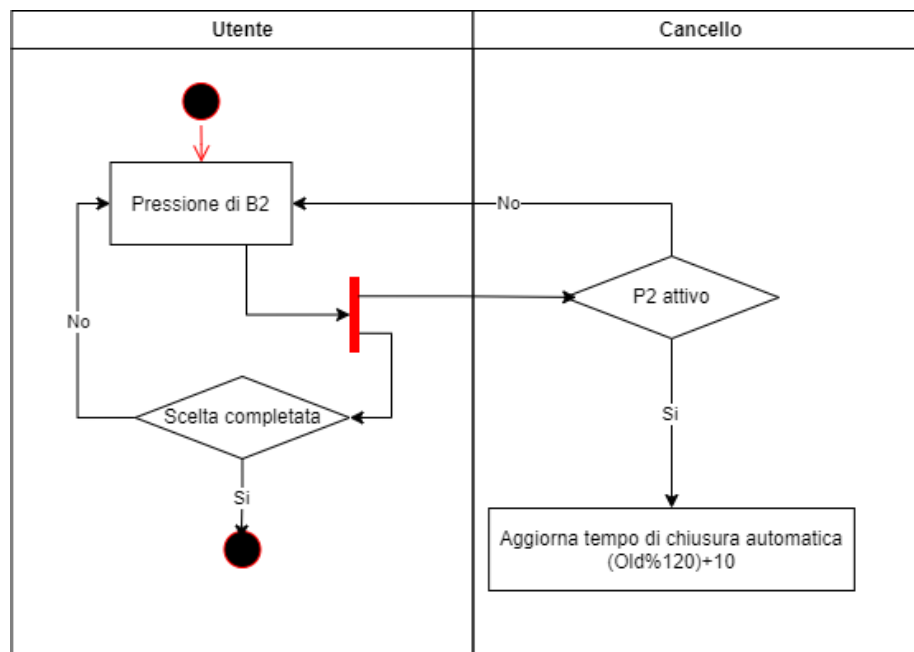


Figure 1: Activity Diagram, Aggiornamento del tempo di lavoro

### 1.2.2 Errore nella chiusura del cancello

Il seguente diagramma di attività illustra come il sistema possa entrare e uscire dallo stato di errore. Al termine della fase di chiusura del cancello, il sensore P2 verifica la completa chiusura dello stesso. In caso contrario, al termine del tempo di lavoro, si esegue il diagramma di attività relativo allo stato di errore, il quale, trascorsi 10 secondi, accenderà il LED ROSSO. È possibile uscire dallo stato di errore chiudendo manualmente il cancello o reinizializzando il sistema.

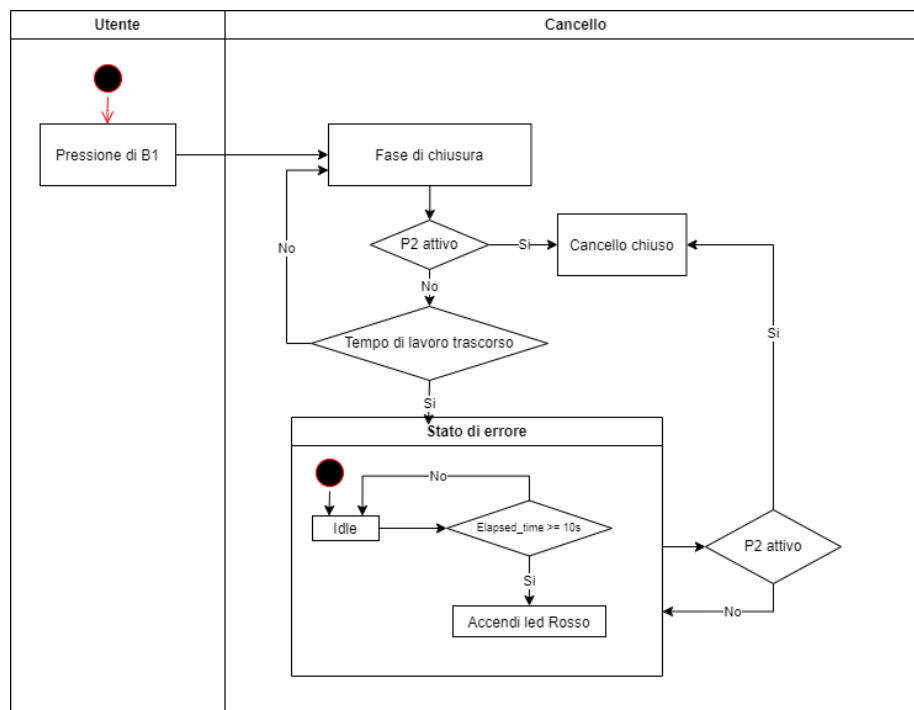


Figure 2: Activity Diagram, Errore nella chiusura del cancello

### 1.2.3 Attività di chiusura automatica

Il seguente diagramma di attività illustra la funzionalità di chiusura automatica del cancello. All'interno del diagramma supponiamo di trovarci in uno stato in cui il cancello è chiuso e l'utente preme il pulsante B1 per richiederne l'apertura. Dopo essere trascorso il tempo di lavoro il sistema si porta nello stato di cancello aperto dove permane per un tempo pari al tempo di chiusura automatica, trascorso il quale viene avviata la chiusura del cancello.

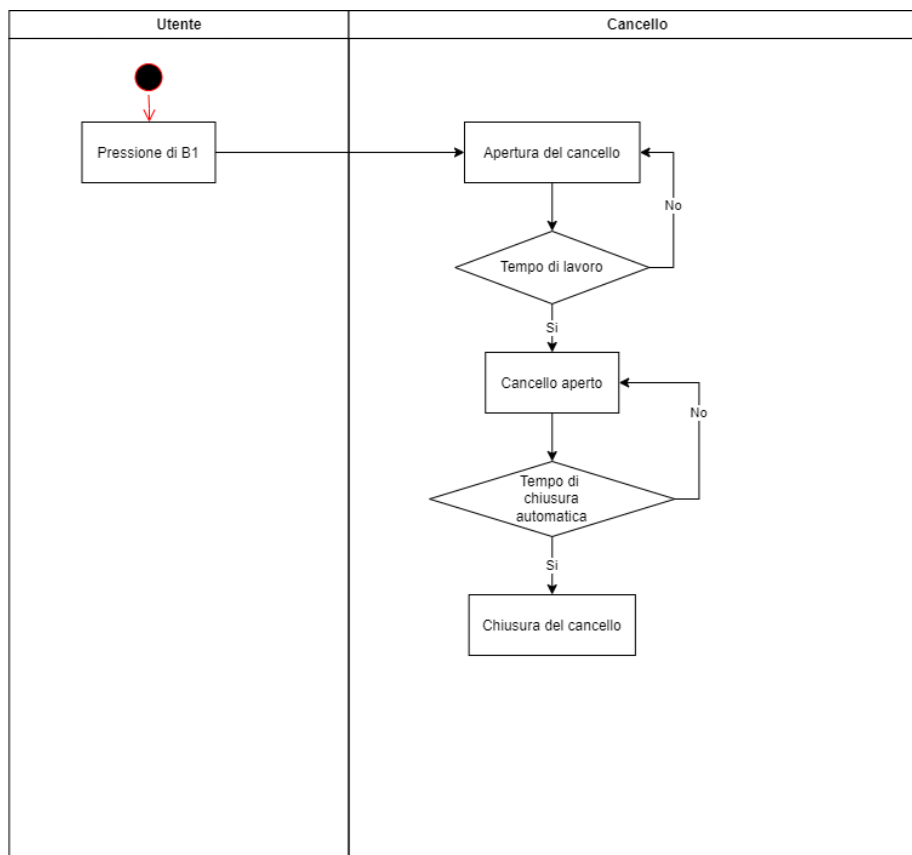


Figure 3: Activity Diagram, Chiusura automatica

### 1.2.4 Esecuzione Standard

Il seguente diagramma di attività illustra il flusso normale di esecuzione del sistema. Quest'ultimo rimane in uno stato di attesa fino alla pressione del pulsante B1. In assenza di una lettura alta sul sensore P1, tale azione provocherà l'apertura/chiusura del cancello. Durante queste fasi, il LED GIALLO lampeggia. Inoltre, eventuali attivazioni del sensore P1 durante la fase di chiusura interromperanno tale fase, avviando l'apertura del cancello.

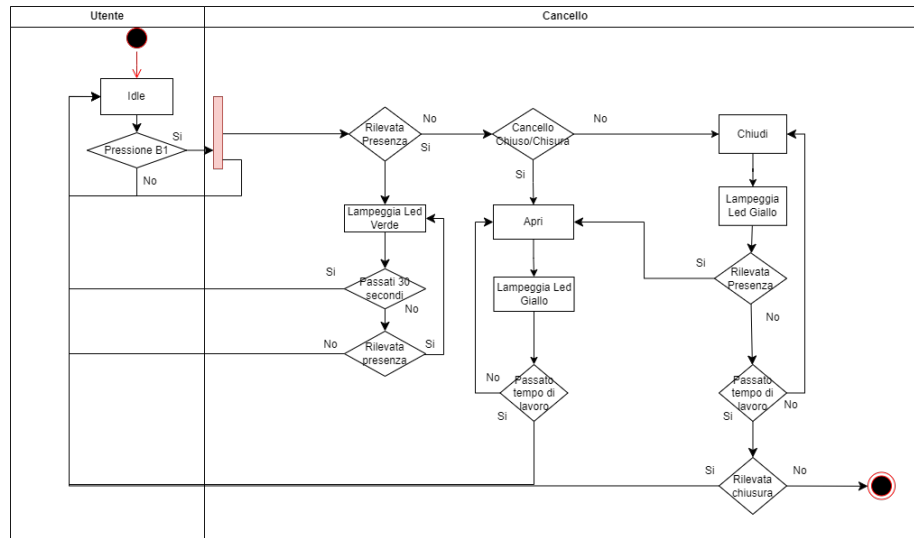


Figure 4: Activity Diagram, Esecuzione standard



### 1.2.5 Casi d'uso

Il seguente diagramma dei casi d'uso illustra un tipico scenario d'utilizzo del sistema da parte di un utente. L'utente può interfacciarsi con il sistema tramite tre bottoni. Uno scenario d'utilizzo del sistema prevede che l'utente, tramite i bottoni B2 e B3, regoli o meno i tempi di chiusura automatica e di lavoro. Quando l'utente ritiene soddisfacente la configurazione ottenuta, può richiedere l'apertura o la chiusura del cancello tramite la pressione del bottone B1.

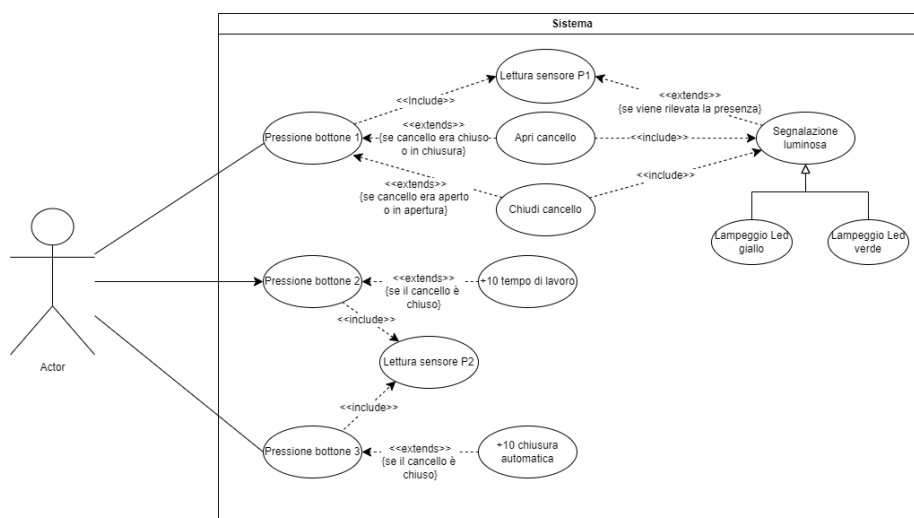


Figure 5: Use case diagram

### 1.2.6 State flow

Lo state flow sottostante descrive il comportamento del sistema in risposta ad eventi.

Nomenclatura delle variabili di ingresso:

- B1: pulsante per l'apertura e la chiusura del cancello
- B2: pulsante per la regolazione del tempo di chiusura automatica
- B3: pulsante per la regolazione del tempo di lavoro
- P1: sensore di presenza
- P2: sensore di chiusura

Nomenclatura delle variabili locali e di uscita:

- LR: led di colore rosso (red)
- LY: led di colore giallo (yellow)
- LG: led di colore verde (green)
- T2: tempo di chiusura automatica (si suppone di default a 10 secondi)
- T3: tempo di lavoro (si suppone di default a 10 secondi)

Nello schema sottostante quando si fa riferimento alla pressione di un qualsiasi pulsante si suppone che il pulsante sia considerato premuto solamente sul proprio fronte di salita/discesa. In questo modo nello schema non si tiene conto della pressione e del rilascio del pulsante riuscendo ad avere una rappresentazione più compatta.

Le variabili locali **T2** e **T3** rappresentano rispettivamente il tempo di chiusura automatica e il tempo di lavoro selezionati dall'utente. Le condizioni contrassegnate da **after(N s)**, dove **N** è una costante numerica o una variabile locale, diventano vere quando è trascorso almeno il tempo indicato da **N**.



## 2 WP2

### 2.1 Stateflow

L'implementazione della Figura 6 è stata effettuata tramite l'utilizzo del linguaggio grafico Stateflow messo a disposizione da MatLab. La traduzione dello state diagram è stata riportata fedelmente. Le operazioni di compattazione effettuate nello state diagram sono state rese esplicite, in particolare:

- La pressione di un pulsante è verificata solamente sul fronte di salita
- Il lampeggio di un led è implementato tramite il passaggio tra due stati differenti (uno dove il led è attivo e uno dove il led è inattivo)

Per semplificare la lettura del modello sono stati creati più stati concorrenti tra loro. Infatti è presente uno stato il quale ne ingloba tre che ha il solo scopo di verificare quando un pulsante ha subito un fronte di salita e inviare l'evento associato a tutti gli stati restanti del modello. All'interno dello stato nel quale il cancello è chiuso sono presenti tre stati tra loro concorrenti: due modificano il rispettivamente il tempo di chiusura automatica e il tempo di lavoro, l'altro si occupa delle operazioni inerenti effettivamente lo stato di cancello chiuso.

Sia gli input sia gli output del sistema sono gli stessi dello state flow. In aggiunta sono presenti tre eventi locali:

- B1\_pressed: fronte di salita del pulsante B1
- B2\_pressed: fronte di salita del pulsante B2
- B3\_pressed: fronte di salita del pulsante B3

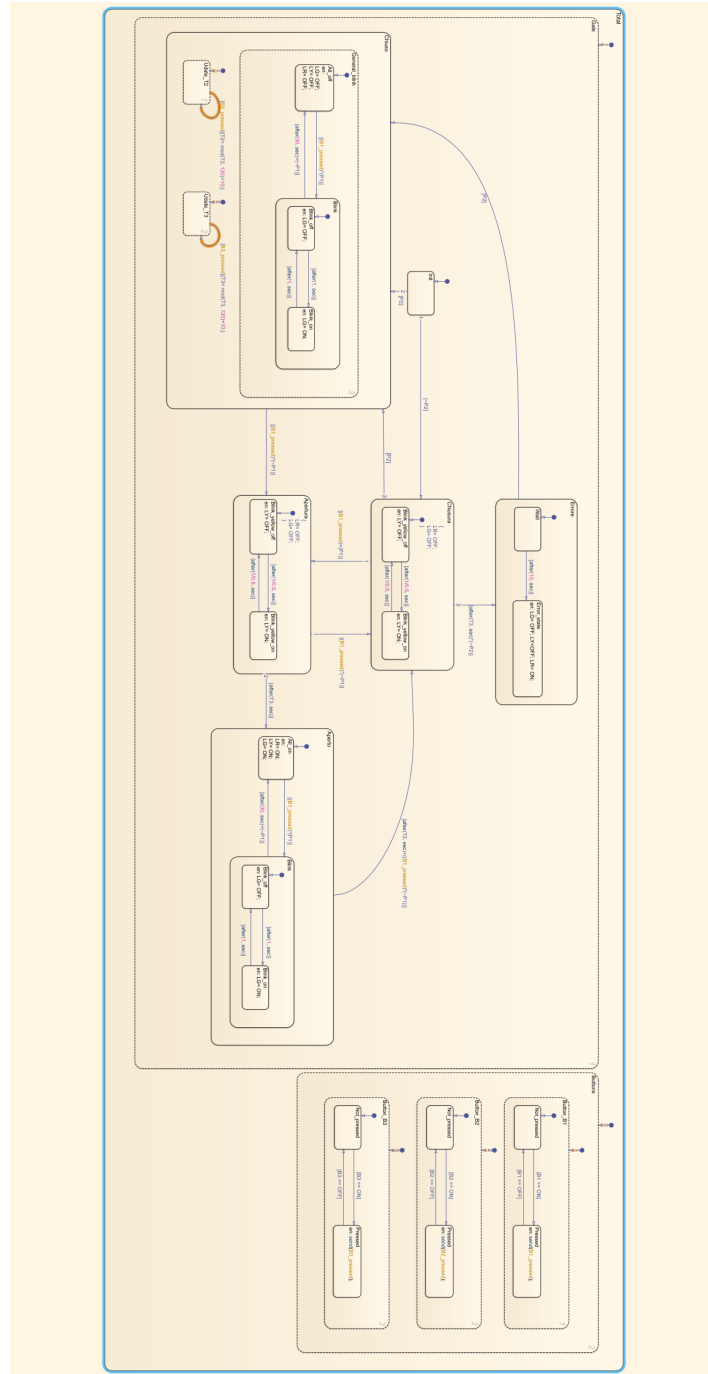


Figure 7: Stateflow

## 2.2 Test effettuati

### 2.2.1 Closure failure

Nello scenario corrente è stato verificato il corretto funzionamento del lampeggio led giallo in fase di apertura e chiusura, e l'ingresso nello stato di errore secondo la condizione del modello.

Step	Transition	Next Step	Description
Run  %% Initialize data outputs. B1 = false; B2 = false; B3 = false; P1 = false; P2 = true;	1. true	Apri ▼	Iniziamo con il cancello chiuso
Apri B1 = true; P2 = false;	1. true	Apertura ▼	Apriamo il cancello
Apertura B1 = false;	1. after(10,sec)	Check... ▼	
BlinkOff verify(LY == 0);	1. after(2,sec)	BlinkOn ▼	
BlinkOn verify(LY == 1);	1. after(2,sec)	BlinkOff ▼	
CheckApertura verify(LY == 1); verify(LR == 1); verify(LG == 1);	1. after(10,sec)	Chiusura ▼	Controlliamo l'avvenuta apertura
Chiusura	1. after(10,sec)	WaitFo... ▼	Dopo 10 secondi inizia a chiudersi
BlinkOff verify(LY == 0);	1. after(2,sec)	BlinkOn ▼	
BlinkOn verify(LY == 1);	1. after(2,sec)	BlinkOff ▼	
WaitForErr verify(LR == 0); verify(LG == 0); verify(LY == 0);	1. after(10,sec)	CheckErr ▼	
CheckErr verify(LY == 0); verify(LG == 0); verify(LR == 1);			Dopo altri 10 secondi controlliamo che viene rilevato l'errore dato che non è stato ricevuto il segnale P2

Figure 8: Test fallimento chiusura

### 2.2.2 Green blink from open

Nello scenario corrente è stato verificato il corretto funzionamento del lampeggio led verde in fase di cancello aperto quando si prova a premere il pulsante B1 con il sensore P1 alto. Il lampeggio permane finché non sono passati 30 secondi oppure quando il sensore P1 diventa basso oppure se il tempo di chiusura automatica scade (in questo caso si passa allo stato di chiusura). Per questo motivo il tempo di chiusura automatica è stato aumentato in modo da mantenere il cancello nello stato aperto. È stata verificata la condizione di uscita sia dopo i 30 secondi sia quando il sensore P1 diventa basso.

Step	Transition	Next Step
<div> <div>Stato_Aperto</div> <div>           Setup            B1 = false;            B2 = false;            B3 = false;            P1 = false;            P2 = true;            T2counter = 0;         </div> </div>	1. after(120,sec)	end ▼
<div> <div>AumentaT2</div> <div>           AumentaT2_2            B2 = false;         </div> </div>	1. true	AumentaT2 ▼
<div> <div>AumentaT2_1</div> <div>           B2 = true;            T2counter = T2counter + 1;         </div> </div>	1. true	AumentaT2_2 ▼
<div> <div>Apri</div> <div>           B1 = true;         </div> </div>	1. after(11,sec)	CheckApertura ▼
<div> <div>CheckApertura</div> <div>           B1 = false;            verify(LY == 1);            verify(LG == 1);            verify(LR == 1);         </div> </div>	1. true	PressioneWithP1 ▼
<div> <div>PressioneWithP1</div> <div>           P1 = true;            B1 = true;         </div> </div>	1. true	CheckAfter30 ▼





Step	Transition	Next Step	Description
<div> <div>Stato_Chiuso</div> <div> <div>Run</div> <pre>%% Initialize data outputs. B1 = true; B2 = false; B3 = false; P1 = true; P2 = true;</pre> </div> </div>	1. true	Prep	Andiamo in stato chiusura
<div> <div>Prep</div> <pre>P2 = true; P1 = true; B1 = true;</pre> </div>	1. true	CheckAfter30	Pressione del Bottone B1 quando c'è presenza P1
<div> <div>CheckAfter30</div> <pre>B1 = false;</pre> </div>	1. after(30,sec)	EndAfter30	Check blinking per 30 secondi
<div> <div>CheckAfter30_2</div> <pre>verify(LG == 0);</pre> </div>	1. after(1,sec)	CheckAfter30_1	
<div> <div>CheckAfter30_1</div> <pre>verify(LG == 1);</pre> </div>	1. after(1,sec)	CheckAfter30_2	
<div> <div>EndAfter30</div> <pre>verify(LG == 0); B1 = true;</pre> </div>	1. true	EndWithP1	Ripressione Bottone B1 con presenza ancora attiva
<div> <div>EndWithP1</div> </div>	1. after(15,sec)	P1off	
<div> <div>CheckAfter30_2</div> <pre>verify(LG == 0);</pre> </div>	1. after(1,sec)	CheckAfter30_1	
<div> <div>CheckAfter30_1</div> <pre>verify(LG == 1);</pre> </div>	1. after(1,sec)	CheckAfter30_2	
<div> <div>P1off</div> <pre>P1 = false;</pre> </div>	1. true	CheckEnd	Simuliamo la scomparsa della presenza dopo 15 secondi
<div> <div>CheckEnd</div> <pre>verify(LG == 0); verify(LR == 0); verify(LY == 0);</pre> </div>	1. true	P1off	Controlliamo l'interruzione del blinking

Figure 10: Test lampeggio led verde da chiuso

#### 2.2.4 Init closure

Nello scenario corrente è stato verificato il corretto funzionamento del sistema quando viene acceso. È stato supposto che all'accensione del sistema il cancello sia aperto e che quindi debba essere chiuso. Viene verificato il lampeggio del led giallo durante la fase di chiusura, l'effettiva chiusura del cancello quando il sensore P2 diventa alto e il non passaggio allo stato di errore dopo 10 secondi dalla chiusura del cancello.

Step	Transition	Next Step
<b>Run</b>  %% Initialize data outputs B1 = false; B2 = false; B3 = false; P1 = false; P2 = false;	1. after(0.1,sec)	false ▼
false P2 = false;	1. true	Chiusura ▼
<div> <div></div> <div>Chiusura</div> </div>	1. after(10,sec)	CheckChiusura ▼
<div> <div></div> <div>CheckBlinkOff verify(LY == 0);</div> </div>	1. after(2,sec)	ChekBlinkOn ▼
<div> <div></div> <div>ChekBlinkOn verify(LY == 1);</div> </div>	1. after(2,sec)	CheckBlinkOff ▼
CheckChiusura P2 = true; verify(LY == 0); verify(LR == 0); verify(LG == 0);	1. after(10,sec)	CheckNoErr ▼
CheckNoErr verify(LR == 0);		

Figure 11: Test chiusura cancello stato iniziale

### 2.2.5 Reset T2/T3

Nello scenario corrente è stato verificato il corretto funzionamento dell'incremento dei valori T2 (tempo di chiusura automatica) e T3 (tempo di lavoro). L'incremento viene effettuato 13 volte in modo tale che il valore finale coincida con quello iniziale, questo per verificare che l'operazione "modulo 120" funzioni correttamente. Dopo la modifica si effettua l'apertura del cancello seguita dalla sua automatica chiusura. Durante entrambi queste fasi viene verificato che il lampeggio del led giallo non duri più del tempo presumibilmente assunto da T2 e T3, e che il cancello si sia effettivamente aperto

nella prima fase e chiuso nella seconda.

Step	Transition	Next Step	Description
<b>Run</b> <pre> %% Initialize data outputs. B1 = false; B2 = false; B3 = false; P1 = false; P2 = true; NumPress2 = 0; NumPress3 = 0; </pre>	1. true	AumentoT2	Iniziamo con il cancello chiuso
<b>AumentoT2</b> <pre> PressioneB2 NumPress2 = NumPress2 + 1; B2 = true; </pre>	1. NumPress2 == 12	AumentoT3	Premiamo 13 volte B2, T2 dovrebbe diventare pari a 10
<pre> RilascioB2 B2 = false; </pre>	1. true	RilascioB2	
<b>AumentoT3</b> <pre> PressioneB3 NumPress3 = NumPress3 + 1; B3 = true; </pre>	1. NumPress3 == 12	PressioneB1	Premiamo 13 volte B3, T2 dovrebbe diventare pari a 10
<pre> RilascioB3 B3 = false; </pre>	1. true	RilascioB3	
<pre> PressioneB1 B1 = true; </pre>	1. true	PressioneB3	
<b>Apertura</b> <pre> Apertura_2 verify(LY == false) </pre>	1. after(10,sec)	Apertura_1	Controlliamo che l'apertura impiega 10 secondi
<pre> Apertura_1 verify(LY == true) </pre>	1. after(2,sec)	Apertura_2	
<pre> CheckApertura P2 = false; verify(LR == true); verify(LG == true); verify(LY == true); </pre>	1. after(2,sec)	CheckChiusura...	
<b>CheckChiusuraAutomatica</b> <pre> CheckChiusuraAutomatica_2 verify(LY == false) </pre>	1. after(10,sec)	CheckChiusura...	Controlliamo che dopo 10 secondi inizia la chiusura automatica
<pre> CheckChiusuraAutomatica_1 verify(LY == true) </pre>	1. after(2,sec)	CheckChiusura...	
<pre> CheckChiusura P2 = true; verify(LR == false); verify(LG == false); verify(LY == false); </pre>	1. true	End	Controlliamo che la chiusura impiega 10 secondi
End			

Figure 12: Test aumento T2 e T3

## 2.2.6 Switch open-close

Nello scenario corrente è stato verificato il corretto funzionamento dell'apertura del cancello con la pressione del pulsante B1 mentre quest'ultimo si sta chiudendo. Quando il sistema viene acceso lo

si porta nello stato di chiusura. Dopo aver verificato che, per un certo periodo, si trova nello stato di chiusura, si effettua l'apertura del cancello premendo il pulsante B1 e si attende che sia passato il tempo sufficiente per verificare che il cancello si sia effettivamente aperto. Durante l'attesa viene verificato che il led giallo lampeggi.

Step	Transition	Next Step	Description
Run %% Initialize data outputs B1 = false; B2 = false; B3 = false; P1 = false; P2 = false;	1. after(0.1,sec)	setfalse	Iniziamo la chiusura
setfalse P2 = false;	1. true	Chisura	
Chisura	1. after(5,sec)	PressioneB1	Essendo P2 basso, inizia la chiusura
Chisura_2 verify(LY == 0);	1. after(2,sec)	Chisura_1	
Chisura_1 verify(LY == 1);	1. after(2,sec)	Chisura_2	
PressioneB1 B1 = true;	1. true	Apertura	Dopo 5 secondi mentre si sta chiudendo premo il bottone B1
Apertura	1. after(10,sec)	CheckAperto	
CheckAperturaBlinkOn verify(LY == 0);	1. after(2,sec)	CheckAperturaBlinkOn	
CheckAperturaBlinkOff verify(LY == 1);	1. after(2,sec)	CheckAperturaBlinkOff	
CheckAperto verify(LY == 1); verify(LR == 1); verify(LG == 1);	1. true	End	Dopo 10 secondi verifico che mi trovo nello stato di aperto
End			

Figure 13: Test apertura durante la chiusura

### 2.2.7 T2 increase

Nello scenario corrente è stato verificato il corretto funzionamento della chiusura automatica quando il valore di T2 (tempo di chiusura automatica) viene incrementato. All'accensione del sistema il cancello è chiuso. In questo modo è possibile aumentare T2 e verificare che il valore presumibilmente assunto sia stato effettivamente assunto e rispettato. Con la pressione del pulsante B1 si porta il cancello nello stato di apertura, si verifica che durante questa fase

il led giallo lampeggi, e si verifica che dopo il tempo T3 (tempo di lavoro) il cancello sia effettivamente aperto. Si attende il tempo T2 e si verifica che il sistema entri nello stato di chiusura del cancello. Le verifiche in questo stato sono che il led giallo lampeggi e che il cancello sia chiuso (per farlo si imposta manualmente P2 alto).

Step	Transition	Next Step	Description
Init %% Initialize data outputs. B1 = false; B2 = false; B3 = false; P1 = false; P2 = true;	1. true	AumentaT2	Inizia con cancello chiuso
AumentaT2	1. after(5,sec)	PressioneB1	Portiamo il tempo di chiusura automatica a 30 sec
RilascioB2 B2 = false;	1. after(1,sec)	PressioneB2	
PressioneB2 B2 = true;	1. after(1,sec)	RilascioB2	
PressioneB1 B1 = true;	1. true	Apertura	Apriamo il cancello
Apertura	1. after(10,sec)	CheckApertura	Per controllora che l'apertura dura 10 secondi (il tempo default), controlliamo il blinking del led giallo per 10 secondi
AperturaBlinkOff verify(LY == false)	1. after(2,sec)	AperturaBlinkOn	
AperturaBlinkOn verify(LY == true)	1. after(2,sec)	AperturaBlinkOff	
CheckApertura P2 = false; verify(LR == true); verify(LG == true); verify(LY == true);	1. after(30,sec)	Chiusura	Controlliamo l'avvenuta apertura, segnalata dai tre led accesi. Controlliamo la persistenza in questo stato con i tre led accesi per 30 secondi
Chiusura	1. after(10,sec)	CheckChiusura	Controlliamo che dopo 30 secondi è iniziata la chiusura
ChiusuraBlinkOff verify(LY == false)	1. after(2,sec)	ChiusuraBlinkOn	
ChiusuraBlinkOn verify(LY == true)	1. after(2,sec)	ChiusuraBlinkOff	
CheckChiusura P2 = true; verify(LR == false); verify(LG == false); verify(LY == false);	1. true	End	Impostiamo P2 a vero, per simulare una corretta chiusura e terminare il test
End			

Figure 14: Test aumento tempo chiusura automatica

### 2.2.8 T3 increase

Nello scenario corrente è stato verificato il corretto funzionamento della apertura/chiusura quando il valore di T3 (tempo di lavoro) viene incrementato. All'accensione del sistema il cancello è chiuso.

In questo modo è possibile aumentare T3 e verificare che il valore presumibilmente assunto sia stato effettivamente assunto e rispettato. Con la pressione del pulsante B1 si porta il cancello nello stato di apertura, si verifica che durante questa fase il led giallo lampeggi e si verifica che dopo il tempo T3 il cancello sia effettivamente aperto. Si attende il tempo T2 (tempo di chiusura automatica) e si verifica che il sistema entri nello stato di chiusura del cancello. Le verifiche in questo stato sono che il led giallo lampeggi e che il cancello sia chiuso (per farlo si imposta manualmente P2 alto).

Step	Transition	Next Step	Description
<b>Init</b> %% Initialize data outputs. B1 = false; B2 = false; B3 = false; P1 = false; P2 = true;	1. true	AumentaT3 ▼	Partiamo da uno stato chiuso
AumentaT3 P2 = true;	1. after(5,sec)	PressioneB1 ▼	Portiamo il tempo di lavoro a 30 sec
RilascioB3 B3 = false;	1. after(1,sec)	PressioneB2 ▼	
PressioneB2 B3 = true;	1. after(1,sec)	RilascioB3 ▼	
PressioneB1 B1 = true;	1. true	Apertura ▼	Pressione del bottone 1 causa l'inizio dell'apertura
Apertura	1. after(30,sec)	CheckApertura ▼	Per controllora che l'apertura dura 30 secondi, controlliamo il blinking del led giallo per 30 secondi
AperturaBlinkOff verify(LY == false)	1. after(2,sec)	AperturaBlinkOn ▼	
AperturaBlinkOn verify(LY == true)	1. after(2,sec)	AperturaBlinkOff ▼	

CheckApertura P2 = false; verify(LR == true); verify(LG == true); verify(LY == true);	1. after(10,sec)	Chiusura ▼	Controlliamo l'avvenuta apertura, segnalata dai tre led accesi.
Chiusura	1. after(30,sec)	CheckChiusura ▼	Passato 10 secondi, il tempo di chiusura automatica, parte la chiusura.
ChiusuraBlinkOff verify(LY == false)	1. after(2,sec)	ChiusuraBlinkOn ▼	
ChiusuraBlinkOn verify(LY == true)	1. after(2,sec)	ChiusuraBlinkOff ▼	
CheckChiusura P2 = true; verify(LR == false); verify(LG == false); verify(LY == false);	1. true	End ▼	Impostiamo P2 a vero, per simulare una corretta chiusura e terminare il test
End			

Figure 15: Test aumento tempo di lavoro

### 2.2.9 Check obstacle always present

Nello scenario corrente viene testato il funzionamento del sistema quando viene introdotto un ostacolo durante la fase di chiusura e l'ostacolo presente per un periodo di tempo anche dopo che il cancello si è aperto. All'accensione del sistema il cancello entra nella fase di chiusura perchè il sensore di chiusura P2 non è attivo. Durante la fase di chiusura viene introdotto un ostacolo che porta il cancello dallo stato di chiusura allo stato di apertura. Dopo che è trascorso il tempo di lavoro il cancello si trova nello stato di apertura e il sensore P1 è ancora attivo. Il test prosegue verificando che quando il cancello è aperto e viene premuto il pulsante B1 il cancello non entra in fase di chiusura ma avvia il lampeggio del led verde per segnalare che non è possibile chiudere il cancello a causa della presenza dell'ostacolo. Successivamente viene rimosso l'ostacolo e viene richiesta nuovamente la chiusura del cancello che questa volta porta il cancello nello stato di chiusura fino a quando non viene rilevata la chiusura del cancello tramite il sensore P2 che indica il completamento della chiusura del cancello.

Step	Transition	Next Step	Description
Scenario_1			
Run %% Initialize data outputs. B1 = false; B2 = false; B3 = false; P1 = false; P2 = false;	1. P2==false	Chiusura	▼ Inizialmente il cancello non è chiuso
Chiusura verify(LR==0); verify(LG==0);	1. after(6,sec)	Ostacolo	▼ Resto nello stato di chiusura per 6 secondi, quindi meno del tempo di lavoro, per controllare che il led giallo lampeggi. Passati 6 secondi attivo il sensore P1 per rilevare un ostacolo.
Ostacolo P1=true;	1. B1==true    P1==true	Apertura	▼
Apertura verify(LR==0); verify(LG==0);	1. after(10,sec)	Aperto	▼ Dopo 10 secondi ( tempo di lavoro) il cancello è aperto
Apertura_blinkOff verify(LY==0);	1. after(2,sec)	Apertura_blinkOn	▼
Apertura_blinkOn verify(LY==1);	1. after(2,sec)	Apertura_blinkOff	▼
Aperto	1. after(8,sec)	RichiestaChiusuraNoOstacolo	▼ Dopo 8 secondi l'ostacolo viene rimosso e viene premuto B1
Aperto_led_verify verify(LR==1); verify(LG==1); verify(LY==1);	1. after(2,sec)	Press_B1	▼
Press_B1 B1 = true;	1. B1==true && P1==true	Aperto_blink	▼
Aperto_blink B1=false;	1. after(30,sec)	Press_B1	▼
Aperto_blink_off verify(LG==0);	1. after(1,sec)	Aperto_blink_on	▼
Aperto_blink_on verify(LG==1);	1. after(1,sec)	Aperto_blink_off	▼
RichiestaChiusuraNoOstacolo P1=false; B1=true;	1. P1==false && B1==true	Chiusura1	▼
Chiusura1 B1=false; P2=true;	1. P2==true	Chiuso	▼
Chiuso verify(LR==0); verify(LY==0); verify(LG==0);	1. true	End	▼
End			

Figure 16: Test Always Present

### 2.2.10 Check obstacle

Nello scenario corrente è stato verificato il corretto funzionamento dell'apertura del cancello quando si trova nella fase di chiusura e il sensore P1 è alto. All'accensione del sistema il cancello è aperto. Durante la fase di chiusura, prima che possa entrare nello stato di errore, il sensore P2 diventa alto. Si verifica che atteso il tempo di lavoro il cancello si trova nello stato aperto. Successivamente



viene portato nella fase di chiusura e si verifica che entra nello stato chiuso.

Step	Transition	Next Step	Description
Scenario_2			
Run %% Initialize data outputs. B1 = false; B2 = false; B3 = false; P1 = false; P2 = false;	1. P2==false	Chiusura	▼ Inizialmente il cancello non è chiuso
Chiusura verify(LR==0); verify(LG==0);	1. after(6,sec)	Ostacolo	▼ Resto nello stato di chiusura per 6 secondi, quindi meno del tempo di lavoro, per controllare che il led giallo lampeggi. Passati 6 secondi attivo il sensore P1 per rilevare un ostacolo.
Ostacolo P1=true;	1. B1==true    P1==true	Apertura	▼
Apertura verify(LR==0); verify(LG==0);	1. after(6,sec)	RichiestaChiusuraNoOstacolo	▼ Dopo 10 secondi ( tempo di lavoro) il cancello è aperto
Apertura_blinkOff verify(LY==0);	1. after(2,sec)	Apertura_blinkOn	▼
Apertura_blinkOn verify(LY==1);	1. after(2,sec)	Apertura_blinkOff	▼
RichiestaChiusuraNoOstacolo P1=false; B1=true;	1. B1==true && P1==false	Chiusura1	▼
Chiusura1 verify(LR==0); verify(LG==0); B1=false; P2=true;	1. P2==true	Chiuso	▼
Chiuso verify(LR==0); verify(LG==0); verify(LY==0);	1. true	End	▼
End			

Figure 17: Test ostacolo in chiusura

## 3 WP3

### 3.1 Circuito usato

Per l'implementazione del modello su un dispositivo hardware è stato usato l'applicazione Embedded Coder messo a disposizione da MatLab.

Il collegamento dei pin è stato effettuato nel seguente modo:

Componente	PIN
B1	PC13
B2	PC8
B3	PC9
P1	PC10
P2	PC11
LG	PB3
LY	PB5
LR	PB4

Il collegamento è effettuato come indicato dalla figura:

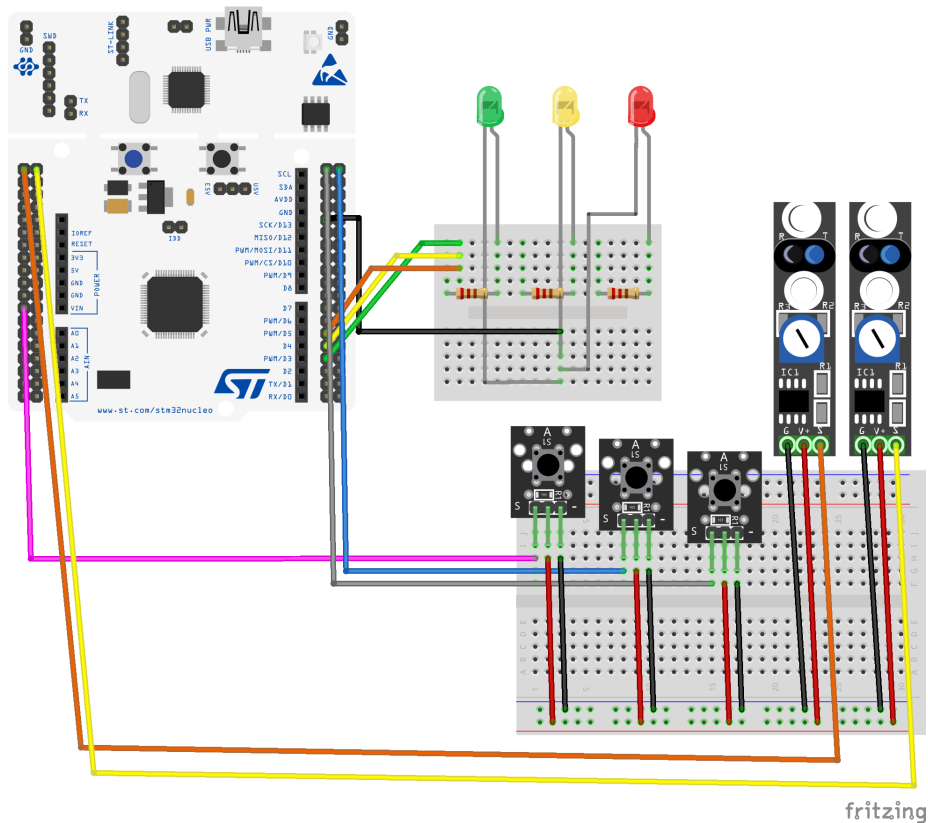


Figure 18: Circuito su NUCLEO-G474RE

### 3.2 Test effettuati sul dispositivo

Sul dispositivo sono stati effettuati tutti i test indicati nella sezione corrispondente del report sia altri test aggiuntivi per raggiungere lo statement coverage. In particolare è stato anche verificato:

- La partenza del sistema nello stato chiuso
- La partenza del sistema nello stato di chiusura
- L'apertura del cancello (lampeggio del led giallo)
- La chiusura del cancello (lampeggio del led giallo)
- Il cancello chiuso (tutti i led spenti)
- Il cancello aperto (tutti i led accesi)
- L'ingresso nello stato di errore (led rosso acceso)
- L'uscita dallo stato di errore
- Il lampeggio del led verde quando si provava a premere B1 con P1 alto negli stati aperto/chiuso
- Il cambio di modalità chiusura/apertura quando si preme B1
- Il cambio di modalità da chiusura a apertura quando P1 diventa alto