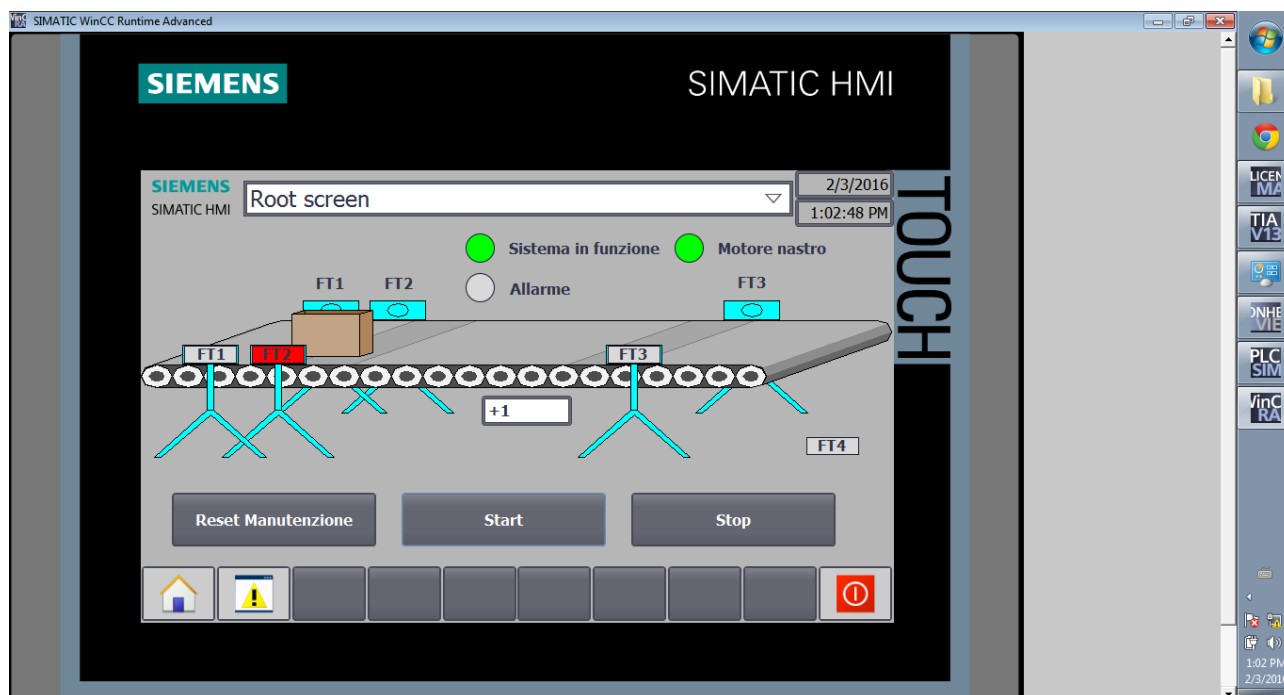


Nastro trasportatore con ribaltamento cassetta.

Il seguente progetto prevede il controllo di un nastro trasportatore (attraverso un PLC Siemens S7-1200 o S7-1500 programmato in LAD) in cui è presente un'isola ribaltatrice.



Il nastro trasportatore è munito di quattro fotocellule:

- **Fotocellula “FT1”**, situata nella parte iniziale, ha la funzione di rilevare quando una cassetta viene depositata nel nastro; se è presente almeno una cassetta allora viene avviato il motore del nastro trasportatore.
- **Fotocellula “FT2”**, immediatamente successiva a FT1, ha la funzione di:
 - contare il numero di cassette che transitano sul nastro incrementando il counter.
 - garantire che venga lasciata una certa distanza tra le cassette, (pertanto, se FT1 e FT2 sono attive contemporaneamente il nastro si ferma).
- **Fotocellula “FT3”**, posizionata in prossimità dell'isola ribaltamento, ha la funzione di:
 - fermare il nastro trasportatore quando l'oggetto raggiunge l'isola ribaltatrice
 - attivare un timer, il quale consente di impostare un tempo di attesa affinché le cassette non vengano ribaltate immediatamente.
- **Fotocellula “FT4”**, situata alla fine del nastro trasportatore, ha la funzione di verificare l'espulsione della cassetta dal nastro e decrementare il counter.

Ogni volta che una cassetta viene rilevata dalla fotocellula “FT3” parte un tempo di attesa, esaurito il quale si aziona l'isola ribaltatrice che solleva leggermente la cassetta fino a raggiungere un fincorsa intermedio (chiamato “FC_salita1”) che ferma il ribaltamento ed aziona una valvola per il vuoto che ha la funzione di aggrappare la cassetta.

Successivamente, quando viene ricevuto il segnale di vuoto creato mediante un pressostato (un contatto che si chiude, chiamato “pressostato_vuoto”), riparte la fase di salita dell'isola fino a

raggiungere un finecorsa di isola tutta ribaltata (chiamato "FC_salita2"). Resta in quella posizione "x" secondi per garantire la fuoriuscita del prodotto della cassetta e successivamente viene riattivato il motore (che gira ovviamente in senso contrario) per riportare l'isola in posizione iniziale.

Durante la fase di discesa viene nuovamente raggiunto il finecorsa intermedio "FC_salita1"; quando tale ingresso diventa alto si effettuano due operazioni:

- fermare il motore*
- annullare il vuoto*

A questo punto annullando il vuoto si apre il contatto del pressostato ed il motore viene riattivato fino a quando non verrà raggiunto il finecorsa di isola in posizione iniziale (chiamato "FC_discesa").

Quindi quando l'ingresso relativo al finecorsa "FC_discesa" diventa alto:

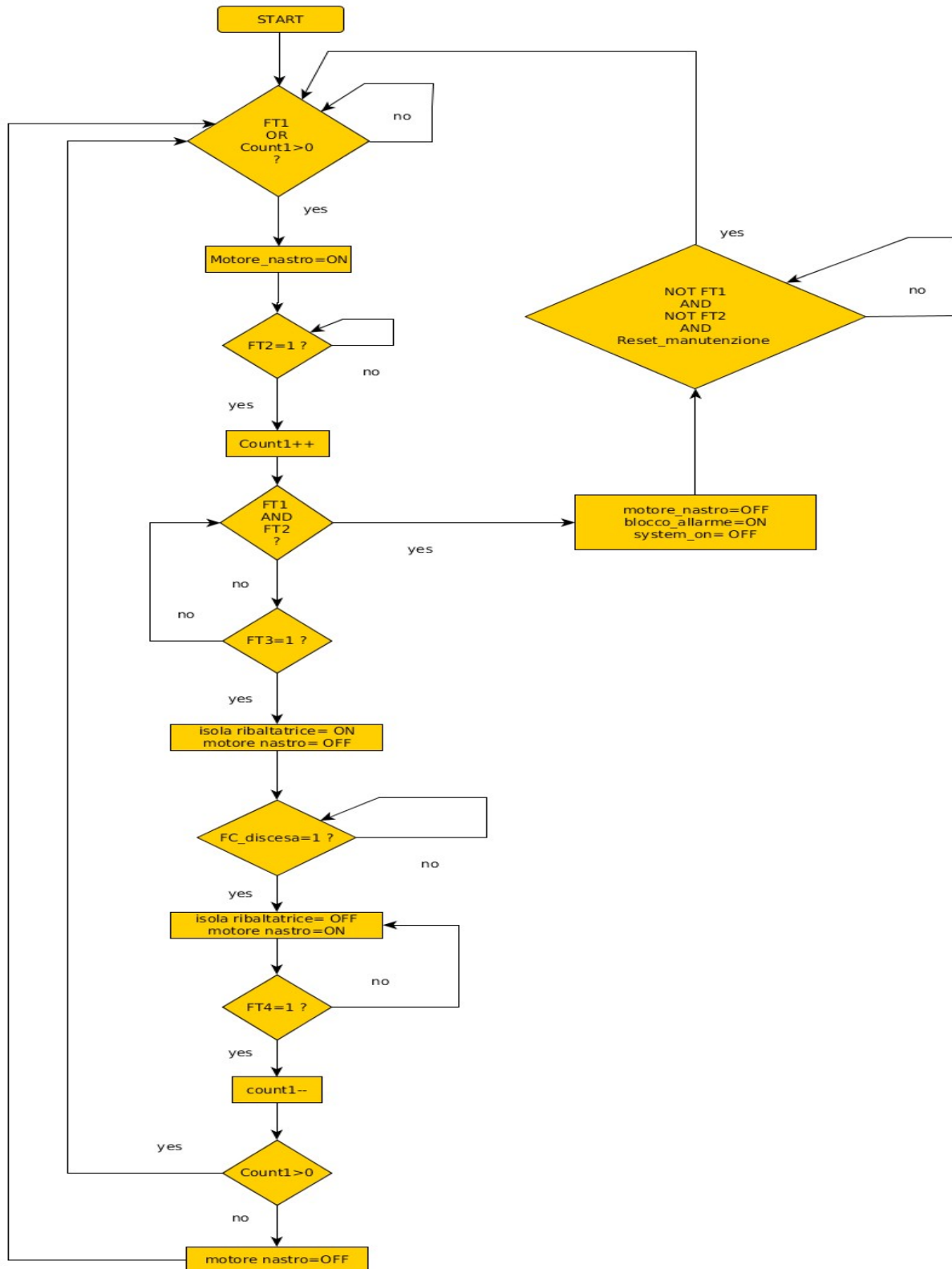
- viene fermato il motore dell'isola ribaltatrice*
- viene riattivato il motore del nastro trasportatore per espellere la cassetta*

Se dopo l'espulsione non è presente nessuna cassetta sul nastro, il motore viene fermato fino a quando non viene rilevata una nuova cassetta da FT1.

Diagramma di flusso e diagramma a stati

E' molto utile tracciare un diagramma di flusso per avere bene in mente tutte le operazioni che devono essere effettuate e in generale per esplicitare meglio la logica di controllo. In questo caso visto che il nastro trasportatore viene fermato quando l'isola ribaltatrice è in funzione, è possibile avere due diagrammi di flusso separati, ovvero uno per il nastro e uno per l'isola ribaltatrice.

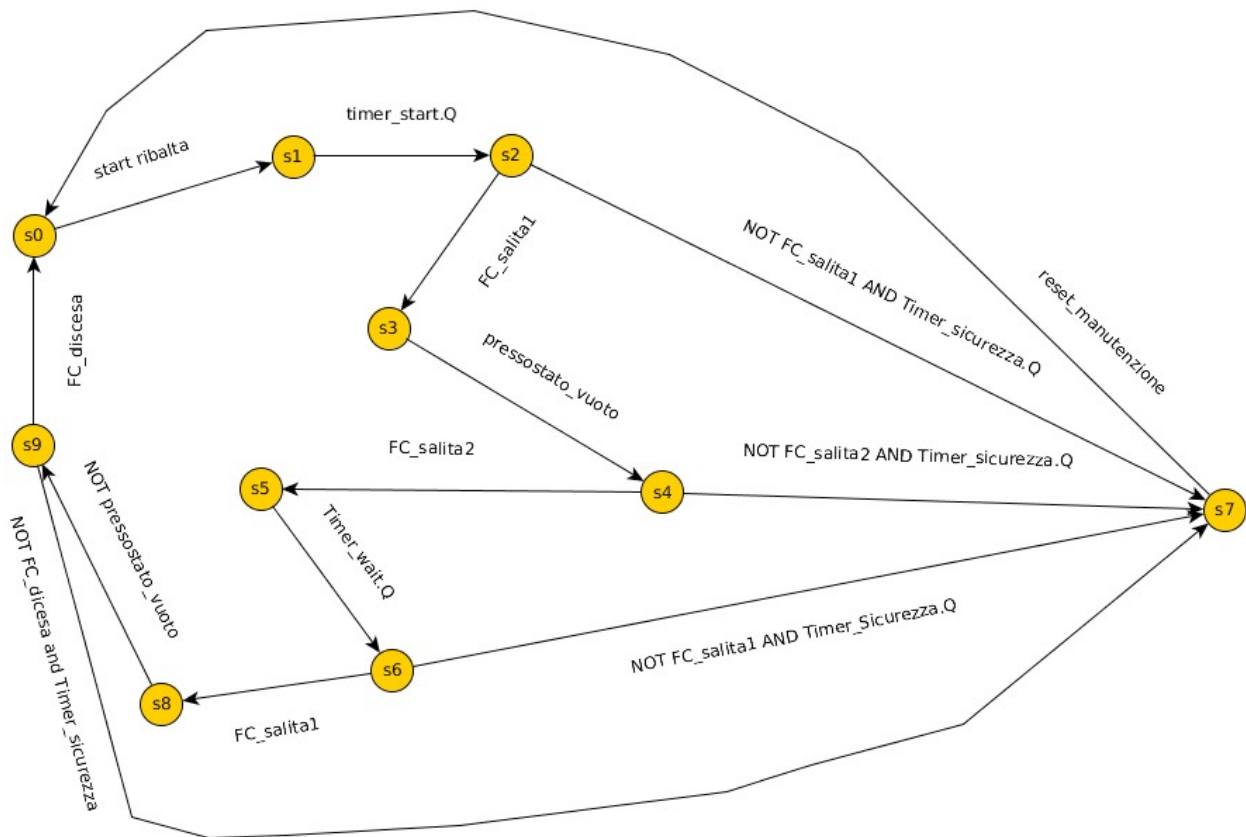
Quindi per quanto riguarda il nastro trasportatore il diagramma di flusso sarà il seguente:



Dove Count1 è il counter che indica il numero di cassette presenti sul nastro. Invece con “Isola ribaltatrice=ON” si intende lo start di tutta la logica di controllo dell'isola ribaltatrice che, come vedremo più avanti, corrisponde al richiamare un'istanza della FB relativa all'isola ribaltatrice.

Adesso anziché tracciare un altro diagramma di flusso per il controllo dell'isola ribaltatrice, preferisco invece utilizzare il diagramma a stati.

In particolare tale sistema può essere rappresentato con 10 stati:



Descrizione del diagramma:

s0: stato iniziale, il sistema va nello stato s0 quando;

- trovandosi nello stato s9 viene chiuso il contatto di finecorsa FC_discesa
- trovandosi nello stato s7 viene premuto il pulsante reset_manutenzione

Azioni Eseguite: motore_nastro=off, motore_ribalta_salita/discesa=off

s1: il sistema va nello stato s1 quando si trova in s0 e viene avviata l'isola ribaltatrice (dalla fotocellula FT3, vedi diagramma di flusso); Azioni Eseguite: avvio timer_start (per inserire un tempo di attesa prima di avviare il motore in salita).

s2: il sistema va nello stato s2 quando si trova in s1 e viene abilitata l'uscita di timer_start. Azioni Eseguite: motore_ribalta_salita=on, avvio timer_sicurezza

s3: il sistema va nello stato s3 quando si trova in s2 e viene chiuso il contatto del primo finecorsa in salita (FC_salita1). Azioni Eseguite: motore_ribalta_salita=off, valvola_vuoto=on

s4: il sistema va nello stato s4 quando si trova in s3 e il contatto (normalmente aperto) del pressostato si chiude. Azioni Eseguite: motore_ribalta_salita=on, avvio timer_sicurezza

s5: il sistema va nello stato s5 quando si trova in s4 e viene chiuso il secondo contatto di finecorsa (FC_salita2). Azioni Eseguite: motore_ribalta_salita=off, start Timer_wait (introduce un tempo di attesa prima di invertire il senso del motore)

s6: il sistema va nello stato s6 quando si trova in s5 e viene abilitata l'uscita del timer Timer_wait.

Azioni Eseguite: motore_ribalta_discesa=on, avvio timer_sicurezza

s8: il sistema va nello stato s8 quando si trova in s6 e viene chiuso il contatto di finecorsa FC_salita1. Azioni Eseguite: motore_ribalta_discesa=off, valvola_vuoto=off

s9: il sistema va nello stato s9 quando si trova in s8 e il contatto del pressostato diventa nuovamente aperto. Azioni Eseguite: motore_ribalta_discesa=on, avvio timer_sicurezza

s7: Se il sistema va in questo stato significa che c'è stato un guasto. Infatti è stato introdotto un timer chiamato timer_sicurezza attraverso il quale viene impostato un tempo massimo affinché vengano raggiunti i finecorsa dopo che il motore è stato avviato (sia in salita che in discesa).

In particolare il sistema va nello stato s7 quando:

- si trova nello stato s2 e l'uscita del timer_sicurezza diventa alta
- si trova nello stato s4 e l'uscita del timer_sicurezza diventa alta
- si trova nello stato s6 e l'uscita del timer_sicurezza diventa alta
- si trova nello stato s9 e l'uscita del timer_sicurezza diventa alta

Programmazione PLC

Nelle pagine precedenti ho deciso di descrivere separatamente la logica di controllo del nastro e dell'isola ribaltatrice per due motivi:

1. Diminuire la complessità di tutta la logica di controllo
2. Avere la possibilità di ampliare l'impianto ad esempio utilizzando più isole ribaltatrici per un singolo nastro trasportatore (nel qual caso occorrerà apportare delle opportune modifiche).

Ho scelto pertanto di realizzare due Function Block (FB), una per il nastro trasportatore e una per l'isola ribaltatrice. In entrambi i casi per semplicità realizzerò le FB in LAD.

Per il nastro trasportatore utilizzerò il diagramma di flusso tracciato precedentemente, mentre per l'isola ribaltatrice utilizzerò la tecnica del diagramma a stati.

E' importante notare quindi che la scelta di utilizzare delle Function Block non è stata casuale, infatti in questo modo posso disporre di più istanze di tali FB avendo di conseguenza la possibilità di controllare eventualmente più nastri trasportatori e /o isole ribaltatrici.

Di seguito sono raffigurate le due Function Block, la prima è quella relativa all'isola ribaltatrice mentre la seconda è quella del nastro trasportatore:

nastro_trasportatore_isola_rib_PLCTForum / PLC_1 [CPU 1518-4 PN/DP] / Program blocks

isola_ribaltatrice_diag_stati [FB3]

isola_ribaltatrice_diag_stati Properties

General

Name	isola_ribaltatrice_diag_stati	Number	3	Type	FB
Language	LAD	Numbering	automatic		

Information

Title	Isola Ribaltatrice (sviluppata in Ladder con la tecnica del diagramma a stati).	Author		Comment	<p>Anche in questo caso creo una Function Block FB e non una Function FC, questo per prevedere una eventuale espansione dell'impianto in cui potrebbero essere presenti più isole ribaltatrici per ogni singolo nastro trasportatore. In questo modo posso quindi controllare più isole ribaltatrici richiamando un'istanza di questa FB (FB3) all'interno della Function Block del nastro trasportatore.</p> <p>Questa Function Block è stata sviluppata in logica Ladder utilizzando la tecnica del diagramma a stati. In questo caso abbiamo 8 stati: s0, s1, s2, s3, s4, s5, s6, s7, s8.</p>
Family		Version	0.1	User-defined ID	

isola_ribaltatrice_diag_stati

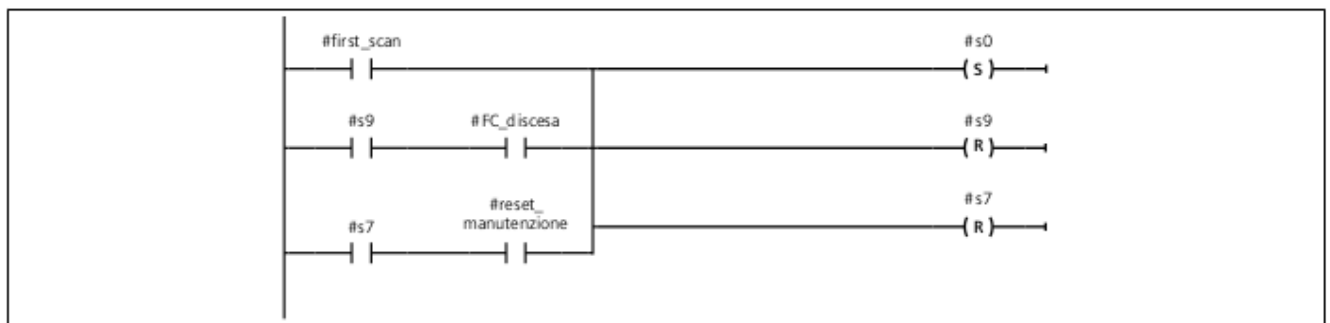
Name	Data type	Default value	Retain
▼ Input			
first_scan	Bool	false	Non-retain
start_ribalta	Bool	false	Non-retain
FC_salita1	Bool	false	Non-retain
FC_salita2	Bool	false	Non-retain
FC_discesa	Bool	false	Non-retain
pressostato_vuoto	Bool	false	Non-retain
reset_manutenzione	Bool	false	Non-retain
▼ Output			
motore_ribalta_salita	Bool	false	Non-retain
motore_ribalta_discesa	Bool	false	Non-retain
valvola_vuoto	Bool	false	Non-retain
lampada_manutenzione	Bool	false	Non-retain
InOut			
▼ Static			
s0	Bool	false	Non-retain

Name	Data type	Default value	Retain
s1	Bool	false	Non-retain
s2	Bool	false	Non-retain
s3	Bool	false	Non-retain
s4	Bool	false	Non-retain
s5	Bool	false	Non-retain
s6	Bool	false	Non-retain
s7	Bool	false	Non-retain
s8	Bool	false	Non-retain
s9	Bool	false	Non-retain
▼ Timer_inizio	IEC_TIMER		Non-retain
ST	Time	T#0ms	Non-retain
PT	Time	T#0ms	Non-retain
ET	Time	T#0ms	Non-retain
RU	Bool	false	Non-retain
IN	Bool	false	Non-retain
Q	Bool	false	Non-retain
▼ Timer_sicurezza	IEC_TIMER		Non-retain
ST	Time	T#0ms	Non-retain
PT	Time	T#0ms	Non-retain
ET	Time	T#0ms	Non-retain
RU	Bool	false	Non-retain
IN	Bool	false	Non-retain
Q	Bool	false	Non-retain
▼ Timer_wait	IEC_TIMER		Non-retain
ST	Time	T#0ms	Non-retain
PT	Time	T#0ms	Non-retain
ET	Time	T#0ms	Non-retain
RU	Bool	false	Non-retain
IN	Bool	false	Non-retain
Q	Bool	false	Non-retain
Temp			
Constant			

Network 1: Diagramma a stati: Stato s0

Il sistema passa allo stato s0 quando:

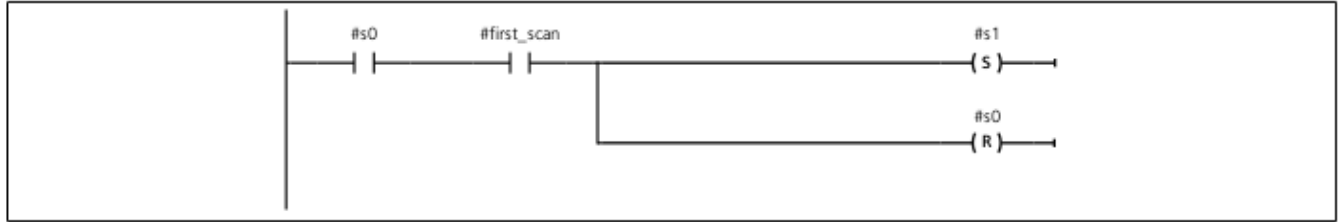
- subito dopo il primo scan (vedi richiamo di FB3 in FB1)
- quando trovandosi nello stato s6 viene chiuso il contatto di finecorsa FC_discesa
- quando trovandosi nello stato s7 viene premuto il pulsante reset_manutenzione



Network 2: Diagramma a stati: Stato s1

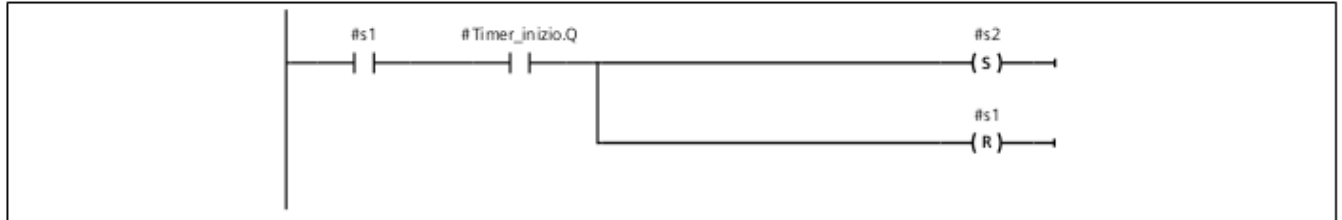
Il sistema passa allo stato s1 quando si trova sullo stato s0 subito dopo il primo scan (a prima vista può sembrare uno stato ridondante, ma non è così, per capirlo bisogna guardare il diagramma a stati).

--	--	--



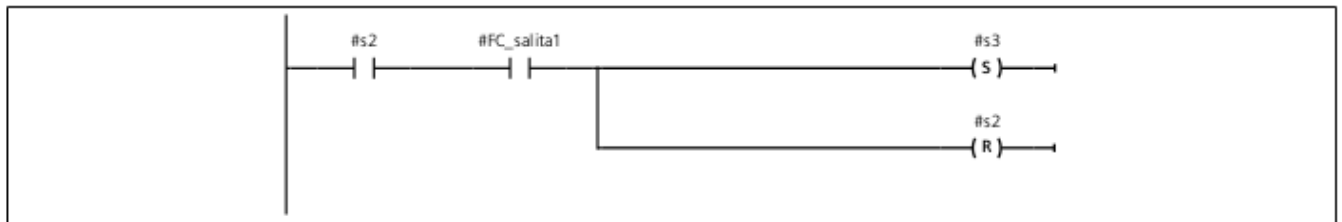
Network 3: Diagramma a stati: Stato s2

Il sistema passa allo stato s2 quando si trova sullo stato s1 e l'uscita del timer "Timer_inizio" diventa alta



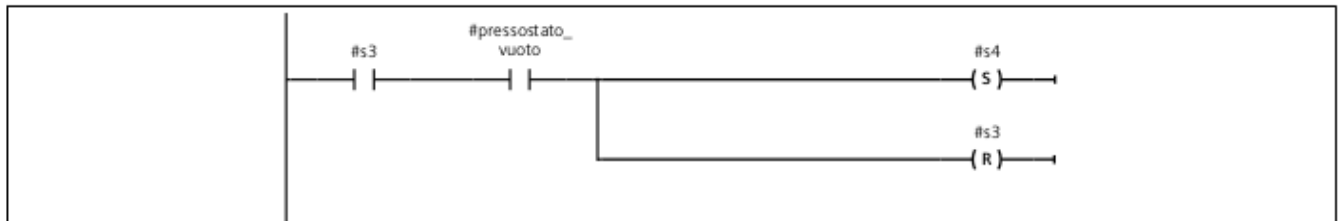
Network 4: Diagramma a stati: Stato s3

Il sistema passa allo stato s3 quando si trova sullo stato s2 e il primo contatto di fine corsa in fase di salita FC_salita1 viene chiuso



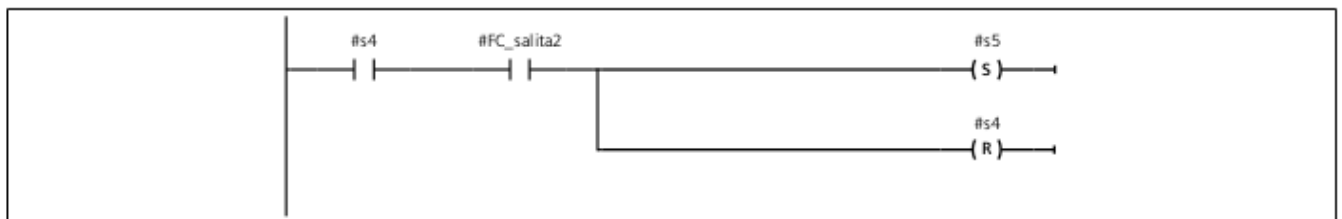
Network 5: Diagramma a stati: Stato s4

Il sistema passa allo stato s4 quando si trova sullo stato s3 e il contatto del pressostato (normalmente aperto) viene chiuso.



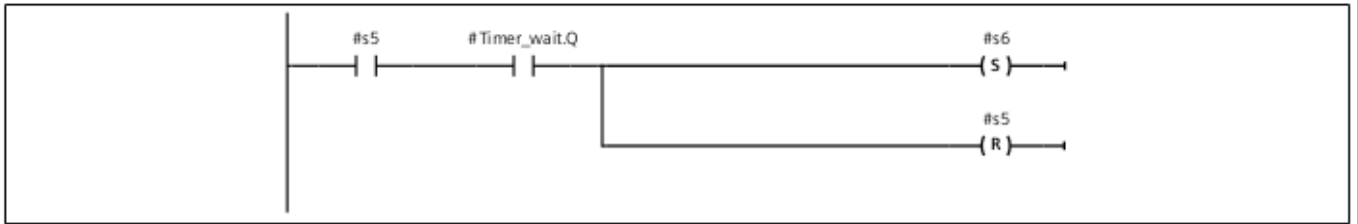
Network 6: Diagramma a stati: Stato s5

Il sistema passa allo stato s5 quando si trova sullo stato s4 e il secondo contatto di fine corsa in fase di salita FC_salita2 viene chiuso



Network 7: Diagramma a stati: Stato s6

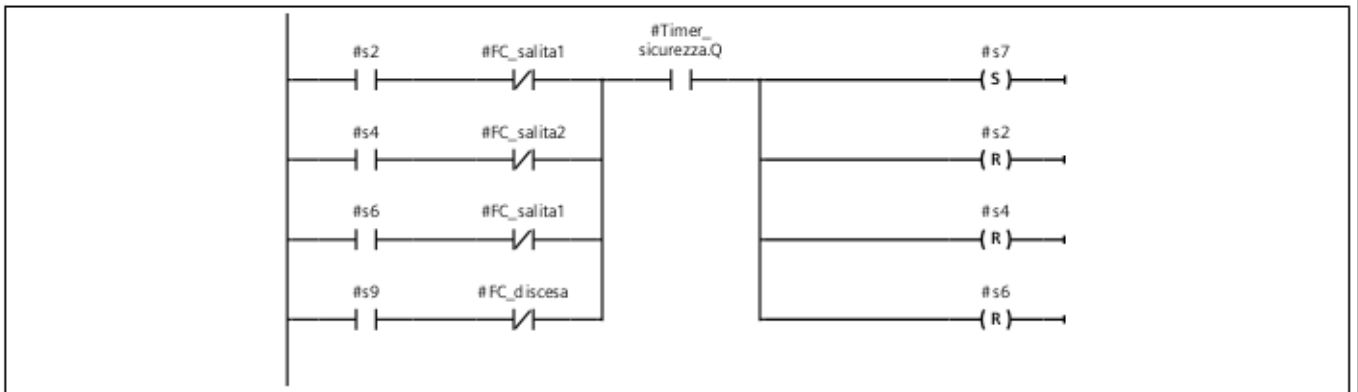
Il sistema passa allo stato s6 quando si trova sullo stato s5 e l'uscita del timer "Timer2" diventa alta (con il Timer_wait imposto il tempo di ritardo di attivazione del motore in fase di discesa dopo che è stato raggiunto il finecorsa FC_salita2 in fase di salita, vedi "Network 14").



Network 8: Diagramma a stati: Stato s7

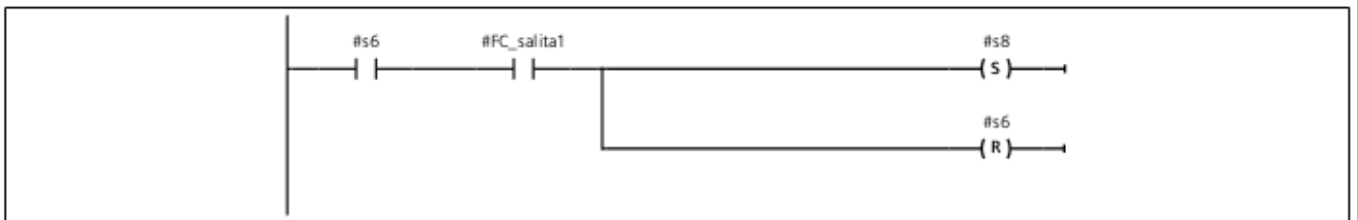
Il sistema passa allo stato s7 quando:

- quando trovandosi nello stato s2 con il finecorsa FC_salita1 aperto viene attivato il timer "Timer_sicurezza"
- quando trovandosi nello stato s4 con il finecorsa FC_salita2 aperto viene attivato il timer "Timer_sicurezza"
- quando trovandosi nello stato s6 con il finecorsa FC_salita1 aperto viene attivato il timer "Timer_sicurezza"
- quando trovandosi nello stato s9 con il finecorsa discesa aperto viene attivato il timer "Timer_sicurezza"



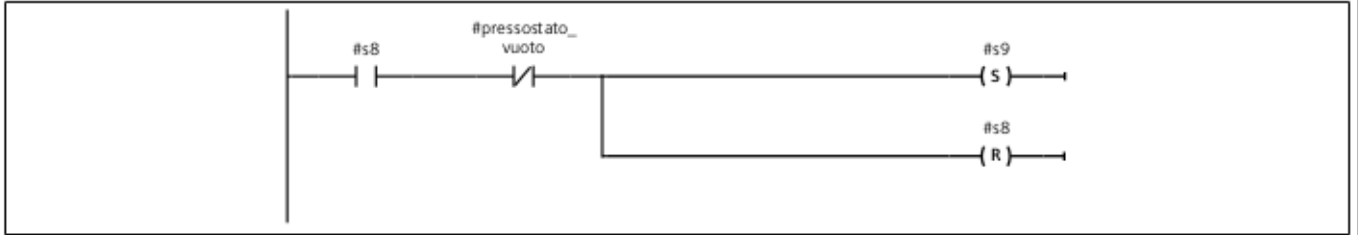
Network 9: Diagramma a stati: Stato s8

Il sistema passa allo stato s8 quando si trova sullo stato s6 e il contatto di fine corsa intremedio FC_salita1 viene chiuso



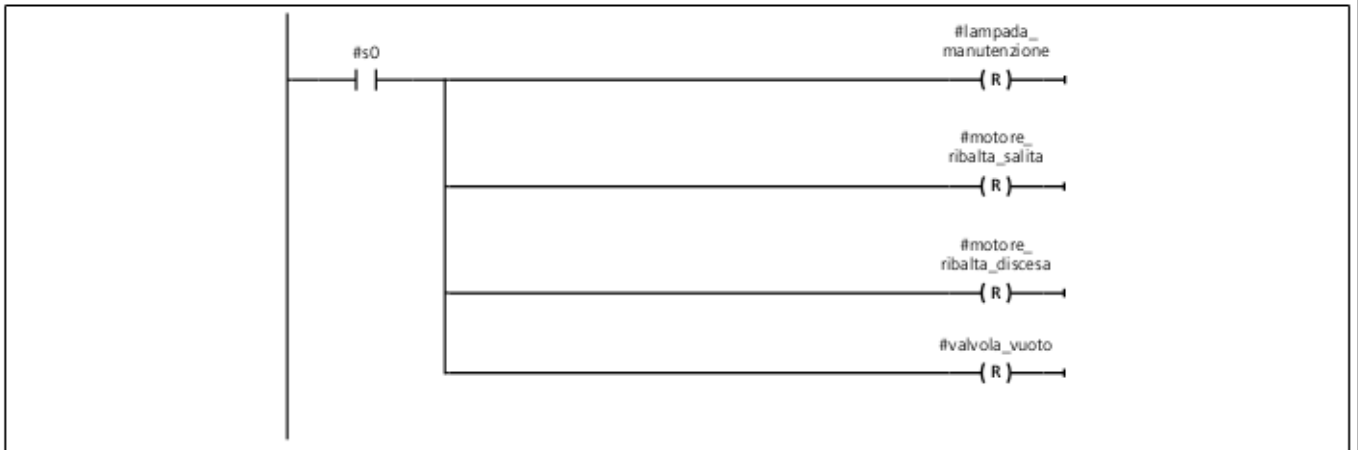
Network 10: Diagramma a stati: Stato s9

Il sistema passa allo stato s9 quando si trova sullo stato s8 e il contatto del pressostato viene aperto



Network 11: Operazioni effettuate nello stato s0 (Ridondante)

Nello stato s0 resetto i seguenti merker (come si può vedere lo stato s0 è lo stato iniziale del sistema). Nota che le operazioni effettuate in questo stato sono ridontanti, ovvero l'intero network 9 è ridondante e potrebbe essere cancellato, ma è bene inserirlo per avere una descrizione di questo stato.



Network 12: Descrizione del Timer "#Timer_inizio"

Nello stato s1 attivo il timer "Timer_inizio", in cui imposto il tempo di ritardo per il passaggio allo stato s2 (vedi "Network 3") ossia il ritardo di avvio del motore in fase di salita dopo che la cassetta viene rilevata da FT3

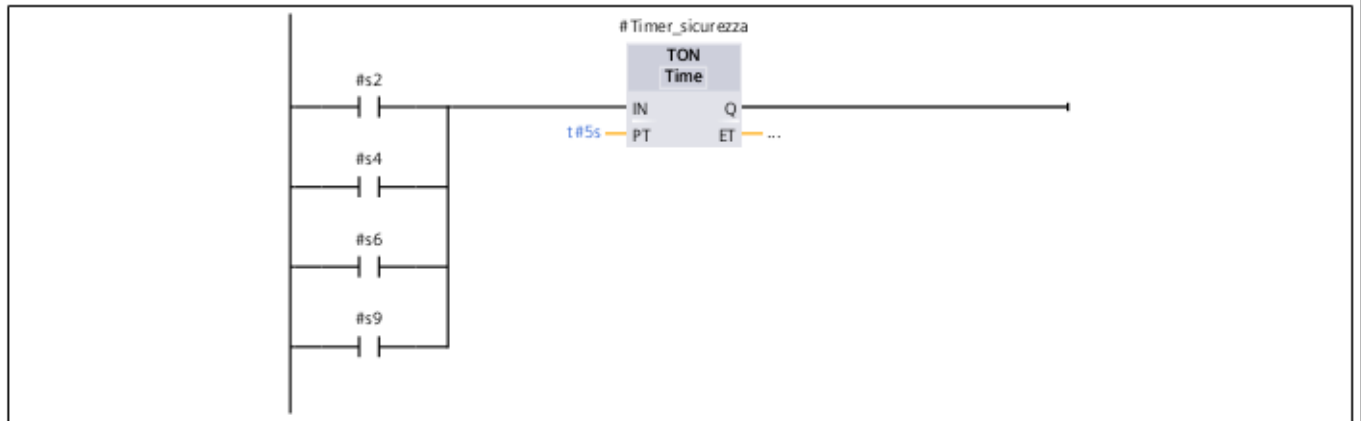


Network 13: Descrizione del Timer "#Timer_sicurezza"

Questo timer viene utilizzato per spegnere il motore (sia in fase di salita che di discesa) se i relativi finecorsa non vengono attivati dopo un determinato intervallo di tempo (in questo caso 5 secondi).

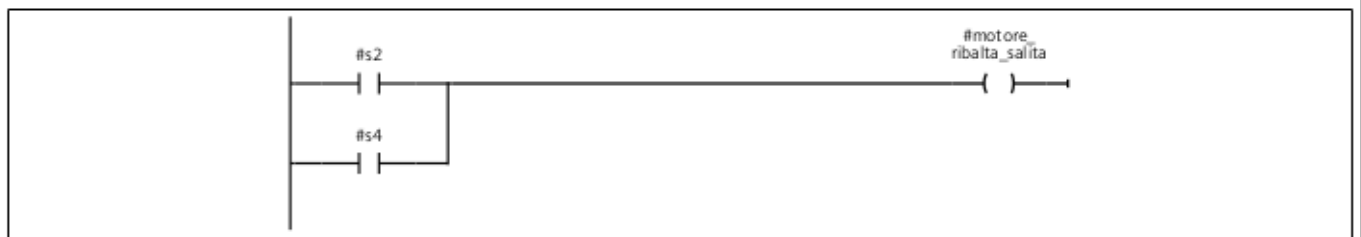
Quindi viene attivato quando ci si trova negli stati:

- s2 (fase di salita in cui non è ancora stato raggiunto il finecorsa FC_salita1)
- s4 (fase di salita in cui non è ancora stato raggiunto il finecorsa FC_salita2)
- s6 (fase di discesa in cui non è ancora stato raggiunto il finecorsa FC_discesa)



Network 14: Attivazione del motore in fase di salita

Il motore in fase di salita viene attivato negli stati s2 ed s3



Network 15: Attivazione Valvola vuoto

La valvola del vuoto deve essere attivata nello stato s3 (ovvero subito dopo che il contatto di finecorsa FC_salita1 viene chiuso) e rimanere attiva negli stati s4, s5, s6.



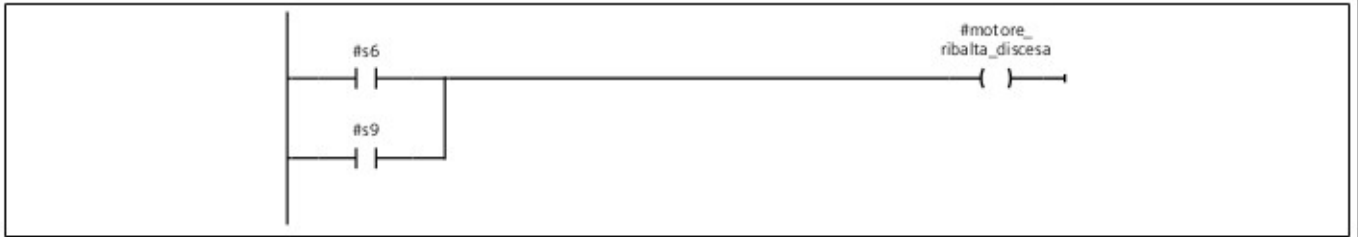
Network 16: Descrizione del Timer "#Timer_wait"

Attraverso questo timer imposto il tempo di ritardo di attivazione del motore in fase di discesa dopo che è stato chiuso il finecorsa FC_salita2



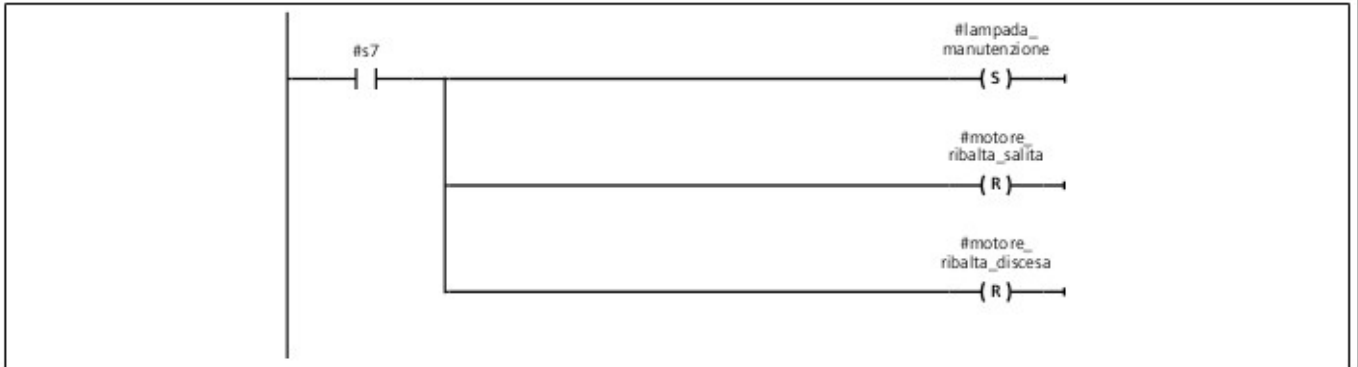
Network 17: Attivazione del motore in fase di discesa

Il motore in fase di discesa è attivo solamente nello stato s6



Network 18: Operazioni nello stato s7 (Ridondante)

Queste operazioni sono ridondanti (o meglio il "Network 16" è ridondante e si potrebbe anche cancellare) ma vengono effettuate anche per esplicitare ciò che è importante avvenga in questo stato.



nastro_trasportatore_isola_rib_PLCForum / PLC_1 [CPU 1518-4 PN/DP] / Program blocks

nastro_trasportatore [FB1]

nastro_trasportatore Properties

General

Name	nastro_trasportatore	Number	1	Type	FB
Language	LAD	Numbering	automatic		

Information

Title	Nastro Trasportatore	Author		Comment	Creo una Function Block FB e non una Function FC per il nastro trasportatore, per prevedere una espansione dell'impianto in cui potrebbero essere presenti più nastri trasportatori. In questo modo posso quindi controllare più nastri semplicemente richiamando un istanza della FB. Per lo stesso motivo decido di non integrare le operazioni relative al ribaltamento della cassetta in questa Function Block, ma appunto creo un'altra FB relativa all'isola ribaltatrice e richiamo un istanza di essa all'interno di FB1.
Family		Version	0.1	User-defined ID	

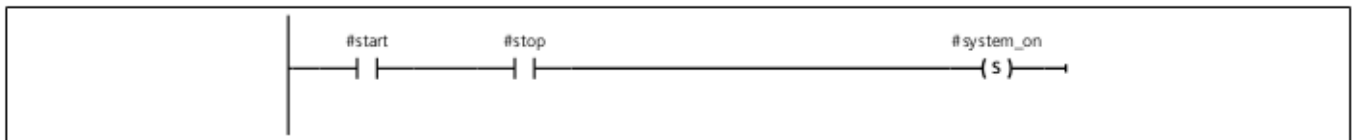
nastro_trasportatore

Name	Data type	Default value	Retain
▼ Input			
start	Bool	false	Non-retain
stop	Bool	false	Non-retain
FT1	Bool	false	Non-retain
FT2	Bool	false	Non-retain
FT3	Bool	false	Non-retain
FT4	Bool	false	Non-retain
FC_salita1	Bool	false	Non-retain
FC_salita2	Bool	false	Non-retain
FC_discesa	Bool	false	Non-retain
pressostato_vuoto	Bool	false	Non-retain
reset_manutenzione	Bool	false	Non-retain
▼ Output			
motore_nasto	Bool	false	Non-retain
motore_ribalta_salita	Bool	false	Non-retain
motore_ribalta_discesa	Bool	false	Non-retain
blocco_per_allarme	Bool	false	Non-retain
valvola_vuoto	Bool	false	Non-retain
system_on	Bool	false	Non-retain
posizione_cassetta_hmi	Int	0	Non-retain

Name	Data type	Default value	Retain
InOut			
▼ Static			
▼ counter1	IEC_COUNTER		Retain
CU	Bool	false	Retain
CD	Bool	false	Retain
R	Bool	false	Retain
LD	Bool	false	Retain
QU	Bool	false	Retain
QD	Bool	false	Retain
PV	Int	0	Retain
CV	Int	0	Retain
isola_ribalta	Bool	false	Non-retain
bit_mem1	Bool	false	Non-retain
posizione_1_hmi	Bool	false	Non-retain
posizione_2_hmi	Bool	false	Non-retain
posizione_3_hmi	Bool	false	Non-retain
Temp			
Constant			

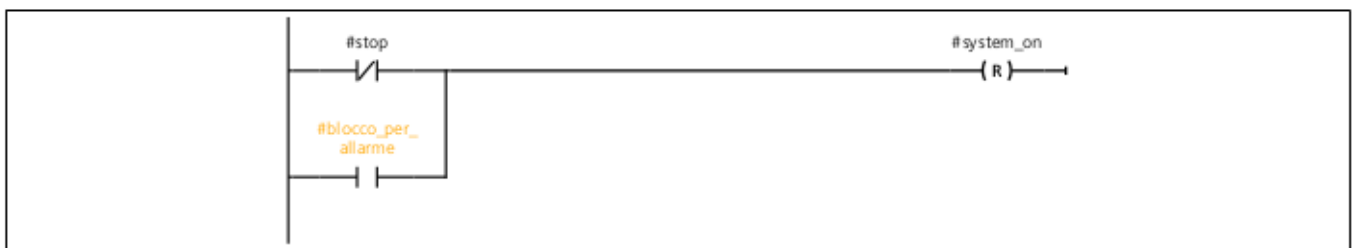
Network 1: Attivo l'impianto settando #system_on

Uso un merker "system_on" che mi indica quando il sistema è in funzione. (da notare che il pulsante stop è normalmente chiuso, pertanto nella logica ladder sarà normalmente aperto).



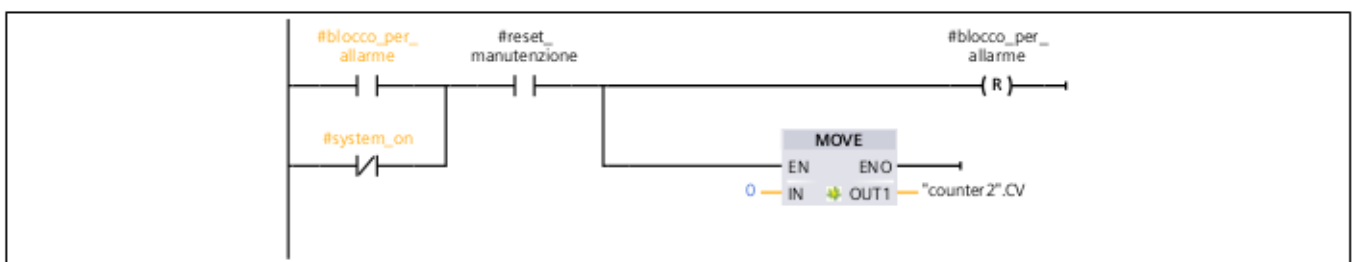
Network 2: Disattivo l'impianto resettando #system_on

Disattivo tutto il sistema quando premo il pulsante di stop e quando l'uscita "blocco_per_allarme" è attiva (tale uscita viene attivata ad esempio quando i pulsanti di fine corsa non vengono azionati entro un certo intervallo di tempo impostabile a piacere).



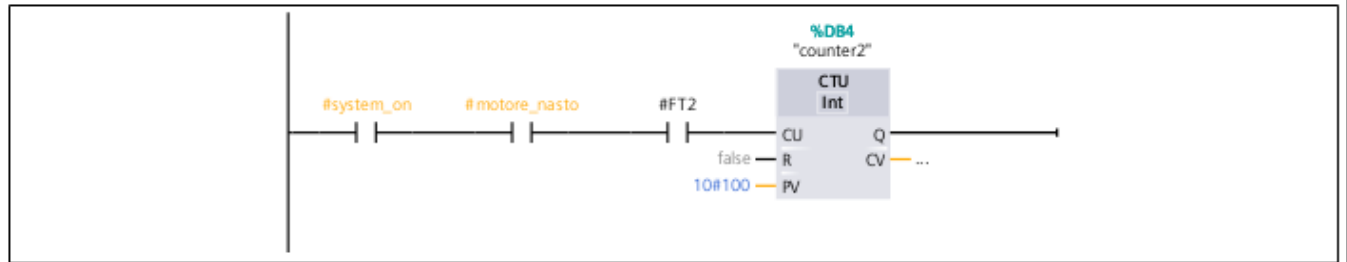
Network 3: Funzione del pulsante "Reset_manutenzione"

Quando premo il pulsante "reset_manutenzione" (pulsante che viene premuto dopo che è stata eseguita una manutenzione a seguito di un blocco dell'impianto, vedi ad esempio il "Network 10"), mi assicuro di resettare #system_on (in teoria non sarebbe necessario) ed ovviamente resetto il merker #blocco_per_allarme



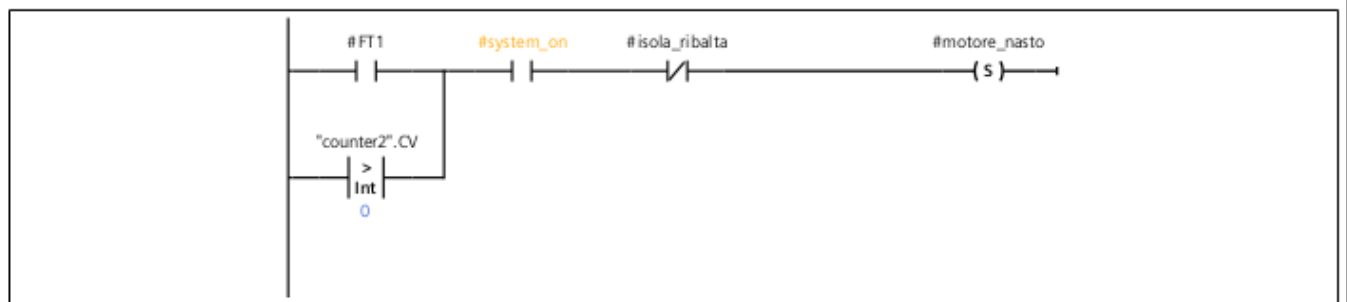
Network 4: Incrementa il Counter

Ogni volta che la fotocellula FT1 rileva una cassetta, il counter viene incrementato di 1



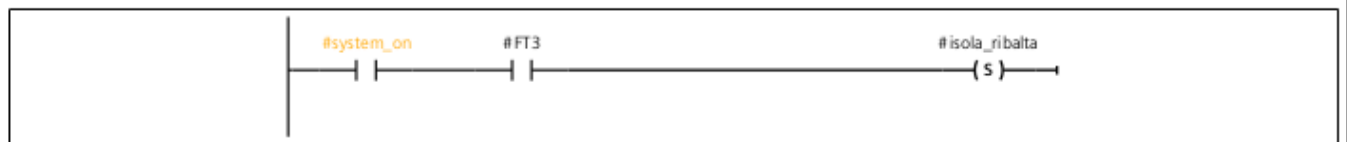
Network 5: Attiva il motore del nastro trasportatore

Il motore che fa girare il nastro viene attivato qualora ci sia almeno una cassetta nel nastro e l'isola ribaltatrice non sia in funzione



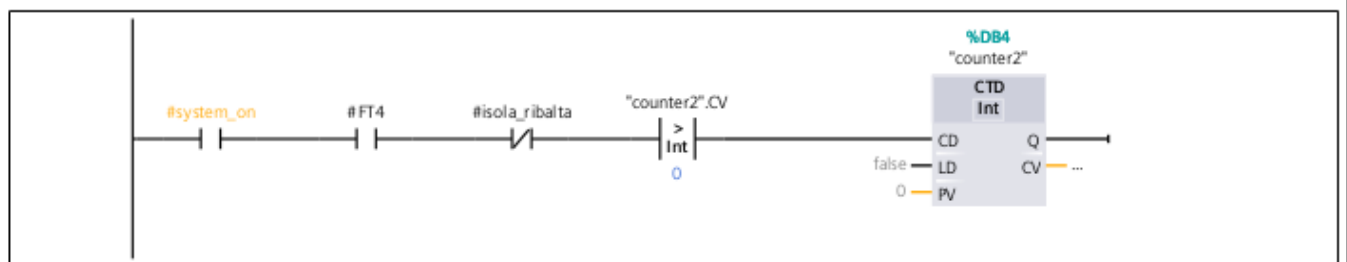
Network 6: Attivazione isola ribaltatrice

Quando una cassetta arriva alla fotocellula FT3 attivo il merker "isola_ribalta" (Se #isola_ribalta=1 il motore del nastro viene spento, vedi "Network 11").



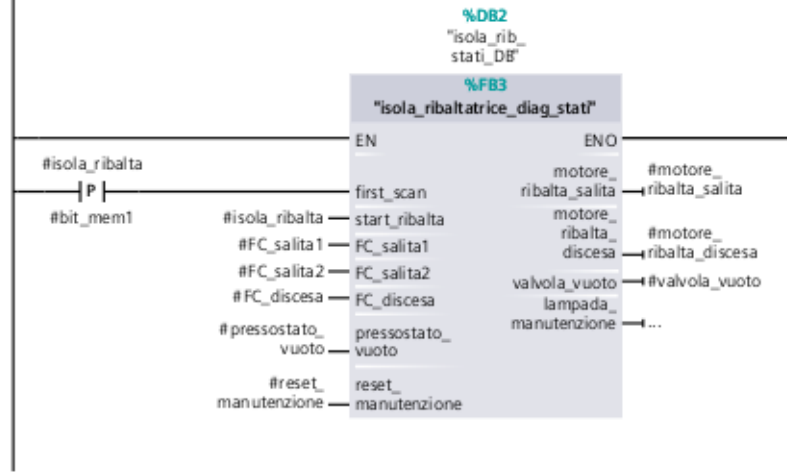
Network 7: Decrementa il counter

Quando la cassetta viene espulsa, viene rilevata dalla fotocellula FT4 e pertanto viene decrementato il contatore.



Network 8: Richiamo della FB "isola_ribaltatrice_diag_stati" (FB3)

Richiamo un istanza della FB "isola_ribaltatrice_diag_stati" (come vedremo in seguito si tratta di una FB realizzata in logica Ladder con la tecnica del digramma a stati). Da notare che il first_scan corrisponde al primo scan in cui il merker "#isola_ribalta" passa da 0 a 1.



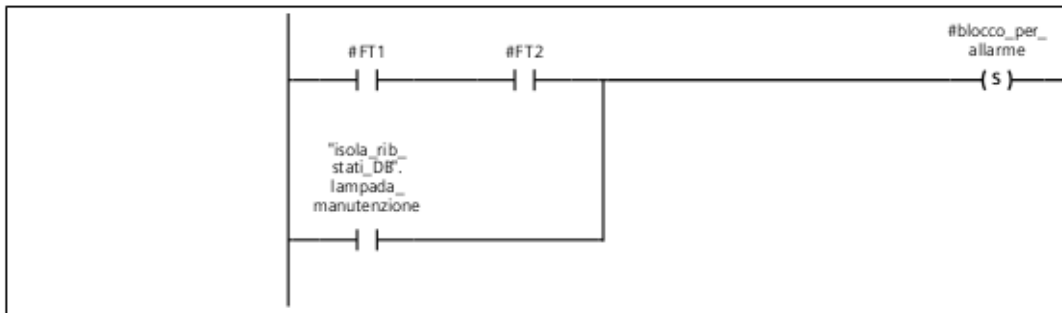
Network 9: Resetta il merker #isola_ribalta

Quando mi trovo nel primo stato, ovvero s1 allora resetto il merker "isola_ribalta" in quanto in quello stato tutte le operazioni per il ribaltamento della cassetta sono state effettuate.



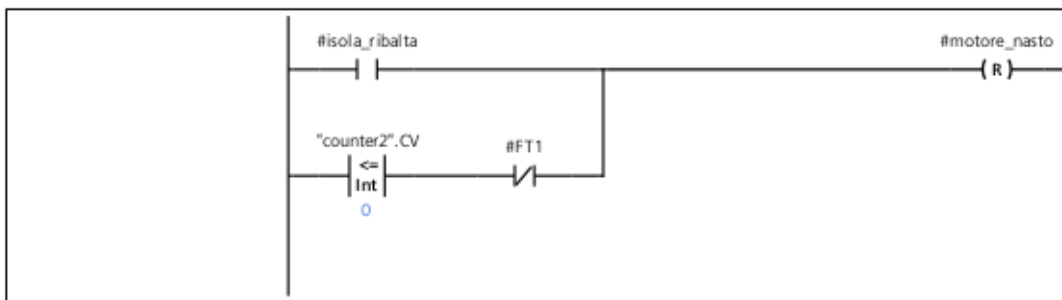
Network 10: Attivazione del merker "blocco_per_allarme"

Attivo il merker "blocco_per_allarme" (che mi indica che è necessario effettuare una manutenzione e che inoltre blocca tutto l'impianto, vedi "Network 2" e "Network 12") se le fotocellule FT1 ed FT2 sono attive contemporaneamente e se l'uscita "lampada_manutenzione" di FB3 è alta.



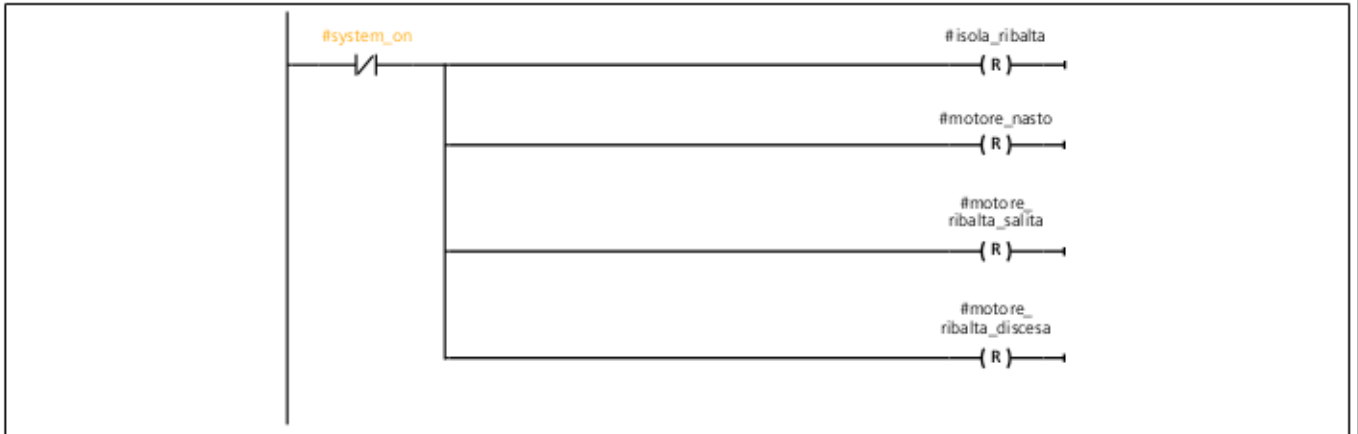
Network 11: Spegnimento motore nastro trasportatore

Il nastro deve essere fermo durante tutte le operazioni di ribaltamento della cassetta, ed inoltre deve essere ovviamente fermo se non sono presenti cassette.



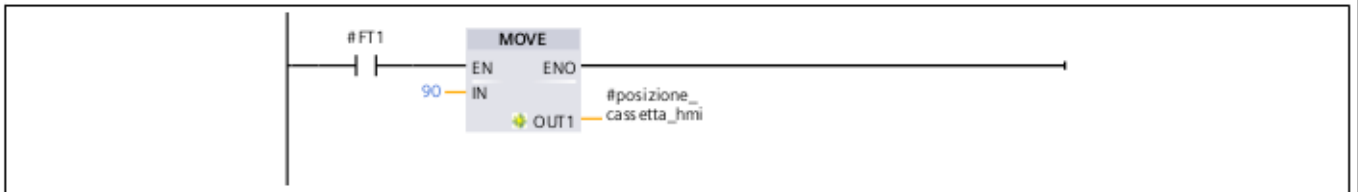
Network 12: Spegnimento motori e reset merker

Quando il sistema non è in funzione mi assicuro di resettare i seguenti merker e uscite



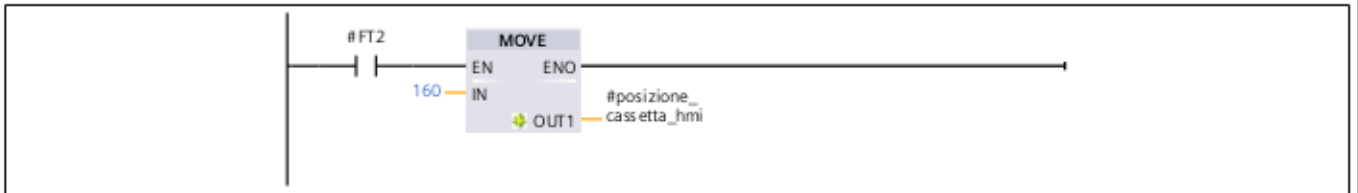
Network 13: HMI - posizione cassetta nel pannello operatore

Per la rappresentazione sul pannello operatore, quando la cassetta passa per FT1 pongo posizione_cassetta_hmi=90



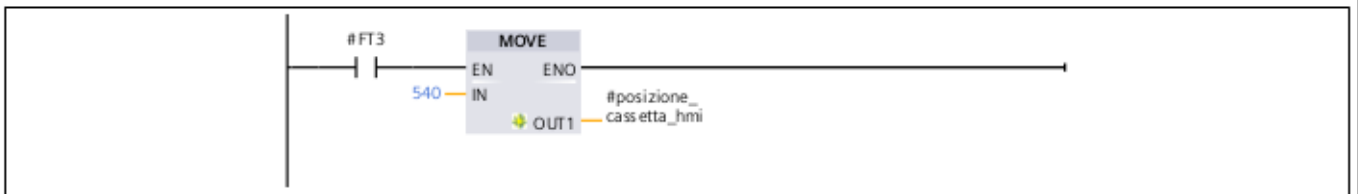
Network 14: HMI - posizione cassetta nel pannello operatore

Per la rappresentazione sul pannello operatore, quando la cassetta passa per FT2 pongo posizione_cassetta_hmi=160



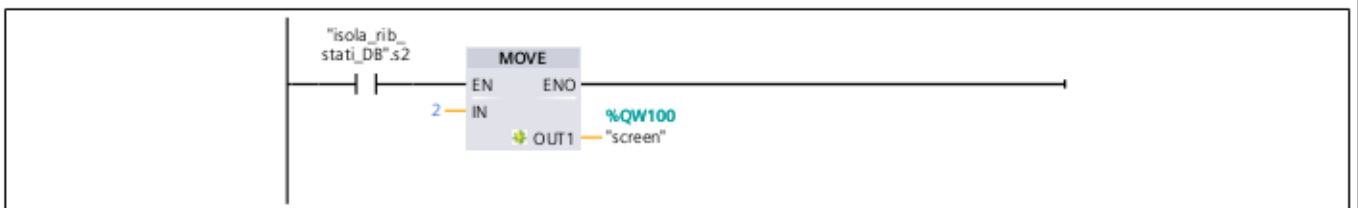
Network 15: HMI - posizione cassetta nel pannello operatore

Per la rappresentazione sul pannello operatore, quando la cassetta passa per FT3 pongo posizione_cassetta_hmi=540



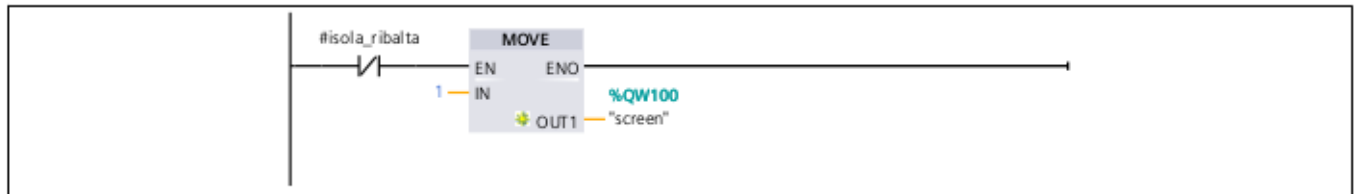
Network 16: HMI- Cambia screen quando l'isola ribaltatrice è in funzione

Appena il motore dell'isola ribaltatrice viene avviato (stato s2) cambio la schemata del pannello operatore



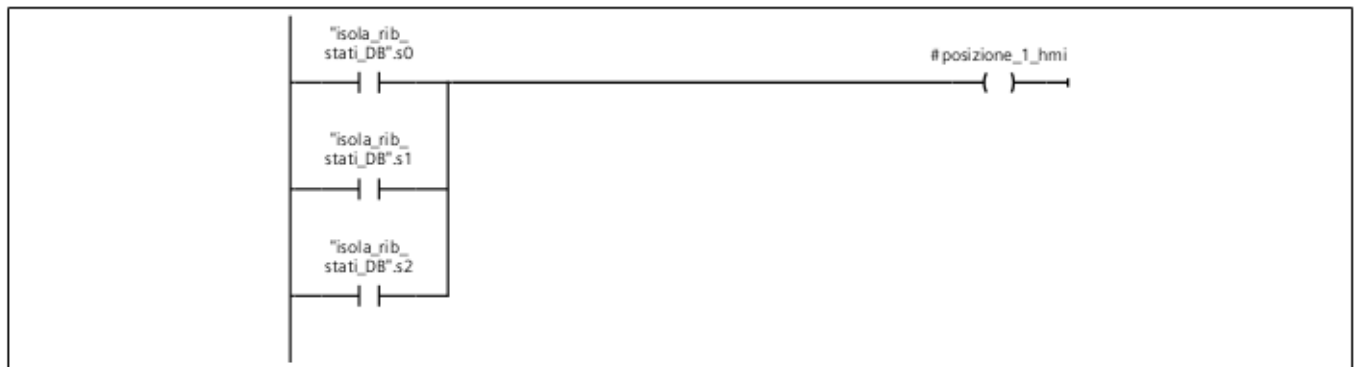
Network 17: HMI- Cambia screen quando l'isola ribaltatrice non è in funzione

Quando l'isola ribaltatrice non è in funzione viene visualizzata lo screen numero 1 nel pannello operatore (ossia quello del nastro trasportatore)



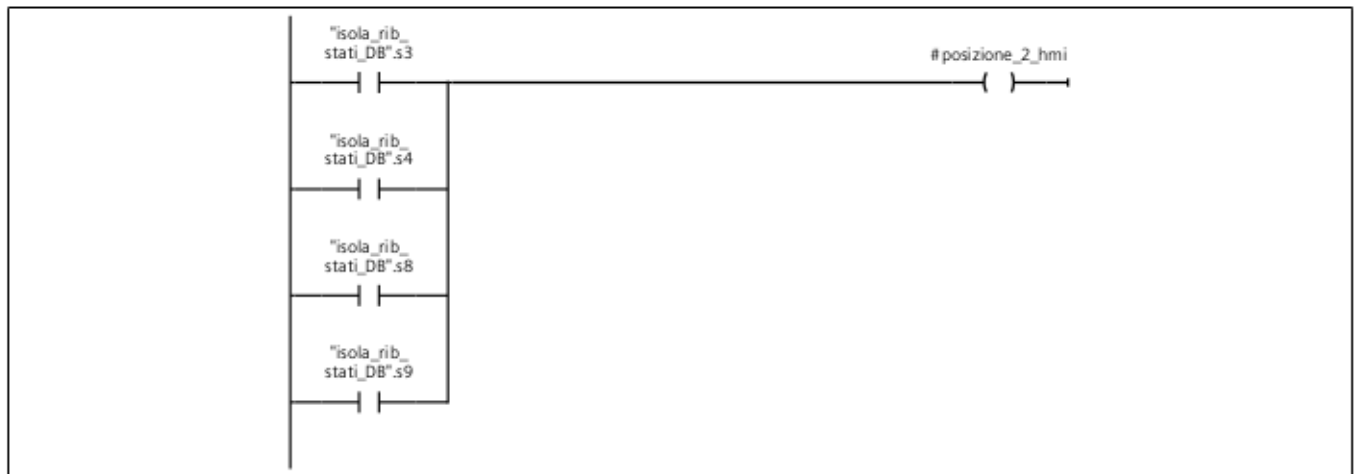
Network 18: HMI - Posizione iniziale cassetta isola ribaltatrice

Serve per indicare la posizione iniziale della cassetta nella seconda schermata del pannello operatore



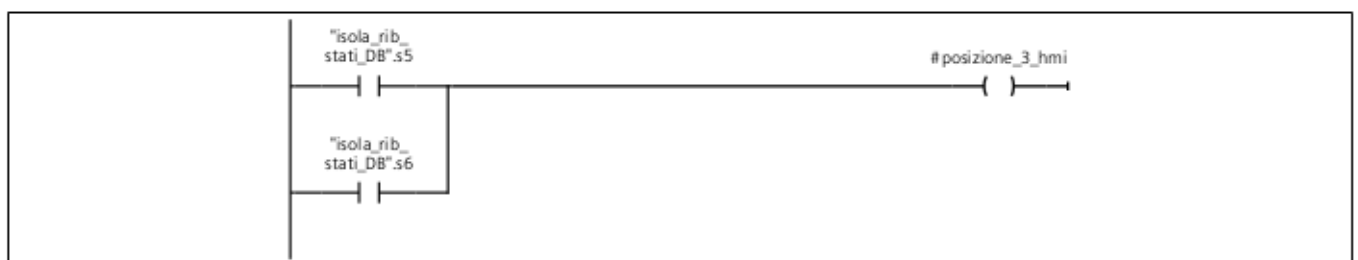
Network 19: Posizione semiribaltata cassetta HMI

Serve per indicare la posizione semiribaltata della cassetta nella seconda schermata del pannello operatore



Network 20: Posizione ribaltata cassetta HMI

Serve per indicare la posizione completamente ribaltata della cassetta nella seconda schermata del pannello operatore



Creazione Function Block che simula il processo

Poiché non possiamo testare quanto fatto in un impianto reale, ho deciso di creare una FB che simula il comportamento del sistema, ovvero simula la chiusura dei contatti di finecorsa e il rilevamento della cassetta nelle varie fotocellule.

Dal momento che questa è una funzione che non verrà utilizzata nella realtà, possiamo anche non preoccuparci più di tanto dell'eleganza e dell'efficienza del codice. La FB che ho creato è per comodità in SCL, ed è la seguente:

```
FUNCTION_BLOCK "simula_processoFB"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
VAR_INPUT
    sys_on : Bool;
    reset_man : Bool;
END_VAR

VAR_OUTPUT
    FT1 : Bool;
    FT2 : Bool;
    FT3 : Bool;
    FT4 : Bool;
    FC_salita1 : Bool;
    FC_salita2 : Bool;
    FC_discesa : Bool;
    pressostato_vuoto : Bool;
    posizione_cassetta : Int;
END_VAR

VAR
    timer1 {OriginalPartName := 'IEC_TIMER'; LibVersion := '1.0'} : TON_TIME;
    timer2 {OriginalPartName := 'IEC_TIMER'; LibVersion := '1.0'} : TON_TIME;
    timer3 {OriginalPartName := 'IEC_TIMER'; LibVersion := '1.0'} : TON_TIME;
    timer4 {OriginalPartName := 'IEC_TIMER'; LibVersion := '1.0'} : TON_TIME;
    timer5 {OriginalPartName := 'IEC_TIMER'; LibVersion := '1.0'} : TON_TIME;
    timer6 {OriginalPartName := 'IEC_TIMER'; LibVersion := '1.0'} : TON_TIME;
    timer7 {OriginalPartName := 'IEC_TIMER'; LibVersion := '1.0'} : TON_TIME;
    timer8 {OriginalPartName := 'IEC_TIMER'; LibVersion := '1.0'} : TON_TIME;
    timer9 {OriginalPartName := 'IEC_TIMER'; LibVersion := '1.0'} : TON_TIME;
    timer10 {OriginalPartName := 'IEC_TIMER'; LibVersion := '1.0'} : TP_TIME;
END_VAR

VAR_TEMP
    start_timer1 : Bool;
END_VAR

BEGIN
    (* In questa Function Block effettuo una semplice simulazione del processo.
    Simulo il caso più semplice di una sola cassetta nel nastro, la simulazione prevede
    - simulazione del passaggio nelle fotocellule FT1, FT2, FT3, FT4
```

- simulazione della chiusura dei fincorsa FC_salita1, FC_salita2, FC_discesa
*)

```
#start_timer1 := ("isola_rib_stati_DB".s0 OR #sys_on) AND NOT #timer2.IN AND NOT  
#timer3.IN AND NOT #timer4.IN AND NOT #timer5.IN AND NOT #timer6.IN AND NOT  
#timer7.IN AND NOT #timer8.IN AND NOT #FT3;
```

```
#timer1(IN:=#start_timer1, PT:=t#3s);  
#timer2(IN := #FT1 AND #sys_on,PT := t#2s);  
#timer3(IN := #FT2 AND #sys_on,PT := t#4s);  
#timer4(IN := "isola_rib_stati_DB".s2 AND #sys_on,PT := t#3s);  
#timer5(IN := "isola_rib_stati_DB".s3 AND #sys_on,PT := t#3s);  
#timer6(IN := "isola_rib_stati_DB".s4 AND #sys_on,PT := t#3s);  
#timer7(IN := "isola_rib_stati_DB".s6 AND #sys_on,PT := t#3s);  
#timer8(IN := "isola_rib_stati_DB".s8 AND #sys_on,PT := t#3s);  
#timer9(IN := "isola_rib_stati_DB".s9 AND #sys_on,PT := t#3s);  
#timer10(IN := #timer9.Q AND #sys_on,PT := t#3s);
```

```
IF #timer1.Q THEN  
    #FT1 := 1;  
    RESET_TIMER(#timer1);  
END_IF;
```

```
IF #timer2.Q THEN  
    #FT1 := 0;  
    #FT2 := 1;  
END_IF;
```

```
IF #timer3.Q THEN  
    #FT2 := 0;  
    #FT3 := 1;  
END_IF;
```

```
IF #timer4.Q THEN  
    #FC_salita1 := 1;  
END_IF;
```

```
IF #timer5.Q THEN  
    #pressostato_vuoto := 1;  
    #FC_salita1 := 0;  
END_IF;
```

```
IF #timer6.Q THEN  
    #FC_salita2 := 1;  
END_IF;
```

```
IF "isola_rib_stati_DB".s6 THEN  
    #FC_salita2 := 0;  
END_IF;
```

```
IF #timer7.Q THEN  
    #FC_salita1 := 1;  
END_IF;
```

```
IF #timer8.Q THEN
    #pressostato_vuoto := 0;
    #FC_salita1 := 0;
END_IF;

IF #timer9.Q THEN
    #FC_discesa := 1;
    #FT3 := 0;
END_IF;

#FT4 := #timer10.Q;

IF "isola_rib_stati_DB".s1 OR "isola_rib_stati_DB".s2 THEN
    #FC_discesa := 0;
END_IF;

IF #reset_man THEN
    RESET_TIMER(#timer1);
    RESET_TIMER(#timer2);
    RESET_TIMER(#timer3);
    RESET_TIMER(#timer4);
    RESET_TIMER(#timer5);
    RESET_TIMER(#timer6);
    RESET_TIMER(#timer7);
    RESET_TIMER(#timer8);
    RESET_TIMER(#timer9);
    RESET_TIMER(#timer10);
    #FT1 := 0;
    #FT2 := 0;
    #FT3 := 0;
    #FT4 := 0;
    #FC_salita1 := 0;
    #FC_salita2 := 0;
    #FC_discesa := 0;
END_IF;
```

```
END_FUNCTION_BLOCK
```

Richiamo le FB nel Main

A questo punto non ci resta che richiamare le FB nel Main, ovvero:

Totally Integrated Automation Portal		
--------------------------------------	--	--

nastro_trasportatore_isola_rib_PLCFForum / PLC_1 [CPU 1518-4 PN/DP] / Program blocks

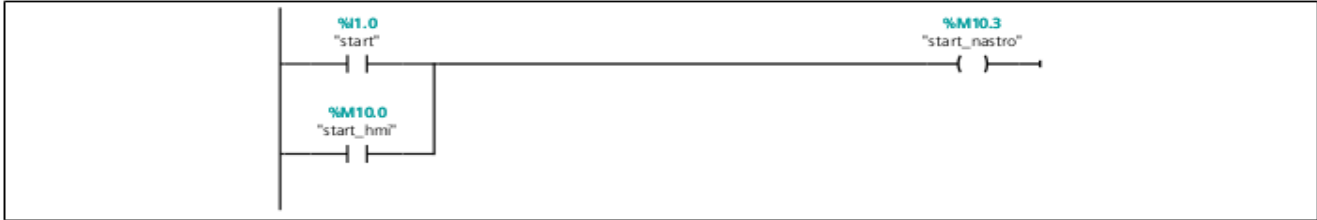
Main [OB1]

Main Properties					
General					
Name	Main	Number	1	Type	OB
Language	LAD	Numbering	automatic		
Information					
Title	"Main Program Sweep (Cycle)"	Author		Comment	
Family		Version	0.1	User-defined ID	

Main		
Name	Data type	Default value
▼ Input		
Initial_Call	Bool	
Remanence	Bool	
Temp		
Constant		

Network 1: Start Nastro trasportatore


Il pulsante Start è dato dal parallelo tra il pulsante Start Hardware (normalmente aperto) e quello virtuale nel pannello operatore



```
graph LR; M10["%M1.0 'start'"] --- AND1(( )); M100["%M10.0 'start_hmi'"] --- AND1; AND1 --- M103["%M10.3 'start_nastro'"]
```

Network 2: Stop Nastro trasportatore

Il pulsante Stop è dato dal parallelo tra il pulsante Stop Hardware (normalmente chiuso) e quello virtuale nel pannello operatore



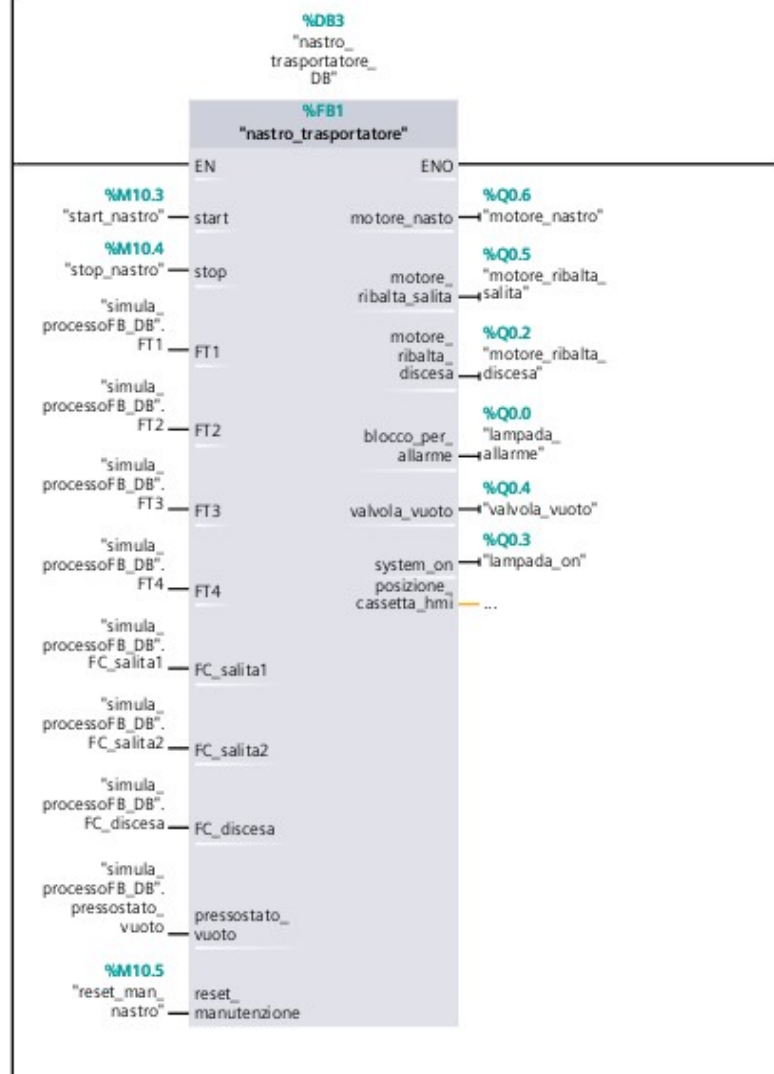
```
graph LR; M11["%M1.1 'stop'"] --- AND2(( )); M101["%M10.1 'stop_hmi'"] --- AND2; AND2 --- M104["%M10.4 'stop_nastro'"]
```

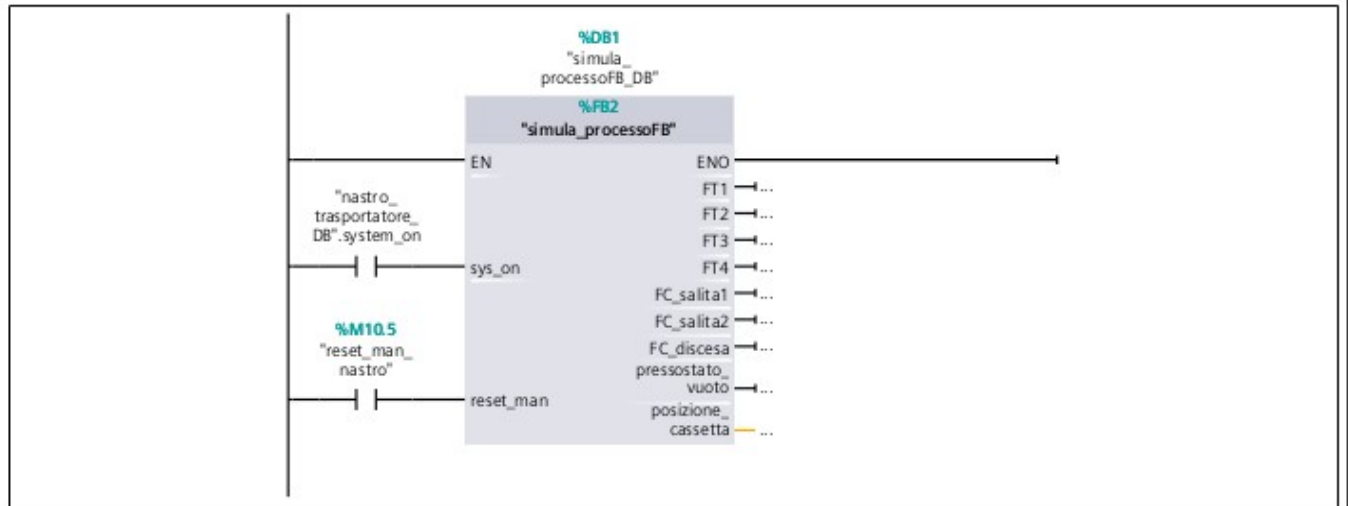
Network 3:

Il pulsante Reset Manutenzione è dato dal parallelo tra il pulsante Reset Manutenzione Hardware (normalmente aperto) e quello virtuale nel pannello operatore.



Network 4: Richiamo di un'istanza della function Block Nastro trasportatore



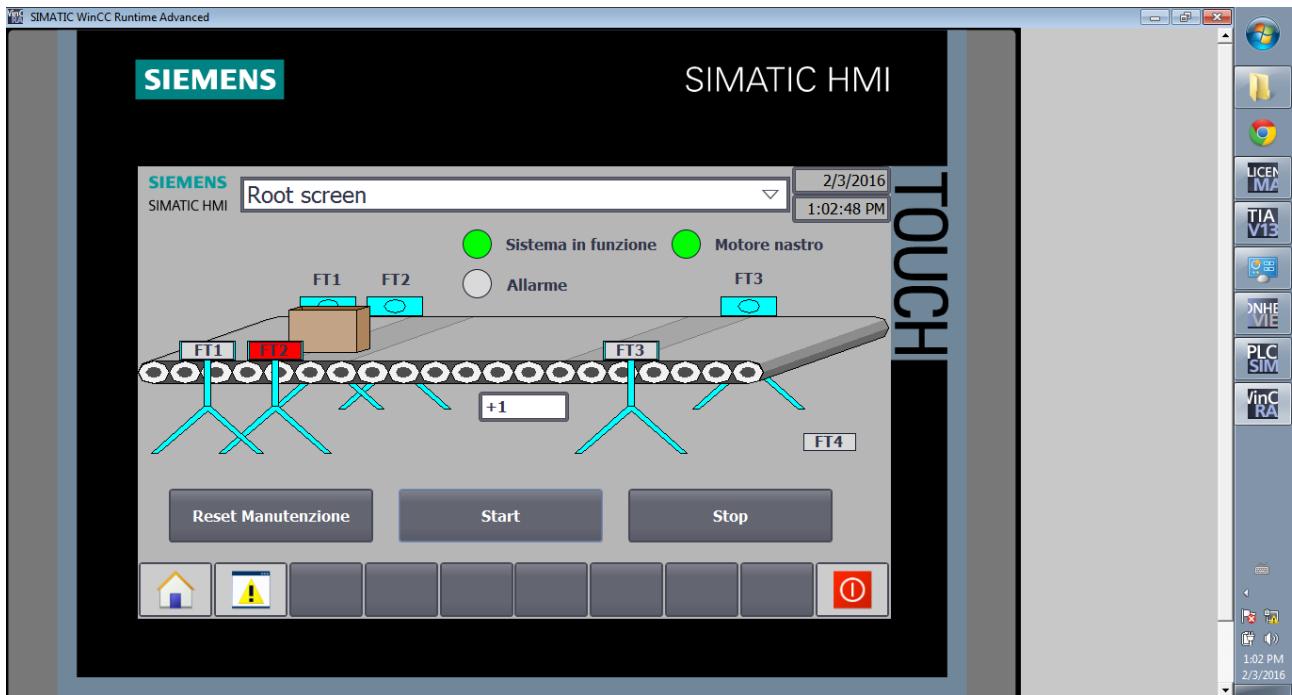


Pannello operatore

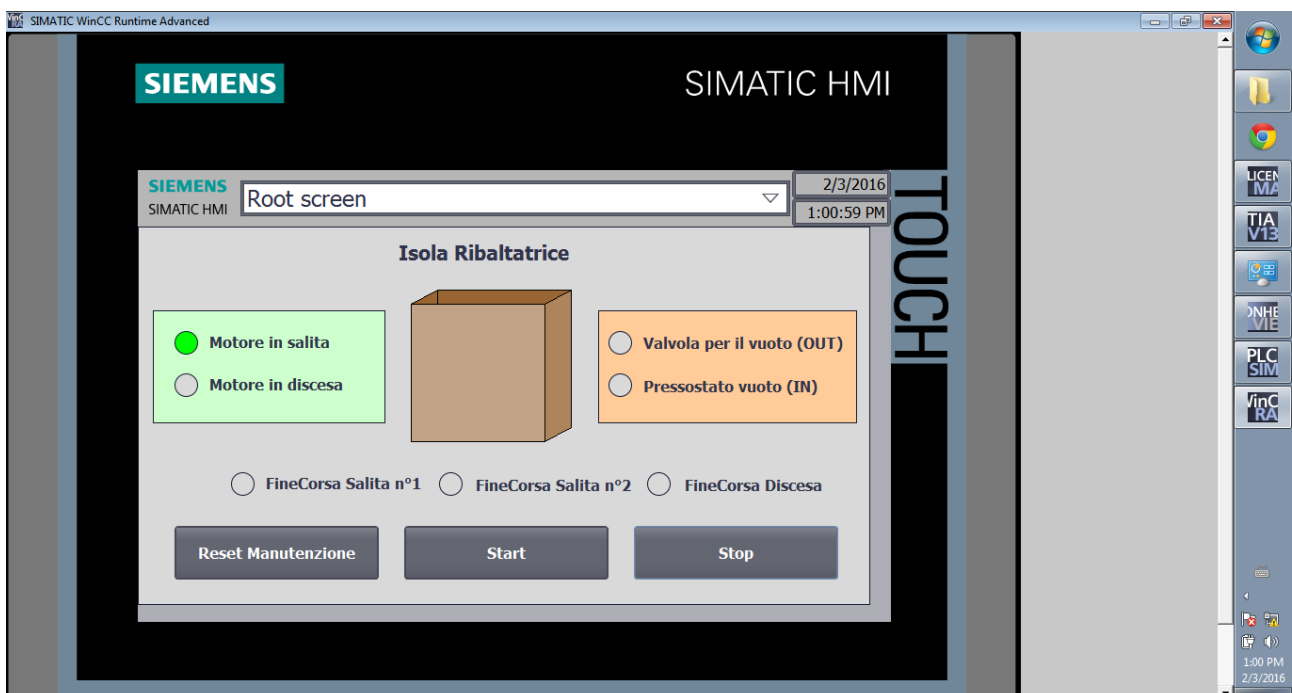
Nel progetto è stato inserito anche un pannello operatore in cui vengono visualizzate le varie operazioni e sono inoltre presenti i tre pulsanti.

In particolare sono state realizzate due schermate, la prima che riguarda il nastro trasportatore e la seconda l'isola ribaltatrice.

Schermata numero 1, nastro trasportatore:



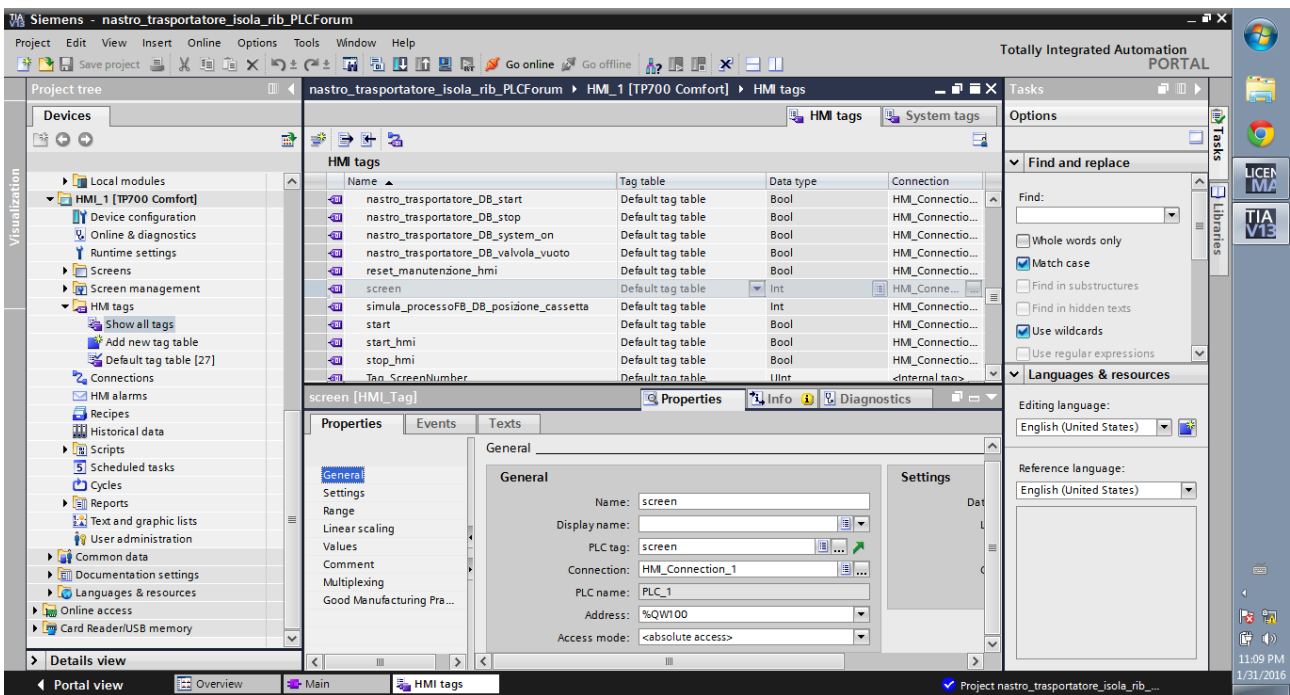
Schermata numero 2, isola ribaltatrice:



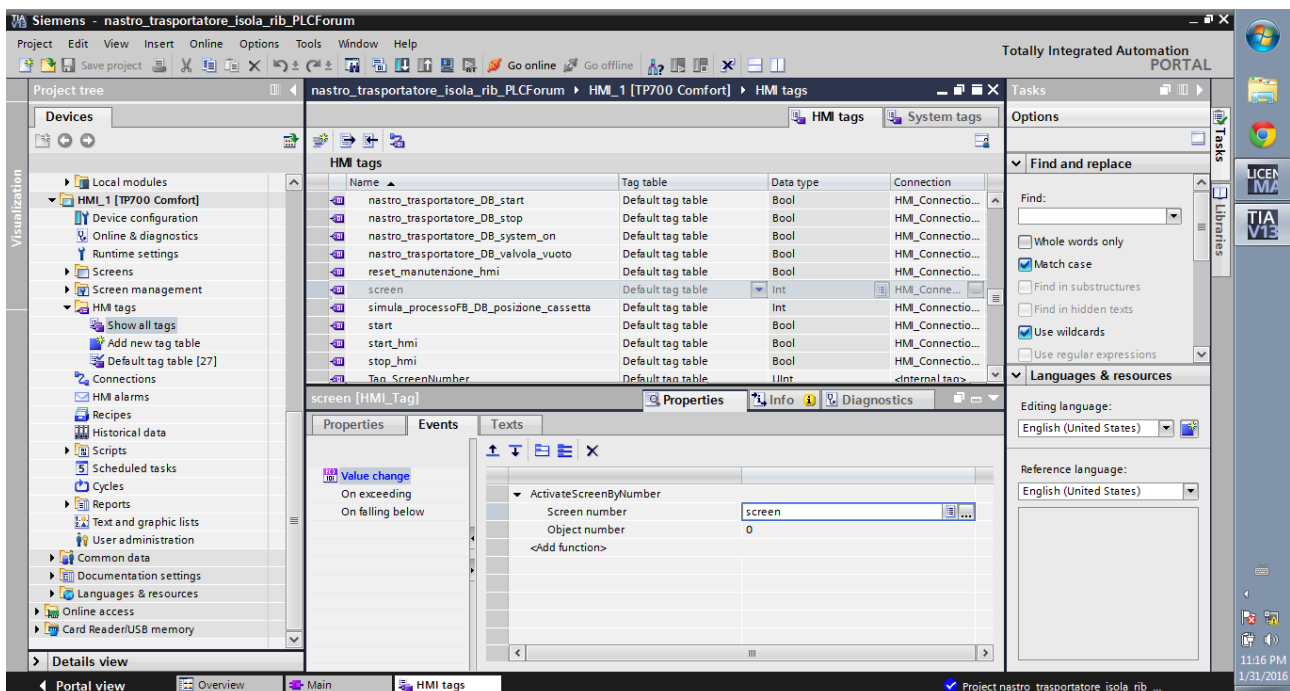
La schermata numero 1 viene visualizzata automaticamente quando l'isola ribaltatrice non è in funzione, mentre al contrario quella numero 2 viene automaticamente visualizzata quando il nastro è fermo e l'isola ribaltatrice è attiva.

Può essere utile descrivere come attivare una particolare schermata del pannello al variare di una determinata variabile del PLC, operazione che appunto è stata effettuata in questo caso.

Per fare questo definisco una variabile di tipo INT nel PLC e chiamo questa variabile “screen”. Collego poi tale variabile ad una variabile con lo stesso nome nel pannello operatore, ovvero:



A questo punto associo un evento alla variabile “screen” del pannello operatore cliccando su “Events” -> “Value Change” e selezionando l'evento “ActivateScreenByNumber”; come “screen number” imposterò poi la stessa variabile “screen”, ossia:



Quindi quando la variabile “screen” del PLC assume il valore 1 viene visualizzata la schermata numero 1, quando assume il valore 2 viene visualizzata la schermata numero 2 (in questo caso ho creato solo 2 schermate).

Considerazioni finali

Questo progetto è da considerarsi un buon punto di partenza per il controllo di un impianto reale. In base alle necessità è possibile non solo ampliare la gamma di operazioni che devono essere effettuate, ma anche implementare altre funzioni, quali ad esempio la gestione degli allarmi nel pannello operatore, il riposizionamento e/o inizializzazione del sistema, etc...

Download del progetto

E' possibile scaricare questo progetto tramite i seguenti link:

Mega.co.nz --> https://mega.nz/#!eLRhGZrZ!ffe1Q5WU144ozwyuC19_p-taxqPKAbfSJZdfIXmdVM

Google Drive --> <https://drive.google.com/file/d/0B0D8uzD2ZJGrbGU5Q25WZGlsTFk/view?usp=sharing>