

Olimpiadi dell'Automazione Siemens 2022

Istituto Tecnico Tecnologico Statale Guglielmo Marconi di Forlì
Classe 4A EN (Elettronica), a.s. 2021/22

Isola Robotica Automatizzata

Gruppo di lavoro:

1. Alessandro Lazzari: cablaggio, rete e programmazione Safety
2. Samuele D'Ambrosio: programmazione nastro trasportatore, pressa meccanica e comunicazione M2M
3. Lucas Prati: programmazione braccio robotico e macchina a stati per la lavorazione automatica
4. Edoardo Sasso: programmazione HMI

Docenti Referenti:

1. Roberto Versari: referente tecnico del progetto
2. Luigi Navacchia: supporto tecnico

Premessa

Il progetto rappresenta una sfida per il nostro Istituto per vari motivi:

- è la prima partecipazione in assoluto alle Olimpiadi dell'Automazione Siemens
- è la prima volta che viene affrontato l'ambiente di sviluppo TIA Portal in una classe quarta elettronica
- è la prima volta che si realizza un progetto di automazione di tipo avanzato

La partecipazione è stata resa possibile dalla disponibilità di 4 studenti coraggiosi che hanno accettato questa sfida impegnativa.

Il progetto è stato scelto da loro sulla base dei modellini di sistemi automatici in dotazione al laboratorio di Sistemi Automatici: un modellino di nastro trasportatore e di una pressa meccanica, spiegati e programmati durante le lezioni di Sistemi Automatici, e un braccio robotico con 4 assi.

Per rendere il progetto più realistico è stato realizzato un sistema distribuito, dove un PLC controlla la pressa, e un altro PLC controlla il braccio robotico e il nastro trasportatore. I due PLC comunicano tra loro utilizzando il protocollo S7.

Sono stati poi aggiunti alcuni componenti essenziali per rendere completo il progetto da un punto di vista didattico:

- una cella di carico per la misura analogica del peso dei pezzi lavorati
- dei dispositivi safety per garantire la sicurezza dell'impianto
- un pannello operatore per il monitoraggio del sistema

La programmazione degli assi del braccio robotico, la comunicazione Machine to Machine (M2M), la misura del peso dei pezzi dall'ingresso analogico, la programmazione dei dispositivi safety non sono stati affrontati durante le lezioni di Sistemi Automatici: il lavoro di progettazione è stato quasi interamente svolto in orario extra-curricolare, sfruttando i pomeriggi in cui era aperto il Gruppo Prototipi dell'Istituto, una sorta di FabLab interno al nostro Istituto.

La difficoltà maggiore è stata proprio quella di riuscire in orario extra-curricolare, senza trascurare le attività didattiche ordinarie, a completare il progetto in tempo.

Il progetto si configura quindi come un progetto di eccellenza, che va ben al di là della programmazione curricolare delle classi quarte Elettronica.

Introduzione

L'isola robotica è composta da 3 impianti principali:

- un nastro trasportatore
- una pressa
- un braccio robotico

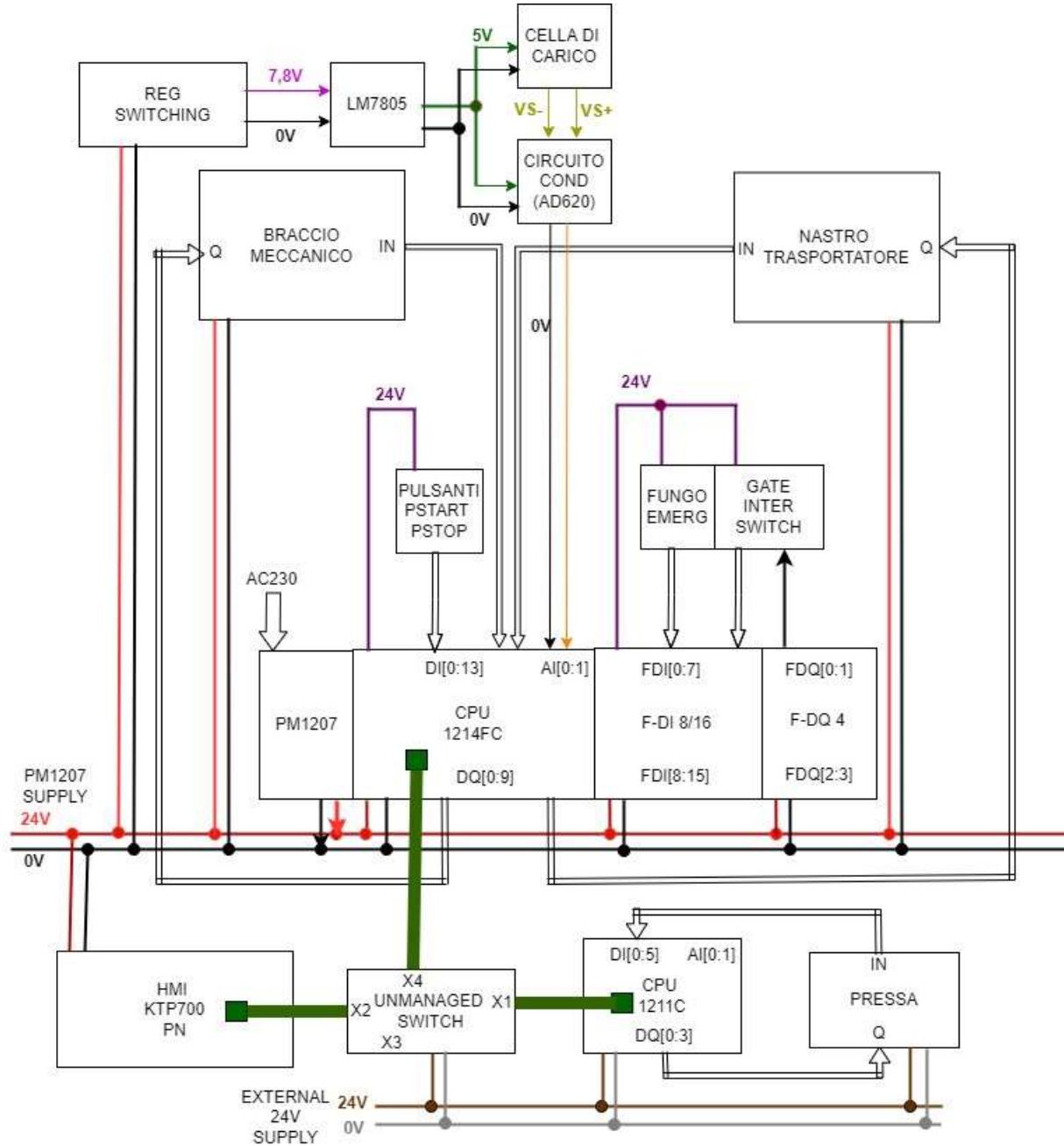
Vi sono poi una cella di carico/bilancia per pesare i pezzi, e un container per i pezzi scartati.

Il sistema ha un solo punto di alimentazione/prelievo: il nastro trasportatore. Una volta che viene inserito un pezzo nel nastro trasportatore, si avvia il processo automatico di lavorazione: il braccio robotico preleva il pezzo dal nastro e lo porta sulla cella di carico per la pesatura. Se il peso misurato è maggiore di 15 grammi il pezzo prosegue la lavorazione e viene portato sulla pressa, altrimenti viene scartato nel container. I pezzi buoni, dopo la pressatura, vengono prelevati dal braccio e riportati sul nastro per essere resi nuovamente disponibili nell'unico punto di alimentazione/prelievo.

Se per qualche motivo (ad esempio un'emergenza) il processo di lavorazione viene interrotto a metà, per sicurezza il braccio robotico esegue l'operazione di scarto del pezzo e il sistema riparte in attesa di un nuovo pezzo.

Schema a blocchi del sistema

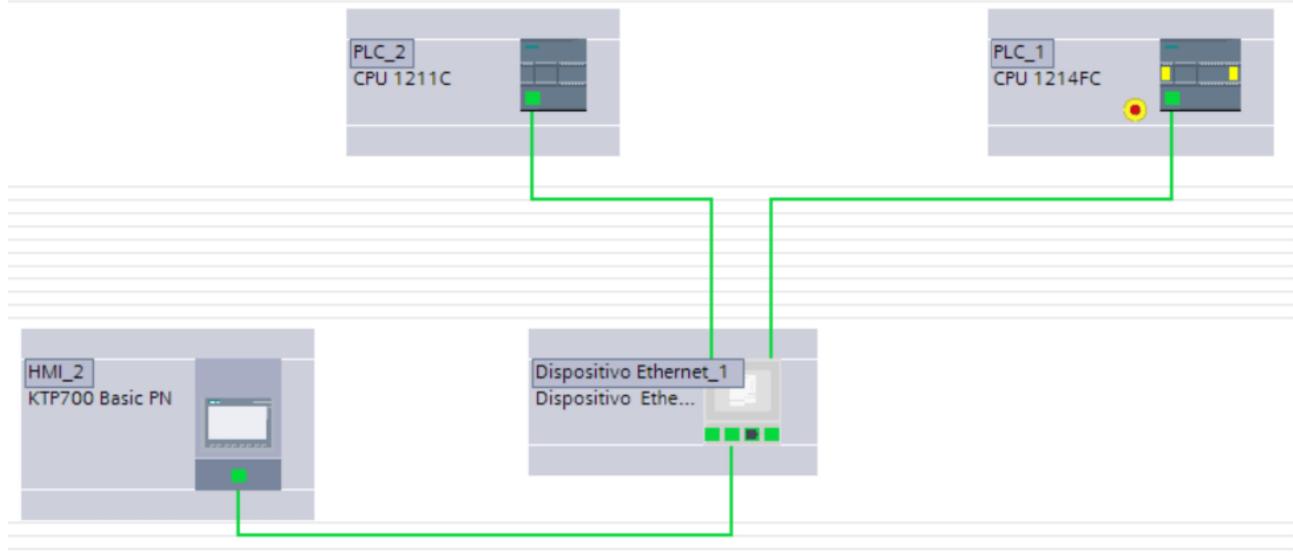
Figura 1: schema a blocchi del sistema



Il sistema è composto da due CPU, una 1214FC (PLC_1) e una 1211C (PLC_2), entrambe di tipo DC/DC/DC. Le due CPU sono collegate in rete tra loro tramite un unmanaged switch, e comunicano sfruttando il protocollo S7 tramite i blocchi funzione PUT e GET. Nella comunicazione PUT/GET la CPU 1214FC è la CPU master, mentre la 1211C è la CPU slave.

Un pannello operatore basic a 7 pollici, il KTP700 Basic PN, è collegato in rete tramite lo stesso switch. Lo switch ha anche una quarta porta Ethernet dove può essere collegato il PC con TIA Portal per mettere online i due PLC.

Figura 2: collegamenti di rete tra le due CPU e il pannello operatore



La CPU 1214FC (PLC_1 o PLC master) è una CPU safety ed è alimentata da un modulo di alimentazione PM1207 24V/2,5A. Il modulo di alimentazione viene utilizzato per alimentare anche:

- il braccio robotico
- il nastro trasportatore
- il pannello operatore
- i regolatori per la cella di carico

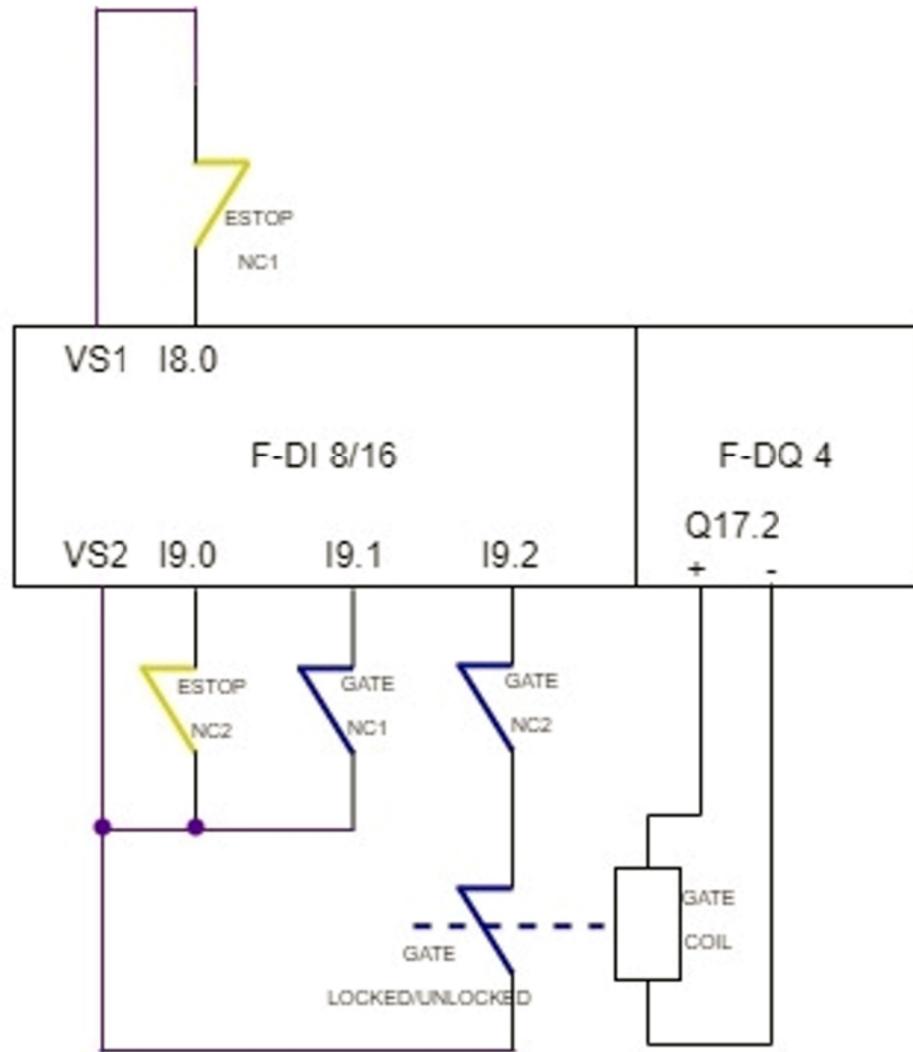
La CPU 1214FC ha due moduli di espansione di tipo safety:

1. F-DI 8/16x24VDC_1Q4X, unità di ingressi digitali con 16 ingressi singoli o 8 ridondati, DC24V, PROFIsafe V2.
2. F-DQ 4x24VDC_1, unità di uscite digitali con 4 uscite, DC24V, PROFIsafe V2.

Due ingressi safety ridondati sono utilizzati per gestire un fungo di emergenza con due contatti NC; due ingressi safety singoli sono utilizzati per gestire un gate interlock switch, anch'esso con due contatti NC. Un'uscita safety è utilizzata per sbloccare/bloccare il gate interlock switch (lock/unlock).

Il ripristino dei dispositivi safety è gestito tramite pulsanti inseriti nel pannello operatore (non sono presenti pulsanti fisici esterni per il ripristino).

Figura 3: collegamenti dei dispositivi Safety. Il fungo di emergenza è collegato agli ingressi %I8.0 e %I9.0 configurati come 1oo2, un unico canale ridondato. Nel programma si farà riferimento al solo ingresso %I8.0. Il gate interlock switch è collegato agli ingressi %I9.1 e %I9.2 configurati 1oo1, cioè come due canali separati. Attivando l'uscita safety %Q17.2 si eccita la bobina del gate interlock switch e si sblocca la serratura, aprendo nel contempo il terzo contatto NC.



La CPU 1214FC gestisce anche due pulsanti esterni per l'accensione/spegnimento del sistema. Questi pulsanti sono presenti anche nel pannello operatore.

Il primo ingresso analogico della CPU, %IW64, è collegato all'uscita del circuito di condizionamento della cella di carico.

La cella di carico è un ponte di Wheatstone a estensimetri progettato per essere alimentato a 5V. Fornisce un'uscita differenziale proporzionale al peso, con tensione di modo comune di circa 2,5V. Il segnale differenziale fornito dalla cella di carico è molto basso e richiede un amplificatore per strumentazione per essere amplificato senza disturbi [8].

Per alimentare a 5V la cella di carico abbiamo utilizzato un alimentatore a doppio stadio progettato nel laboratorio di TPSEE: un primo regolatore switching di tipo Buck che da 24V fornisce 7,8V circa, con efficienza di circa il 90%; un secondo regolatore lineare, il classico LM7805, che riceve i 7,8V in ingresso e fornisce 5V in uscita. Il regolatore

lineare alimenta, oltre alla cella di carico, anche l'amplificatore per strumentazione, un AD620.

Inizialmente si era utilizzato il solo regolatore switching per ottenere direttamente 5V dai 24V del PM1207, ma il ripple in uscita tipico di questi regolatori causava errori di misura del peso. Con il doppio stadio di regolazione le misure del peso sono affidabili e ripetibili e l'efficienza energetica del regolatore è accettabile.

L'amplificatore per strumentazione utilizzato presenta due trimmer regolabili: uno per compensare la tensione di modo comune del segnale differenziale della cella di carico (regolazione dell'offset). In fase di debug, senza alcun pezzo inserito sulla cella di carico, questo trimmer è stato regolato in modo da misurare 0V all'ingresso analogico.

Il secondo trimmer viene usato per regolare il guadagno: in fase di debug, con un peso campione di 20 grammi inserito sulla cella di carico, è stato regolato in modo da avere un fondo scala di circa 1030 grammi (fissato il fondo scala a 1030 grammi si è regolato il trimmer sino a che il peso misurato dall'ingresso analogico non era pari a quello del peso campione).

Si noti che l'amplificatore per strumentazione è alimentato a 5V, pertanto il segnale analogico di uscita varia tra 0 e 5V, mentre l'ingresso analogico della CPU 1214FC ha un range da 0 a 10V. Nonostante abbiano sfruttato solo metà del range di ingresso della CPU, la risoluzione finale della misura è pari a 1 grammo e le misure sono precise e ripetibili.

La CPU1214FC gestisce direttamente due dei tre impianti del sistema:

1. il nastro trasportatore
2. il braccio robotico

Nella tabella 1 sono mostrati gli I/O relativi alla CPU1214FC, con il loro significato.

Come si vede, rimane libero solo il secondo ingresso analogico della CPU, tutti gli altri sono stati utilizzati.

Nella CPU1214FC sono stati abilitati i marker di clock, con indirizzo iniziale impostato a 100. Infatti si è utilizzato il marker Clock_1Hz, %M100.5, nel blocco funzione GET, per prelevare ogni secondo i dati provenienti dal PLC slave.



Tabella 1: I/O della CPU 1214FC (PLC_1, master)

Nome	Tipo	Indirizzo	Tabella delle variabili	Commento
DatoPeso	Int	%IW64	Tabella delle variabili standard	Valore digitale corrispondente al peso misurato
	Int	%IW66		
FaseAEncSuGiù	Bool	%I0.0	Tabella delle variabili standard	Fase A Encoder asse verticale
FaseBEncSuGiù	Bool	%I0.1	Tabella delle variabili standard	Fase B Encoder asse verticale
FaseAEncRot	Bool	%I0.2	Tabella delle variabili standard	Fase A Encoder rotazione
FaseBEncRot	Bool	%I0.3	Tabella delle variabili standard	Fase B Encoder rotazione
FaseEncAvanti	Bool	%I0.4	Tabella delle variabili standard	Fase Encoder asse Avanti/Indietro
FineCorsaApMa...	Bool	%I0.5	Tabella delle variabili standard	Fine corsa apertura morsa
FaseEncChiusur..	Bool	%I0.6	Tabella delle variabili standard	Fase encoder apertura/chiusura morsa
FineCorsaIndietrc	Bool	%I0.7	Tabella delle variabili standard	Fine corsa asse Indietro
FineCorsaSu	Bool	%I1.0	Tabella delle variabili standard	Fine corsa asse Sù
FineCorsaRot	Bool	%I1.1	Tabella delle variabili standard	Fine corsa asse rotativo
PStart	Bool	%I1.2	Tabella delle variabili standard	Pulsante di avvio del sistema
PStop	Bool	%I1.3	Tabella delle variabili standard	Pulsante di arresto del sistema
FCIngn	Bool	%I1.4	Tabella delle variabili standard	Fotocellula di ingresso nastro (punto di alimentazione/prelievo del sistema)
FCUscN	Bool	%I1.5	Tabella delle variabili standard	Fotocellula di uscita nastro (punto di prelievo/immissione del braccio)
ApriMano	Bool	%Q0.0	Tabella delle variabili standard	Motore per l'apertura della morsa
ChiudiMano	Bool	%Q0.1	Tabella delle variabili standard	Motore per la chiusura della morsa
BraccioAvanti	Bool	%Q0.2	Tabella delle variabili standard	Motore per mandare avanti il braccio
BraccioIndietro	Bool	%Q0.3	Tabella delle variabili standard	Motore per mandare indietro il braccio
BraccioGiu	Bool	%Q0.4	Tabella delle variabili standard	Motore per mandare in basso il braccio
BraccioSu	Bool	%Q0.5	Tabella delle variabili standard	Motore per mandare in alto il braccio
BraccioRotOr	Bool	%Q0.6	Tabella delle variabili standard	Motore per la rotazione in senso orario del braccio
BraccioRotAntiOr	Bool	%Q0.7	Tabella delle variabili standard	Motore per la rotazione in senso antiorario del braccio
MotAvN	Bool	%Q1.0	Tabella delle variabili standard	Motore per mandare avanti il nastro
MotIndN	Bool	%Q1.1	Tabella delle variabili standard	Motore per mandare indietro il nastro

La CPU 1211C (PLC_2 o PLC slave) gestisce la pressa meccanica. Essa è alimentata da un alimentatore esterno da banco regolato a 24V DC. Il progetto del blocco funzione relativo all'automazione della pressa è stato sviluppato durante il corso di Sistemi Automatici. L'unica modifica necessaria è l'aggiunta dei blocchi dati non ottimizzati che devono contenere le variabili/segnali che le due CPU si devono scambiare. In particolare, la CPU master deve comunicare alla pressa se il sistema è acceso o no (variabile SystemOn del blocco dati DatiHMIAssi), mentre la CPU slave deve comunicare al master quando il pezzo è pronto per essere prelevato dalla pressa (variabile PrelevaDaPressa del blocco dati DatiLavorazione).

I blocchi funzione PUT e GET sono inseriti nella CPU master, ma devono fare riferimento a variabili della CPU slave indirizzate in modo assoluto. Nel blocco funzione PUT la CPU master trasferisce il contenuto della sua variabile SystemOn nella variabile bool all'indirizzo 0.0 del blocco dati DB1, ossia la variabile SystemOn del blocco dati ReceivedData della CPU slave. Questo segnale viene usato come abilitazione della pressa. Nel blocco funzione GET, invece, la CPU master va a copiare il contenuto della variabile bool all'indirizzo 0.0 del blocco dati DB2 del PLC slave, che corrisponde alla variabile ProntoDaPrelevare del blocco dati SentData, nella variabile PrelevaDaPressa del blocco dati DatiLavorazione del PLC master. In questo modo quando la pressa segnala che c'è un pezzo pressato pronto da prelevare, la variabile PrelevaDaPressa della

CPU master va a 1 e il braccio robotico afferra il pezzo lavorato e lo trasporta in uscita verso il nastro trasportatore.

La tabella 2 mostra i segnali di I/O della CPU1211C, con il loro significato.

Tabella 2: I/O della CPU 1211C (PLC_2, slave)

Nome	Tipo	Indirizzo	Tabella delle variabili	Commento
	Int	%IW64		
	Int	%IW66		
FotocellulaIngr...	Bool	%IO.0	Tabella delle variabili standard	Fotocellula di prelievo/rimmissione pezzi
FotocellulaUscita	Bool	%IO.1	Tabella delle variabili standard	Fotocellula lavorazione pressa
FineCorsaBasso	Bool	%IO.2	Tabella delle variabili standard	Fine corsa pressa giù
FineCorsaAlto	Bool	%IO.3	Tabella delle variabili standard	Fine corsa pressa sù
Tag_1	Bool	%IO.4	Tabella delle variabili standard	n.c.
Tag_2	Bool	%IO.5	Tabella delle variabili standard	n.c.
MotoreNastroA...	Bool	%Q0.0	Tabella delle variabili standard	Motore Nastro Avanti
MotoreNastroIn...	Bool	%Q0.1	Tabella delle variabili standard	Motore Nastro Indietro
PressaGiù	Bool	%Q0.2	Tabella delle variabili standard	Motore pressa giù
PressaSu	Bool	%Q0.3	Tabella delle variabili standard	Motore pressa sù

Le Tabelle 3 e 4 mostrano i blocchi dati non ottimizzati dichiarati nella CPU slave e utilizzati per la comunicazione PUT/GET con la CPU master.

Tabella 3: blocco dati non ottimizzato ReceivedData della CPU 1211C (PLC_2, slave)

ReceivedData					
	Nome	Tipo di dati	Offset	Valore di avvio	Ritenzione
1	Static				
2	SystemOn	Bool	0.0	false	
3	MotorOn	Bool	0.1	false	

La variabile MotorOn è riservata per usi futuri, nel caso in cui le luci di segnalazione siano gestite dalla CPU slave. Un'idea di sviluppo futuro del progetto è quella di aggiungere una signal board con 4 uscite digitali 24V o un modulo di espansione con uscite digitali per pilotare le luci di segnalazione, non presenti in questo progetto per mancanza di output fisici. La variabile MotorOn contiene lo stato degli azionamenti della CPU master, da cui dipendono le luci di segnalazione.

Tabella 4: blocco dati non ottimizzato SentData della CPU 1211C (PLC_2, slave)

SentData								
Nome	Tipo di dati	Offset	Valore di avvio	Ritenzione	Accessibile ...	Scrivi...	Visibile in ...	
Static								
ProntoDaPrelevare	Bool	0.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Dato2	Bool	0.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

La variabile Dato2 non è al momento utilizzata ed è riservata per usi futuri.

Tabella esplicativa delle variabili e dei segnali utilizzati

Più che una tabella esaustiva di tutti i blocchi dati e segnali utilizzati (che sono un centinaio) descriviamo i segnali più significativi di ogni blocco dati associandoli al relativo blocco funzione/segmento di programma, partendo dalla CPU master PLC_1.

Descrizione variabili e segnali del Main del PLC_1

Segmento 1 – accensione spegnimento del sistema tramite pulsantiera esterna o pulsanti HMI

Il segmento 1 è utilizzato per l'accensione/spegnimento del sistema tramite pulsantiera esterna o pulsanti HMI. È costituito da un FFSR a reset dominante, per dare priorità ai segnali di spegnimento rispetto a quelli di accensione. Fa riferimento ai blocchi dati DatiHMIAssi e DatiFromSafety e alle variabili PLC PStart (%I1.2) e PStop (%I1.3).

Il sistema può essere acceso tramite il pulsante fisico esterno di tipo NA collegato a %I1.2, oppure tramite il pulsante HMI (variabile PStart del blocco dati DatiHMIAssi).

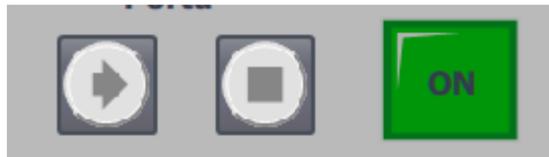
Per lo spegnimento abbiamo varie possibilità:

- attivazione o malfunzionamento di un dispositivo safety (il segnale SafetyOk del blocco dati di tipo failsafe DatiFromSafety va a 0)
- pressione del pulsante fisico esterno di tipo NC collegato a %I1.3, oppure del corrispondente pulsante HMI (variabile PStop del blocco dati DatiHMIAssi).
- pressione dello switch HMI stop di emergenza (variabile JogEmergenza del blocco dati DatiHMIAssi). Questo ultimo switch in realtà è ridondante, ma è stato utilizzato come sistema di blocco degli assi meccanici nelle prime fasi di debug del programma ed è stato mantenuto per comodità anche nel seguito del progetto.

L'uscita del FFSR è il segnale SystemOn del blocco dati DatiHMIAssi, che abilita tutti gli attuatori del sistema: se questo segnale è 0 nessun motore può essere azionato.

Sul fronte di salita del segnale SystemOn, ovvero ad ogni accensione del sistema, viene attivata la variabile AppoggioFrSOn del blocco dati DatiHMIAssi che è utilizzata dalla macchina a stati per avviare il processo di scarto di un pezzo ogni volta che il sistema viene riavviato (utile nel caso in cui il processo di lavorazione sia stato interrotto a metà).

Figura 4: Pulsanti PStart e PStop del pannello operatore con luce di segnalazione associata alla variabile SystemOn



La figura mostra una parte del pannello operatore che fa riferimento alle variabili usate nel segmento 1: il pulsante di sinistra controlla la variabile PStart e può essere usato per l'accensione del sistema in alternativa al pulsante fisico. Il pulsante al centro controlla la variabile PStop e può essere usato per lo spegnimento del sistema in alternativa al pulsante fisico. La luce di segnalazione a destra fa riferimento alla variabile SystemOn e si accende quando questa è a 1 (sistema abilitato).

Questi oggetti sono inseriti nella pagina modello del HMI e quindi sono disponibili in tutte le pagine del HMI.

Segmenti 2, 3, 4 e 5 - Blocchi funzione per la movimentazione degli assi del braccio meccanico

I blocchi funzione relativi agli assi del braccio meccanico sono stati progettati a partire dai tutorial Siemens sul S7-1200 disponibili online [1].

Ogni blocco funzione associato a un asse è in grado di lavorare in due modalità, manuale e automatica. Nella modalità manuale tramite dei pulsanti su pannello operatore si può muovere in una direzione o nell'altra l'asse (pagina Man). Nella modalità automatica si fissa il set point dell'encoder incrementale e con un pulsante su pannello operatore si invia il comando di azionamento degli assi, che si muoveranno simultaneamente sino al raggiungimento del set point fissato per ciascuno.

La modalità manuale è stata molto utile nelle fasi di calibrazione/debug del sistema per individuare i set point corrispondenti ai punti di prelievo/immissione di nastro, pressa, cella di carico e container scarti.

La modalità automatica è stata sfruttata nella macchina a stati per la lavorazione automatica dei pezzi, che non consiste in altro che nel raggiungere, nella giusta sequenza, i set point corrispondenti ai punti di immissione/prelievo di nastro, pressa, cella di carico e container.

I blocchi funzione relativi agli assi utilizzano i contatori veloci (HSC) della CPU 1214C. Gli assi rotativo e verticale hanno un encoder a due fasi, AB, e usano gli HSC 1 e 2, rispettivamente. Gli HSC con due fasi sono in grado di incrementare/decrementare il conteggio in automatico quando l'asse va in una direzione o nell'altra [2].

Viceversa, gli assi Avanti/Indietro e quello relativo alla morsa hanno un encoder a una fase, e usano gli HSC 3 e 4. In questo caso bisogna comunicare al HSC la direzione del conteggio, in modo che incrementi quando il movimento è in una direzione e decrementi quando è nell'altra [2].

Per questo motivo c'è un blocco funzione per gli assi rotativi e verticale, con encoder a due fasi, di nome GestioneAsse_AB, e un altro blocco funzione per l'asse avanti indietro e per la morsa, con encoder a una fase, di nome GestioneAsse_Monofase.

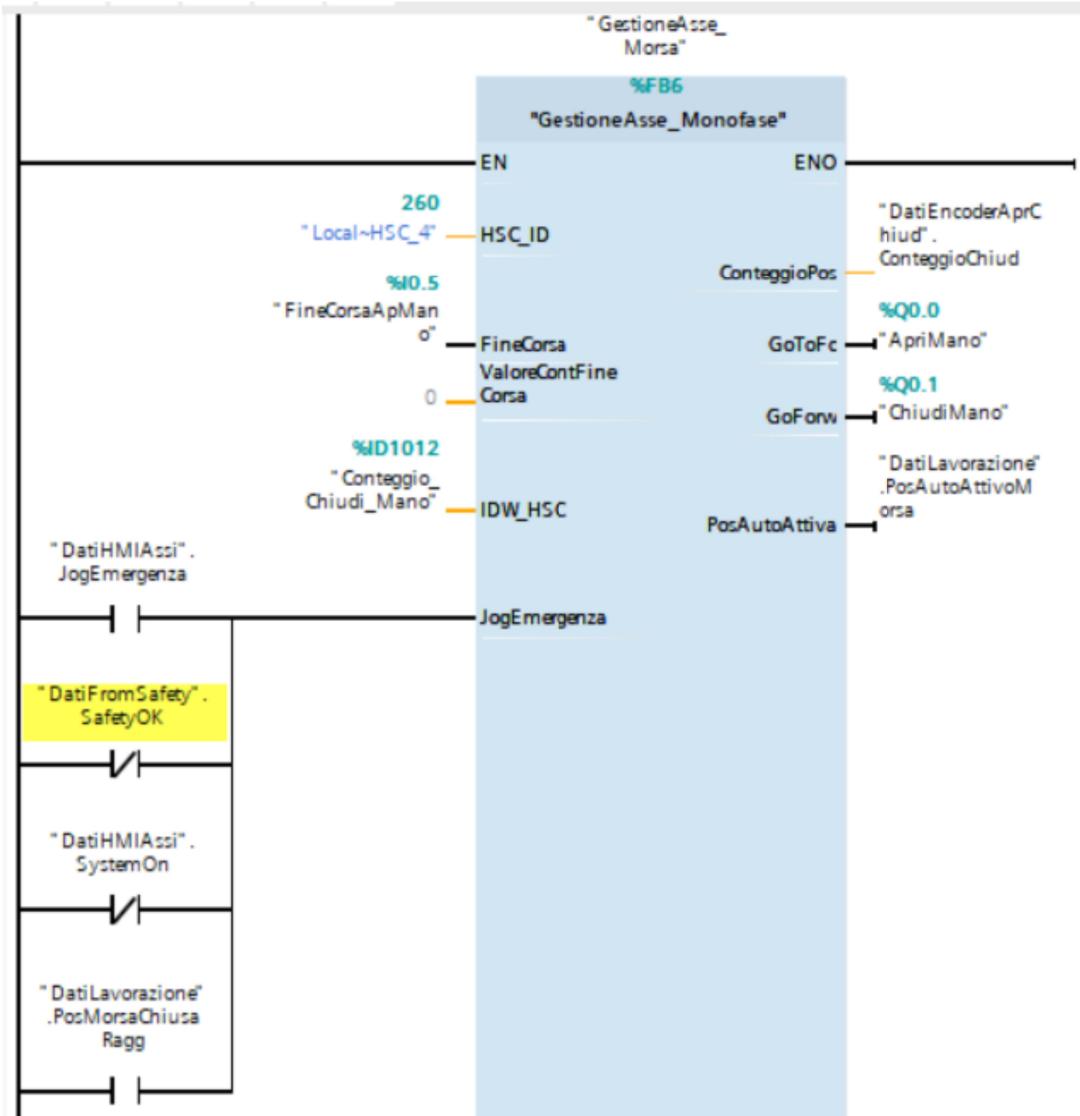
Gli assi avanti/indietro e la morsa hanno inoltre evidenziato una scarsa robustezza meccanica, dovuta al fatto che il sistema è in uso dal 2015 e ha subito alcuni danneggiamenti. Per l'asse avanti indietro si è osservata una diversa trazione a seconda della direzione di movimento: il conteggio in avanti è più veloce rispetto quello all'indietro (le ruote dentate quando l'asse va avanti perdono alcuni giri).

La morsa presenta ancora più problemi: la trasmissione meccanica si interrompe del tutto quando la morsa viene chiusa troppo, e a seconda del livello di adesione delle parti meccaniche, il valore di conteggio dell'encoder per afferrare un pezzo può variare da 37 a 54. Questo crea grossi problemi quando si vuole raggiungere il set point per afferrare un pezzo: se nella macchina a stati è stato impostato il valore di set point 46 ma la morsa ha già stretto il pezzo a 40 e non può chiudersi di più, il motore continua a girare sforzando la morsa. Viceversa, se si imposta il set point a 40 ma il pezzo richiede un set point a 46, la morsa non riesce a stringere il pezzo, che afferrato in modo precario, cadrà.

Non è un caso che i modellini di bracci robotici prodotti di recente dallo stesso costruttore abbiano un diverso sistema per prendere i pezzi, con una pinza di tipo pneumatico dotata di compressore ed elettrovalvole: il pezzo viene fatto aderire alla pinza creando il vuoto.

Nel nostro progetto per superare il problema del set point variabile per afferrare un pezzo si è aggiunto un temporizzatore che limita la stretta a un tempo massimo di 2 secondi, dopo il quale la chiusura della morsa viene interrotta, anche se non si è raggiunto il set point programmato. Questo tempo non è stato necessario parametrizzarlo nel pannello operatore. Inoltre il set point corrispondente alla presa di un pezzo è stato reso programmabile da pannello operatore nella pagina di debug della morsa, in modo che se si osserva un cambiamento significativo si può modificare il suo valore.

Figura 5: Blocco funzione associato all’asse della morsa – parte 1



L’ID 260 fa riferimento al HSC 4, cui corrisponde l’ingresso digitale %I0.6 (FaseEncChiusuraMano delle variabili PLC), che va collegato all’encoder della morsa. Tutti gli ingressi di conteggio degli HSC sono stati configurati con un filtro d’ingresso di valore basso, dell’ordine della decina di microsecondi.

A ogni asse del braccio meccanico corrisponde un fine corsa attivo alto, nel caso della morsa l’ingresso %I0.5 (FineCorsaApMano delle variabili PLC). Quando questo segnale diventa 1 il valore di conteggio degli HSC viene posto a 0 e l’asse è nella posizione di Home.

A ogni contatore veloce corrisponde un valore di conteggio di tipo DInt, doppio intero (32 bit o 4 byte): nel caso della morsa/HSC4 questo valore è dato dalla variabile PLC %ID1012 Conteggio_Chiudi_Manu.

L’ingresso JogEmergenza del blocco funzione è un segnale molto importante, perché la sua attivazione (JogEmergenza=1) disabilita immediatamente i motori associati all’asse. Nel caso della morsa lo spegnimento dei motori per l’apertura/chiusura della mano può avvenire in 4 casi diversi:

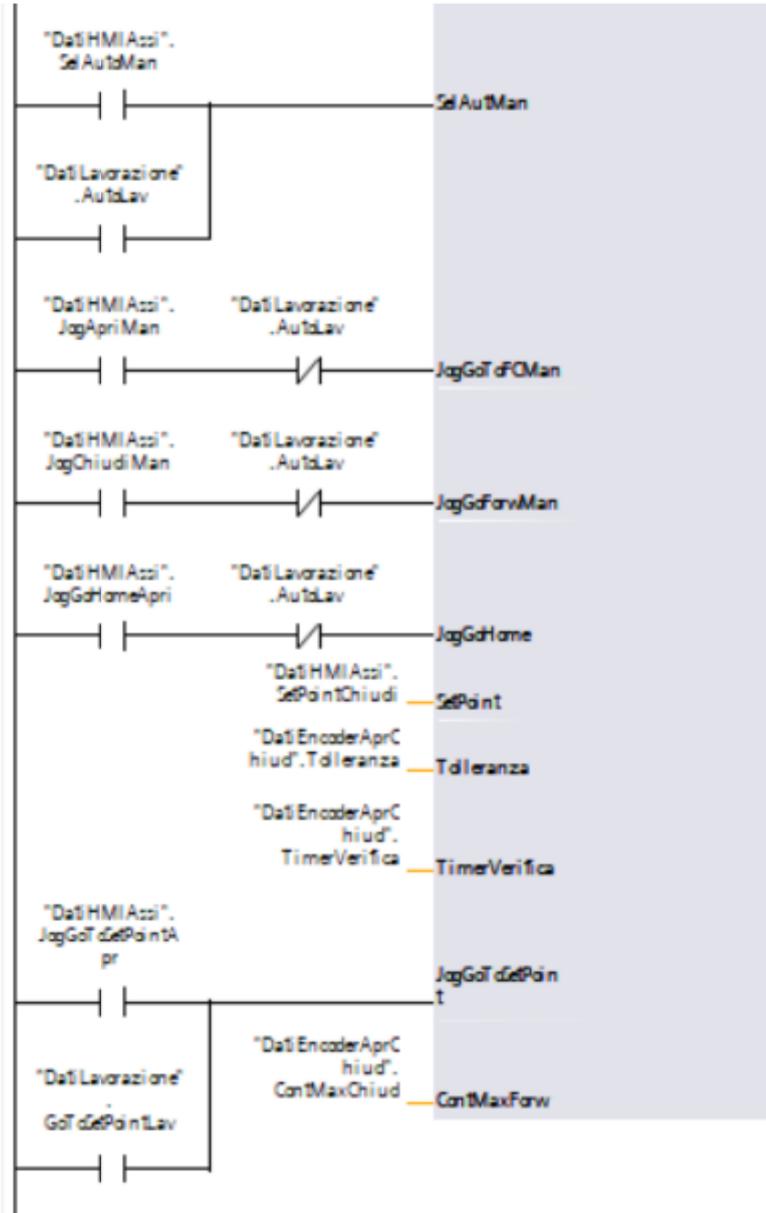
- pressione del pulsante di stop di emergenza del HMI associato alla variabile JogEmergenza del blocco dati DatiHMIAssi
- attivazione o malfunzionamento di un dispositivo safety (il segnale SafetyOk del blocco dati di tipo failsafe DatiFromSafety va a 0)
- spegnimento del sistema tramite i pulsanti PStop fisici o del HMI (il segnale SystemOn del blocco dati DatiHMIAssi va a 0)
- fine conteggio del timer che limita la durata di chiusura della morsa (il segnale PosMorsaChiusaRagg del blocco dati DatiLavorazione va a 1)

I primi 3 casi di spegnimento di un asse sono comuni a tutti e quattro i blocchi funzione degli assi. Il quarto caso è tipico invece del solo asse della morsa, ed è stato aggiunto proprio per risolvere il problema della variabilità del set point della morsa spiegato in precedenza.

Le uscite del blocco funzione associato a un asse sono 4:

- la variabile di conteggio del contatore veloce, che ci indica il set point attuale dell'asse (la posizione attuale dell'asse). Nel caso della morsa la variabile è ConteggioChiud del blocco dati DatiEncApriChiud
- la variabile che ci permette di sapere se il set point impostato in modalità automatica è stato raggiunto o meno. Nel caso della morsa la variabile è PosAutoAttivoMorsa del blocco dati DatiLavorazione: quando questa variabile è 0 vuol dire che la morsa non ha ancora raggiunto il set point impostato. Queste variabili sono utilizzate nella macchina a stati che gestisce la lavorazione automatica per sapere quando i 4 assi hanno raggiunto il set point programmato.
- Abbiamo poi le due variabili PLC associate alle due direzioni di movimento del motore dell'asse. Nel caso della morsa %Q0.0 (ApriMano) e %Q01 (ChiudiMano).

Figura 6: Blocco funzione associato all’asse della morsa – parte 2



I blocchi funzione degli assi sono dotati di numerosi altri ingressi per la gestione manuale/automatica.

Quando l’ingresso SelAutMan vale 1 si attiva la movimentazione automatica dell’asse, che si muoverà sino a che non raggiunge il set point impostato. Viceversa, quando questo ingresso vale 0, l’asse può essere movimentato manualmente, tramite dei pulsanti posizionati su pannello operatore.

L’ingresso SelAutMan di ogni asse viene controllato da due variabili:

- SelAutoMan del blocco dati DatiHMIAssi, che corrisponde a uno switch nel pannello operatore
- AutoLav del blocco dati DatiLavorazione che viene gestita dalla macchina a stati per la lavorazione automatica dei pezzi

Questa doppia abilitazione è funzionale a permettere il controllo della movimentazione del braccio meccanico sia dal pannello operatore sia dalla macchina a stati.

Quando l'ingresso SelAutMan vale 1, il motore dell'asse viene azionato in modo da raggiungere il set point impostato nell'ingresso SetPoint. Questo ingresso è collegato a una variabile di tipo Int che nel caso della morsa è SetPointChiudi del blocco dati DatiHMIAssi. Questa variabile corrisponde a un campo valori presente nel pannello operatore, ed è quindi impostabile da pannello operatore. Questo è stato molto utile durante la fase di debug del sistema per trovare i set point associati alle varie posizioni di prelievo/immissione dei pezzi della macchina a stati.

Durante la lavorazione automatica questa variabile viene impostata dalla macchina a stati in base al set point da raggiungere.

Gli ingressi Tolleranza e TimerVerifica sono due ingressi di tipo Int e Time impostati a un valore fisso in fase di debug del sistema tramite due campi valori presenti nel pannello operatore. Nel caso della pressa le variabili associate a questi due ingressi sono Tolleranza e TimerVerifica entrambe presenti nel blocco dati DatiEncoderAprChiud. Lo scopo di queste variabili è quello di definire un intervallo di tolleranza per segnalare il raggiungimento del set point nel caso di funzionamento automatico. Il conteggio del HSC può essere infatti più veloce del ciclo programma, e il rischio che si corre a dichiarare il raggiungimento del set point quando il conteggio dell'encoder è esattamente uguale a quello del set point è che l'asse oscilli avanti indietro intorno al set point fissato. Per evitare queste possibili oscillazioni attorno al set point fissato, si fissa una tolleranza: il set point viene dichiarato raggiunto quando il conteggio del HSC è compreso tra il set point fissato + o - la tolleranza impostata per un tempo pari a TimerVerifica. In fase di debug quasi tutti gli assi consentono il raggiungimento preciso del set point senza oscillazioni con valori di tolleranza pari a 4 e un timer verifica di 500 ms.

L'ingresso ContMaxForw è un ingresso di tipo Int impostato a un valore fisso in fase di debug/configurazione del sistema e che corrisponde al valore massimo di conteggio dell'encoder corrispondente all'asse. Ogni asse ha infatti un fine corsa fisico che identifica la sua posizione di home, ma manca il fine corsa per i movimenti nell'altra direzione. Nel caso della morsa manca un fine corsa per la chiusura della mano, ed è rappresentato dalla variabile ContMaxChiud del blocco dati DatiEncoderAprChiud, che fissa il valore massimo di conteggio per la chiusura della mano. Quando il conteggio del HSC raggiunge il valore massimo impostato, sia in modalità manuale sia automatica, il segnale che attiva il motore di chiusura della mano viene disabilitato.

L'ultimo ingresso per il controllo automatico dell'asse è JogGoToSetPoint: il fronte di salita di questo segnale fa partire la movimentazione automatica dell'asse verso il set point impostato (se l'asse è in modalità automatica). Nel caso della morsa questo segnale è controllato da due variabili: JogGoToSetPointApr del blocco dati DatiHMIAssi e GoToSetPointLav del blocco dati DatiLavorazione. La prima variabile è associata a un pulsante nel pannello operatore e permette di avviare da pannello il posizionamento automatico dell'asse. La seconda è invece gestita dalla macchina a stati per la lavorazione automatica dei pezzi, per consentire alla macchina a stati di posizionare gli assi nei set point desiderati.

Ci sono infine i tre ingressi per la gestione manuale degli assi:

- JogGoToFCMan, per muovere l'asse verso il fine corsa/home

- JogGoForwMan, per muovere l'asse nella direzione opposta al fine corsa/home
- JogGoHome, per posizionare in modo automatico l'asse nella posizione del fine corsa/home.

Nel caso della mano questi tre ingressi corrispondono a tre variabili del blocco dati DatiHMIAssi: JogApriMan, JogChiudiMan e JogGoHomeApri. Queste tre variabili corrispondono a tre pulsanti del HMI.

Per evitare che durante la lavorazione automatica dei pezzi da parte della macchina a stati la pressione involontaria di questi pulsanti possa creare malfunzionamenti, questi segnali sono disabilitati quando la variabile AutoLav vale 1.

Tutti e 4 gli assi del braccio meccanico funzionano nello stesso modo descritto sopra. Ogni asse ha le sue variabili specifiche per effettuare il posizionamento manuale o automatico con la precisione richiesta.

L'unico asse gestito in modo particolare all'avvio del sistema è l'asse verticale: sul fronte di salita di SystemOn, ossia all'accensione del sistema, viene messo a 1 l'ingresso JogGoHome del blocco funzione dell'asse verticale per portare l'asse verticale al fine corsa alto. In questo modo il sistema parte posizionando il braccio in modo che non colpisca altre parti del sistema, anche in caso di spegnimento durante una fase intermedia della lavorazione automatica dei pezzi.

Per quanto riguarda il pannello HMI, contiene alcune pagine specifiche dedicate alla gestione degli assi meccanici.

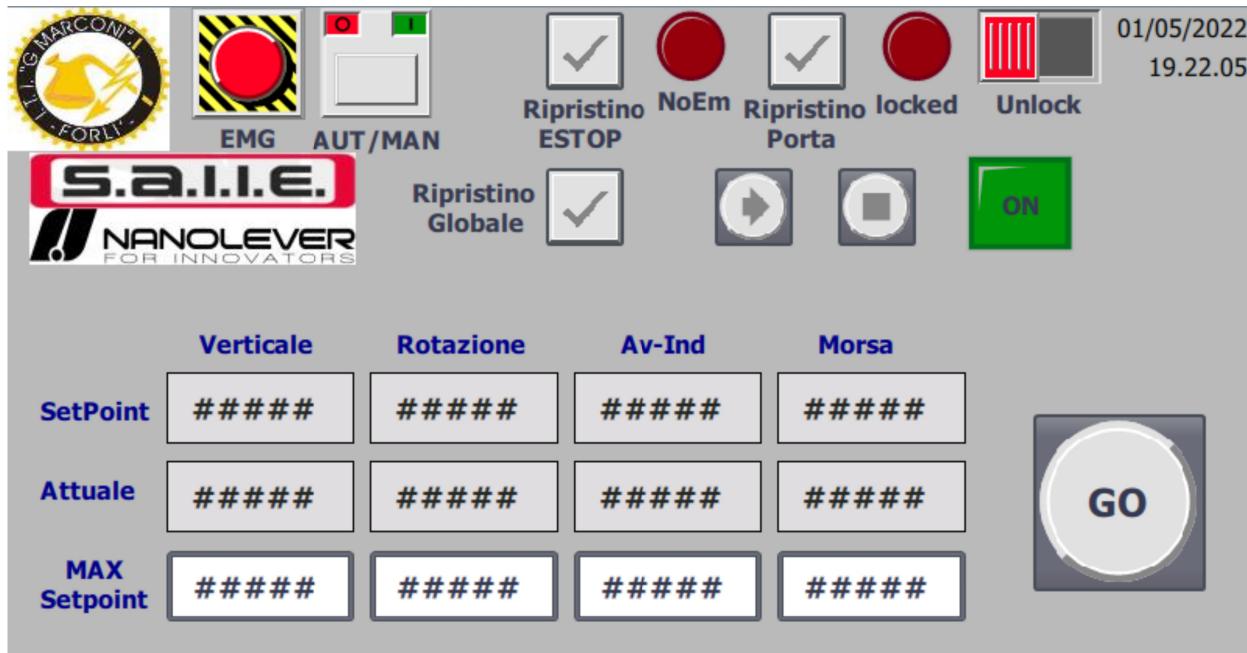
Figura 7: Switch JogEmergenza e SelAutoMan del pannello operatore



Gli switch associati alle variabili JogEmergenza e SelAutoMan del blocco dati DatiHMIAssi sono presenti in alto a sinistra in tutte le pagine HMI, perché sono stati inseriti nel modello.

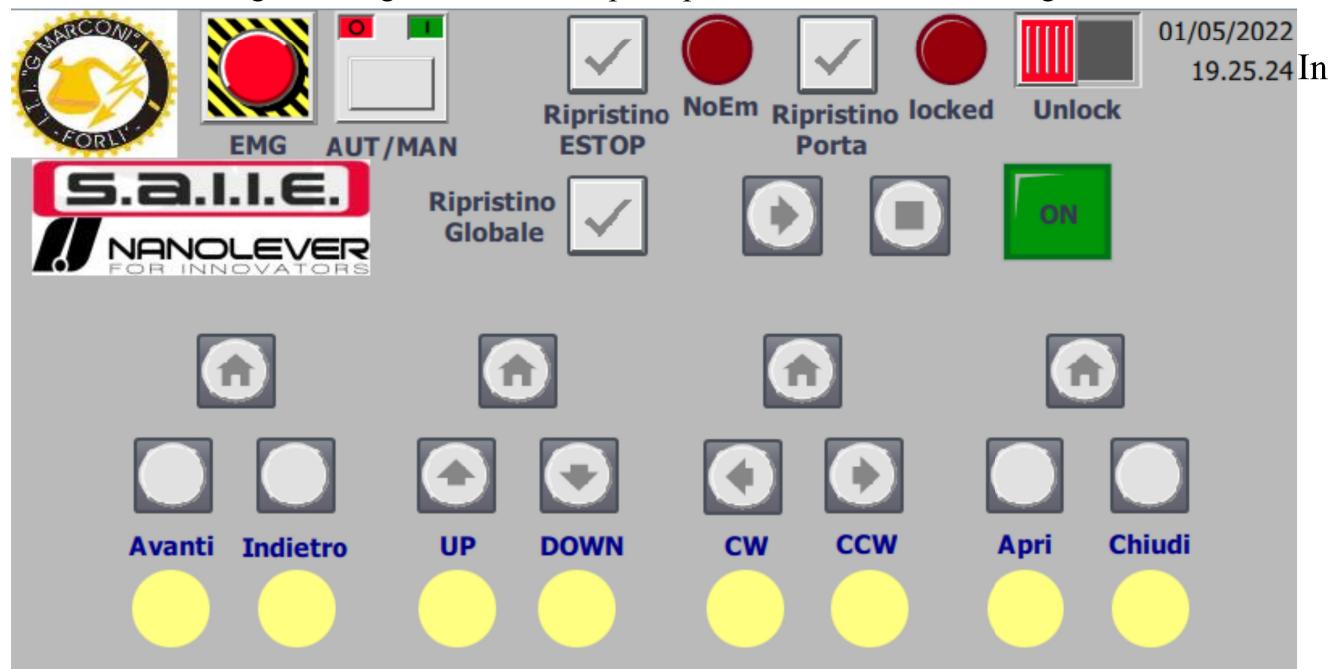
Vi sono poi due pagine HMI dedicate al posizionamento automatico e manuale degli assi: Auto e Man, rispettivamente.

Figura 8: Pagina HMI “Auto” per il posizionamento automatico degli assi



Per posizionare contemporaneamente tutti e 4 gli assi al set point desiderato, si inseriscono i valori dei 4 set point nella riga SetPoint, si preme l'interruttore AUT/MAN e poi il pulsante GO.

Figura 9: Pagina HMI “Man” per il posizionamento manuale degli assi



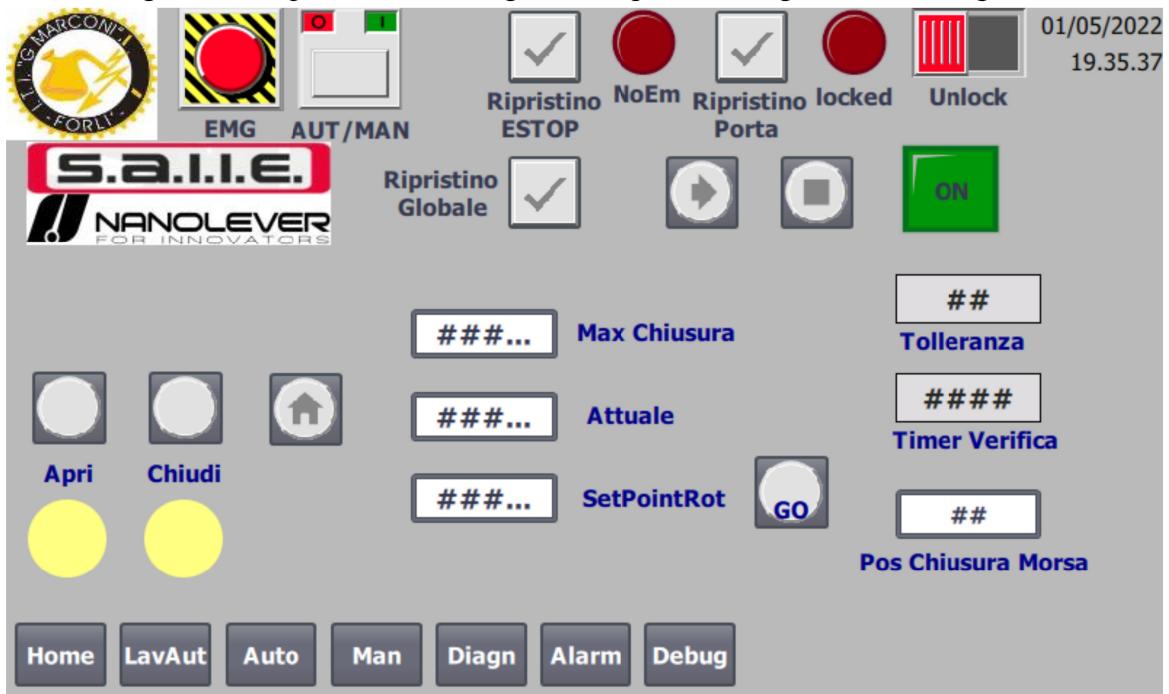
questa pagina ci sono i pulsanti per i comandi manuali dei 4 assi, compreso il pulsante di homing per posizionarli al fine corsa. I cerchi gialli mostrano lo stato del motore associato a ciascun asse: il cerchio diventa rosso quando il motore si muove in quella direzione.

Figura 10: Pagina HMI “Debug” per la configurazione/debug degli assi



Nelle prime fasi del progetto la configurazione e debug di ciascun asse ha richiesto molto tempo e molte prove. Per questo motivo ogni asse aveva una propria pagina HMI contenente entrambe le movimentazioni, automatica e manuale. Queste pagine sono state mantenute come sottomenu della pagina Debug, in modo che non siano visibili dalle pagine principali del HMI.

Figura 11: Pagina HMI “Debug/Morsa” per la configurazione/debug della morsa



Questa pagina, accessibile solo dalla pagina Debug, contiene i pulsanti e i campi valori per la movimentazione manuale e automatica della morsa. Rispetto alle altre pagine di

debug degli assi contiene un campo valore in più, Pos Chiusura Morsa, che corrisponde alla variabile PosChiusuraMorsa del blocco dati DatiLavorazione. Questa variabile contiene il set point corrispondente alla presa di un pezzo. È stata introdotta dopo aver osservato che questo set point cambiava a seconda dello stato meccanico della morsa.

Segmento 6: blocco funzione nastro

Il blocco funzione nastro è stato studiato durante il corso di Sistemi Automatici. Le variabili più significative per la gestione di questo blocco funzione si trovano nel blocco dati DatiNastro.

Tabella 5: blocco dati DatiNastro

DatiNastro						
	Nome	Tipo di dati	Valore di avvio	Ritenzione	Accessibile ...	Scriv...
Static						
■	RitardoInserzione	Time	T#15	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
■	RitardoPezzoLavorato	Time	T#55	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
■	InizializzaNastro	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
■	NastroInErrore	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
■	PezzoDisponibile	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- RitardoInserzione e RitardoPezzoLavorato gestiscono il primo il ritardo tra quando un pezzo è inserito in ingresso al sistema (fotocellula di ingresso attiva) e l'avvio del motore del nastro, il secondo tra quando un pezzo lavorato viene restituito al nastro (fotocellula di uscita attiva) e l'avvio del motore per trasportarlo indietro.
- InizializzaNastro viene usata per resettare lo stato del nastro all'accensione.
- NastroInErrore segnala che il nastro è in uno stato di errore perché aspetta un pezzo lavorato dal braccio (e quindi non accetta in ingresso nuovi pezzi) ma, allo stesso tempo, non è stata settata a 1 la variabile interna Inserito.
- PezzoDisponibile è la variabile che viene utilizzata dalla macchina a stati per avviare il processo di lavorazione automatico, e viene posta a 1 dal blocco funzione nastro quando un pezzo inserito ha raggiunto la fotocellula di uscita del nastro.

Il nastro è programmato in modo che se viene inserito un pezzo quando un altro è in lavorazione il nastro non lo processa e viene generato un errore/allarme (questa funzionalità è ancora da implementare).

In modo analogo, quando il braccio preleva un pezzo dal nastro, il nastro rimane in attesa sino a che il pezzo non viene nuovamente riportato in posizione dal braccio.

L'ingresso ResetN del blocco funzione è importante per inizializzare il nastro all'accensione e quando un pezzo viene scartato. Quando un pezzo viene scartato perché troppo leggero, il nastro non deve aspettare che il braccio riporti indietro il pezzo e deve invece ripartire con un nuovo pezzo in ingresso.

Per questo motivo l'ingresso ResetN viene attivato non solo dalla variabile InizializzaNastro, ma anche dalla macchina a stati per la lavorazione automatica: quando

la macchina a stati è alla fine del processo di scarto, nello stato ‘S’ si reinizializza il nastro.

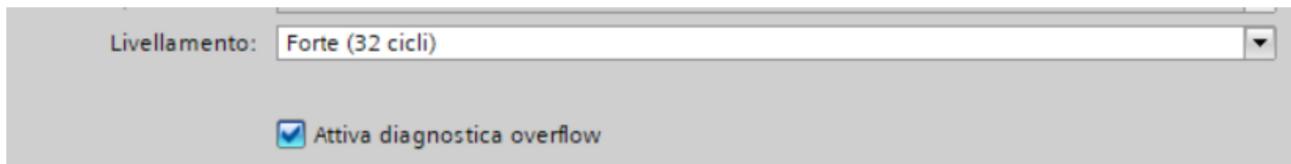
NastroOn è l’ingresso del blocco funzione collegato alla variabile SystemOn, e serve per abilitare il nastro solo quando il sistema è acceso.

Funzionalità da aggiungere: non c’è una pagina HMI specifica dedicata al nastro, in quanto per la sua semplicità non ha richiesto particolari procedure di calibrazione/debug. Nello sviluppo futuro del progetto è prevista però l’aggiunta di una pagina HMI dedicata al nastro, da utilizzare per il suo azionamento manuale e la configurazione dei suoi parametri/ritardi.

Segmento 7: blocco funzione Misura_Peso

Il blocco funzione relativo al peso è stato progettato a partire dai tutorial Siemens sul S7-1200 disponibili online [1]. L’uscita in tensione del circuito di condizionamento della cella di carico è collegato al primo ingresso analogico della CPU 1214FC, %IW64, corrispondente alla variabile PLC DatoPeso. Gli ingressi analogici di questa CPU sono configurabili solo in tensione, con range di ingresso 0:10V, cui corrisponde l’intervallo di valori digitali di 0:27648. Con il blocco funzione NORM_X l’intervallo 0:27648 viene convertito in un numero reale compreso tra 0 e 1 (normalizzazione), #OutNormX. Con il blocco SCALE_X il range 0:1 viene convertito nella scala di peso in grammi corrispondente al range di ingresso 0:10V, nel nostro caso fissata da 0 a 1030 grammi. Il prodotto della variabile #OutNormX moltiplicato per 10 fornisce la tensione presente all’ingresso analogico, utilizzata per la calibrazione del circuito di condizionamento della cella di carico.

La conversione analogico digitale, quella che permette di convertire la tensione 0:10V in un numero da 0 a 27648, è stata impostata da TIA Portal come media di 32 misure, in modo da eliminare gli effetti del rumore.



La calibrazione del circuito di condizionamento della cella di carico avviene come segue:

1. regolazione dell’offset: con la cella di carico a vuoto si abilita la misura del peso e si regola il trimmer relativo all’offset dell’amplificatore per strumentazione sino a che non si legge una tensione nulla sul HMI
2. regolazione del guadagno: si inserisce il peso campione da 20 grammi sulla cella di carico e si abilita la misura del peso. Si regola il trimmer del guadagno dell’amplificatore per strumentazione sino a che il peso rilevato non è pari a 20 grammi.

Dopo aver effettuato la calibrazione sfruttando la pagina HMI del peso, si sono effettuate varie prove aggiungendo e togliendo pezzetti di cartone sulla cella di carico, in modo da verificare la stabilità e precisione delle misure. Il sistema è in grado di rilevare variazioni di 1 grammo con un elevato grado di ripetibilità.

Nel processo di lavorazione automatico, i pezzi sono prima posti sulla cella di carico, poi si apre la morsa, e dopo 1 secondo dall'apertura si effettua la misura del peso. Anche in questo caso le misure hanno mostrato un elevato grado di ripetibilità e precisione di 1 grammo, riuscendo a discriminare senza errori i pezzi gialli da quelli neri da 19 grammi.

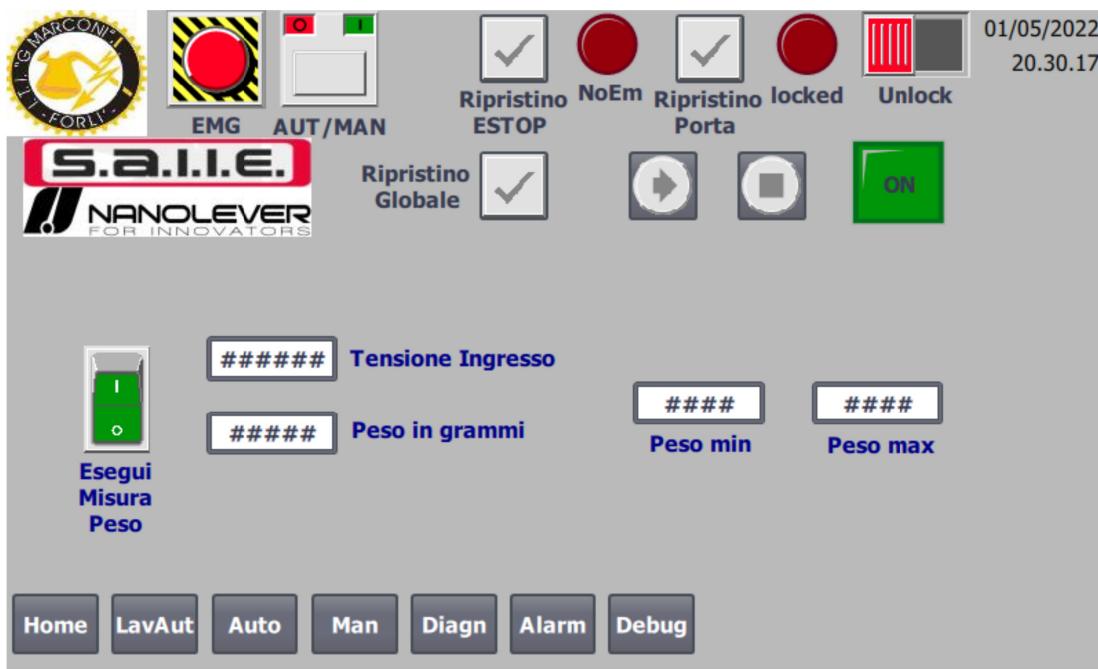
Non c'è un blocco dati specifico per il blocco funzione Misura_Peso, in quanto si fa riferimento a variabili del blocco dati DatiHMIAssi.

PesoMin0V e PesoMax10V sono i valori di peso minimo e massimo corrispondenti al range di misura 0:10V, fissati in fase di calibrazione e modificabili da pannello operatore.

La misura del peso è abilitata solo quando il segnale SystemOn vale 1 e solo quando il segnale EseguMisuraPeso vale 1. Questo segnale viene controllato sia da uno switch posto nel HMI sia dalla macchina a stati per la lavorazione automatica. Questo consente di utilizzare la cella di carico sia in modalità manuale tramite HMI sia in modalità automatica dalla macchina a stati.

Le uscite del blocco funzione sono TensioneV e PesoGr, entrambe di tipo Real, e visualizzate sul HMI. La prima serve per calibrare il circuito di condizionamento della cella di carico, la seconda viene usata dalla macchina a stati per discriminare i pezzi da scartare e quelli da pressare.

Figura 12: Pagina HMI “Debug/Peso” per la configurazione/debug della cella di carico



Questa pagina, accessibile solo dalla pagina Debug, contiene lo switch e i campi valori per la gestione manuale della cella di carico, ed è stata utilizzata per la calibrazione del circuito di condizionamento.

Segmenti 8 e 9: Comunicazione M2M, blocchi funzione PUT e GET

I blocchi funzione PUT e GET per la comunicazione M2M (Machine to Machine) tra PLC_1, master, e PLC_2, slave, vanno inseriti nel solo PLC master. La progettazione di

questi blocchi ha seguito passo passo le indicazioni del tutorial Siemens riguardante [3,4]:

- l'abilitazione di questo tipo di comunicazione in entrambi i PLC
- la creazione in dispositivi e reti del collegamento S7 tra i due PLC
- la creazione di blocchi dati non ottimizzati nel solo PLC Slave
- l'inserimento e configurazione dei blocchi PUT e GET nel solo PLC Master.

Oltre al tutorial online è stata consultata anche la guida del support online Siemens [5]. La comunicazione tramite i blocchi PUT e GET è in protocollo S7 ed è l'unica utilizzabile nel caso si disponga di uno switch di rete unmanaged, che non supporta il protocollo PROFINET. Nel caso si disponga di uno switch PROFINET (ad esempio di uno Scalance dal X-200 in su) è possibile utilizzare la comunicazione IO-Device, basata su protocollo PROFINET, in grado di offrire garanzie migliori sulle tempistiche di trasferimento dei segnali.

Nel blocco funzione PUT la CPU master trasferisce il contenuto della sua variabile SystemOn nella variabile bool all'indirizzo 0.0 del blocco dati DB1, ossia la variabile SystemOn del blocco dati ReceivedData della CPU slave. Questo segnale viene usato come abilitazione della pressa. Il trasferimento dati avviene a ogni fronte di salita o discesa del segnale SystemOn, ossia solo quando serve.

Nel blocco funzione GET, invece, la CPU master va a copiare il contenuto della variabile bool all'indirizzo 0.0 del blocco dati DB2 del PLC slave, che corrisponde alla variabile ProntoDaPrelevare del blocco dati SentData, nella variabile PrelevaDaPressa del blocco dati DatiLavorazione del PLC master. Questa variabile viene monitorata dalla macchina a stati che gestisce la lavorazione automatica: non appena va a 1 il pezzo viene preso e prelevato dalla fotocellula di uscita della pressa.

Il trasferimento associato al blocco funzione GET (dalla CPU slave alla master) avviene ogni secondo, perché utilizza il marker di clock Clock_1Hz (%M100.5).

Non ci sono pagine HMI dedicate alla comunicazione M2M e non sono state utilizzate le uscite dei blocchi funzioni PUT e GET per monitorare e segnalare lo stato della comunicazione: queste funzionalità saranno aggiunte in un secondo momento.

Segmenti 10 e 11: macchina a stati per la lavorazione automatica dei pezzi

La progettazione di macchine a stati è uno degli argomenti fondamentali del quarto anno del corso di studi di Sistemi Automatici, e viene affrontata prima in codice ladder, poi in codice strutturato. L'implementazione più semplice è quella in codice strutturato, e in questo progetto la si è realizzata in codice SCL.

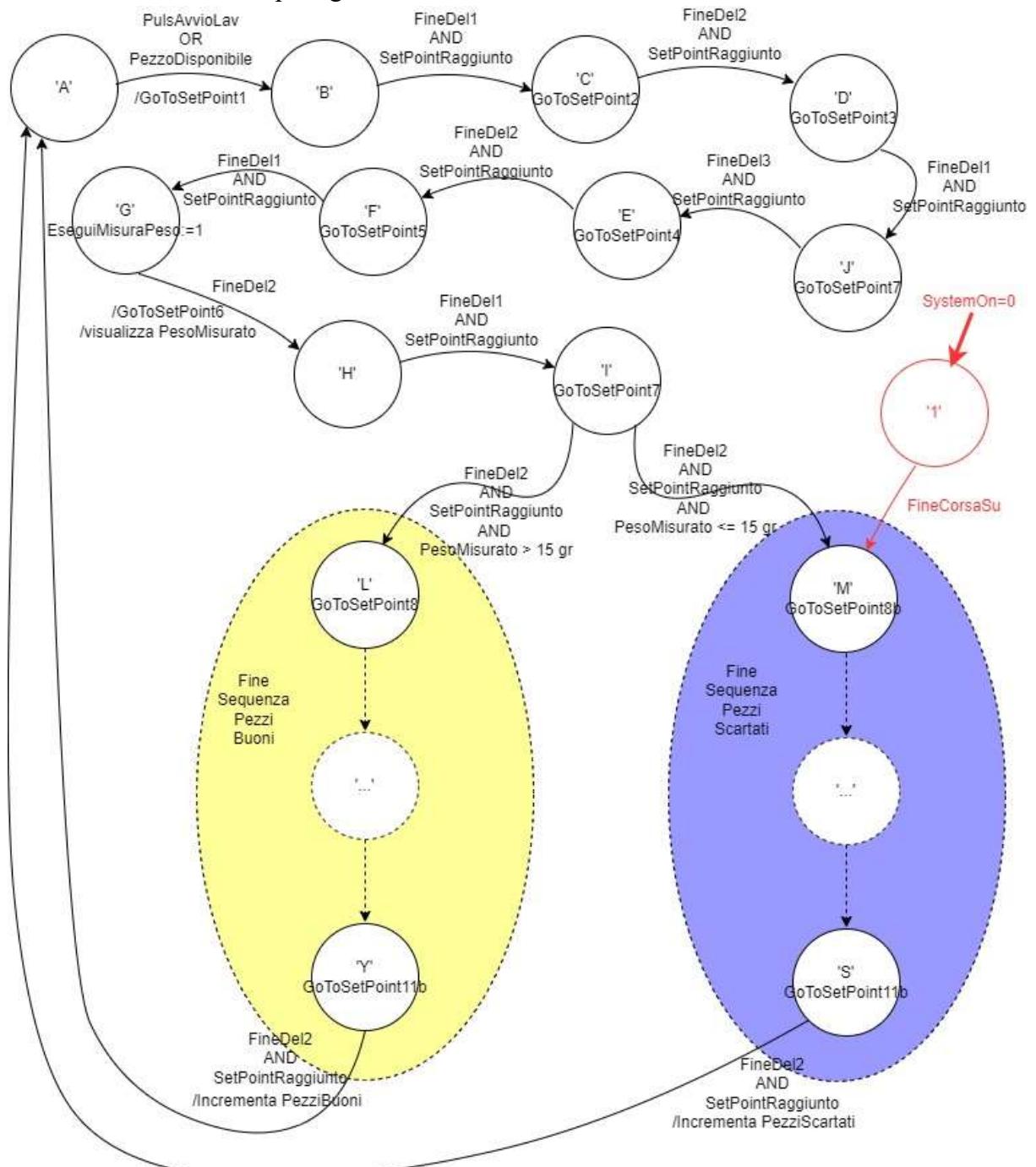
La macchina a stati consiste essenzialmente in una sequenza di stati in ognuno dei quali il braccio robotico raggiunge uno specifico set point.

Il segmento 11 è quello che gestisce il segnale set point raggiunto: solo quando tutti e 4 gli assi hanno raggiunto il set point si può passare allo stato successivo.

Per risolvere i problemi legati al diverso conteggio dell'encoder dell'asse avanti/indietro quando va in una direzione piuttosto che nell'altra, è stato aggiunto anche il fine corsa indietro come condizione di set point raggiunto: nella macchina a stati viene impostato un set point negativo (irraggiungibile) in modo che si è sicuri che il braccio torni indietro sino al fine corsa, che determinerà il set point raggiunto.

Per la mano abbiamo invece il problema che a volte non è in grado di raggiungere il set point di chiusura: è stato aggiunto per questo un temporizzatore da 2 secondi che interviene direttamente sul segnale emergenza del blocco funzione morsa in modo da interromperlo e forzare il raggiungimento del set point. La durata del temporizzatore è determinata dalla variabile TmorsaMax di tipo Time del blocco dati DatiLavorazione. All'interno della macchina a stati questa variabile viene impostata a 2 secondi solo quando la morsa è aperta. Una volta che la morsa si è chiusa questa variabile viene impostata a 10 ms, in modo da garantire che negli stati successivi non venga più azionato il motore per la chiusura della morsa, anche se il set point di chiusura non è stato raggiunto.

Figura 13: Pallogramma della macchina a stati per la gestione della lavorazione automatica dei pezzi. Il pallogramma è stato tradotto in codice SCL.



Una variabile Stato di tipo Char del blocco dati DatiLavorazione identifica i vari stati del sistema. Quando il sistema è spento Stato assume il valore ‘1’, in cui si attende l'accensione. All'accensione l'asse verticale viene portato al fine corsa alto (la variabile AppoggioFronteSOn si attiva sul fronte di salita della variabile SystemOn e attiva l'ingresso JogGoHome dell'asse verticale, portandolo al fine corsa su). Una volta che l'asse verticale ha raggiunto il fine corsa alto, la macchina a stati va nello stato ‘M’, in cui parte il processo di scarto di un pezzo. In questo modo se il processo di lavorazione automatico era stato interrotto a metà, non si rischia che un pezzo venga fatto cadere dall'alto dal braccio. Terminato il processo di scarto fittizio iniziale, il braccio si porta nella posizione di home (tutti gli assi sono al finecorsa), cui corrisponde lo stato ‘A’ e il set point 11b. Dallo stato ‘A’ parte il processo di lavorazione vero e proprio. Il processo di lavorazione parte in due casi:

- quando il segnale PezzoDisponibile del blocco dati DatiNastro diventa 1 (un pezzo è stato inserito nel nastro ed è arrivato alla seconda fotocellula)
- quando si preme il pulsante di avvio lavorazione presente nel HMI (in questo caso il pezzo deve essere stato posizionato manualmente nella seconda fotocellula del nastro). Il pulsante di avvio del HMI corrisponde alla variabile PulsAvvioLav del blocco dati DatiLavorazione.

Nello stato B si imposta il set point corrispondente al punto di prelievo del nastro e si avvia il posizionamento. Dopo 1 secondo si controlla se il set point è stato raggiunto. Il secondo di attesa prima di controllare il set-point è funzionale a controllare la variabile SetPointRaggiunto alcuni cicli macchina dopo l'avvio del posizionamento automatico.

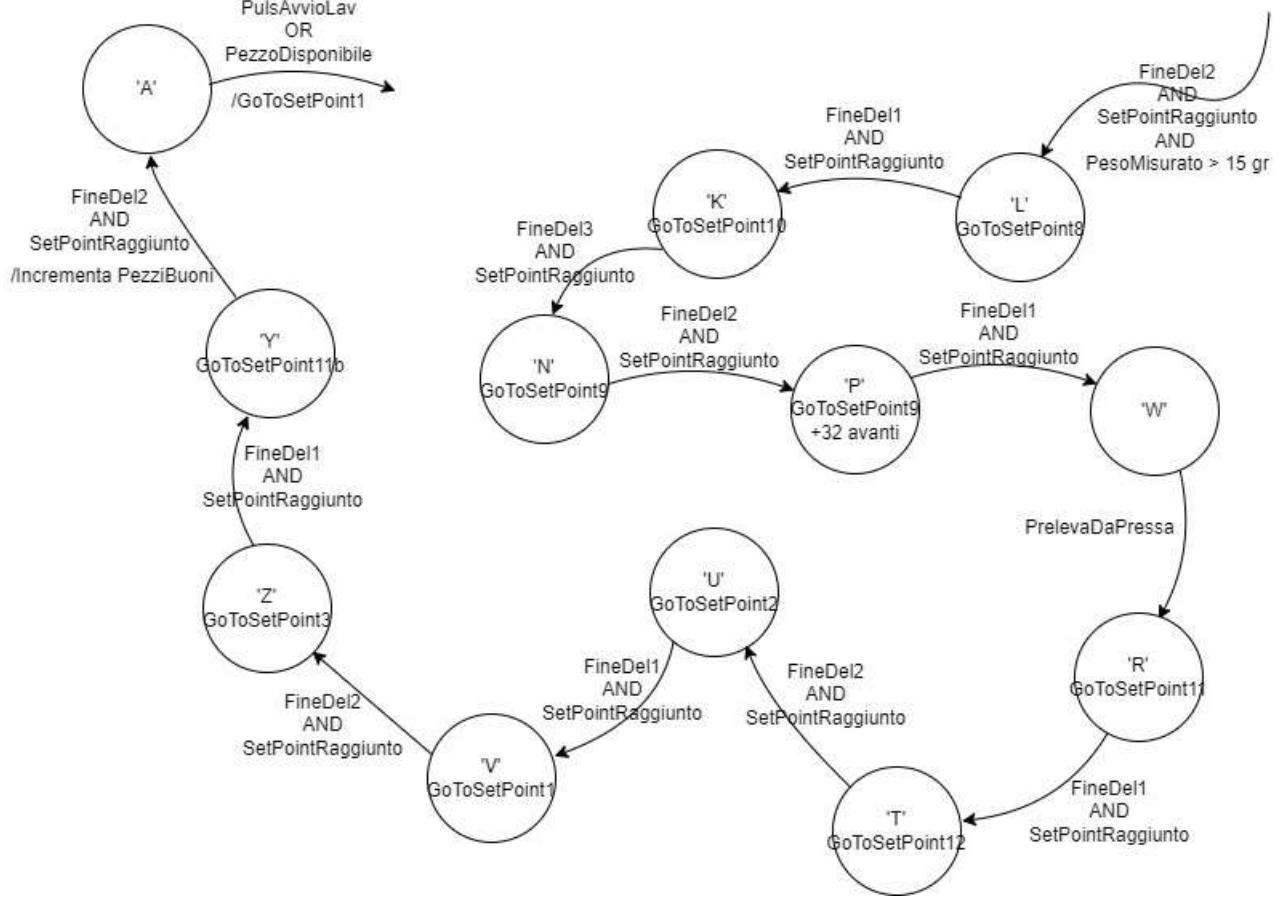
La macchina a stati utilizza 3 ritardi da 1 secondo dichiarati in SCL:

- "IEC_Timer_0_DB".TON
- "IEC_Timer_0_DB_1".TON
- "IEC_Timer_0_DB_2".TON

Questi tre temporizzatori forniscono tre segnali di uscita (FineDel1, FineDel2 e FineDel3) dichiarati nel blocco dati DatiLavorazione utilizzati per ritardare il controllo della variabile SetPointRaggiunto. Poiché stati successivi non possono fare riferimento allo stesso ritardo, FineDel1 e FineDel2 sono stati alternati. FineDel3 è stato aggiunto in un secondo momento quando è stato necessario introdurre alcuni stati intermedi per raffinare il processo di lavorazione e non si voleva modificare la sequenza di stati creata in precedenza.

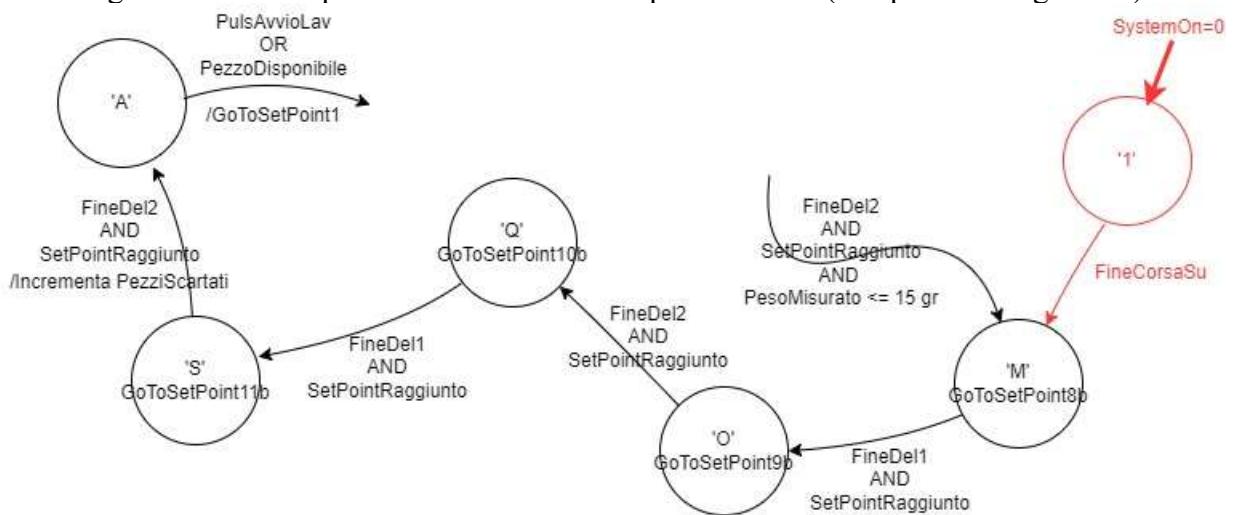
La macchina a stati non è solo una sequenza di set-point da raggiungere: nello stato ‘G’ si effettua la pesatura e dopo 1 secondo (FineDel2) si passa allo stato ‘H’. Dopo la pesatura, nello stato ‘I’, una volta raggiunto il set point, se il peso misurato è stato maggiore di 15 grammi si passa allo stato ‘L’, per portare il pezzo buono verso la pressa, altrimenti si passa allo stato ‘M’, per avviare il processo di scarto del pezzo.

Figura 14: Fine sequenza di lavorazione dei pezzi buoni (con peso > 15 grammi)



Nello stato 'W', invece, la macchina attende che il segnale PrelevaDaPressa diventi 1 e poi passa allo stato 'R'. Questo passaggio di stato è possibile solo se la comunicazione M2M funziona, altrimenti la macchina resterebbe nello stato 'W' in attesa che arrivi dalla CPU slave il segnale che il pezzo è pronto per essere prelevato dalla pressa.

Figura 15: Fine sequenza di lavorazione dei pezzi scartati (con peso <= 15 grammi)



Una volta terminata la sequenza degli stati per la lavorazione o lo scarto di un pezzo, il sistema incrementa il numero di pezzi lavorati o scartati e ritorna nello stato 'A', in

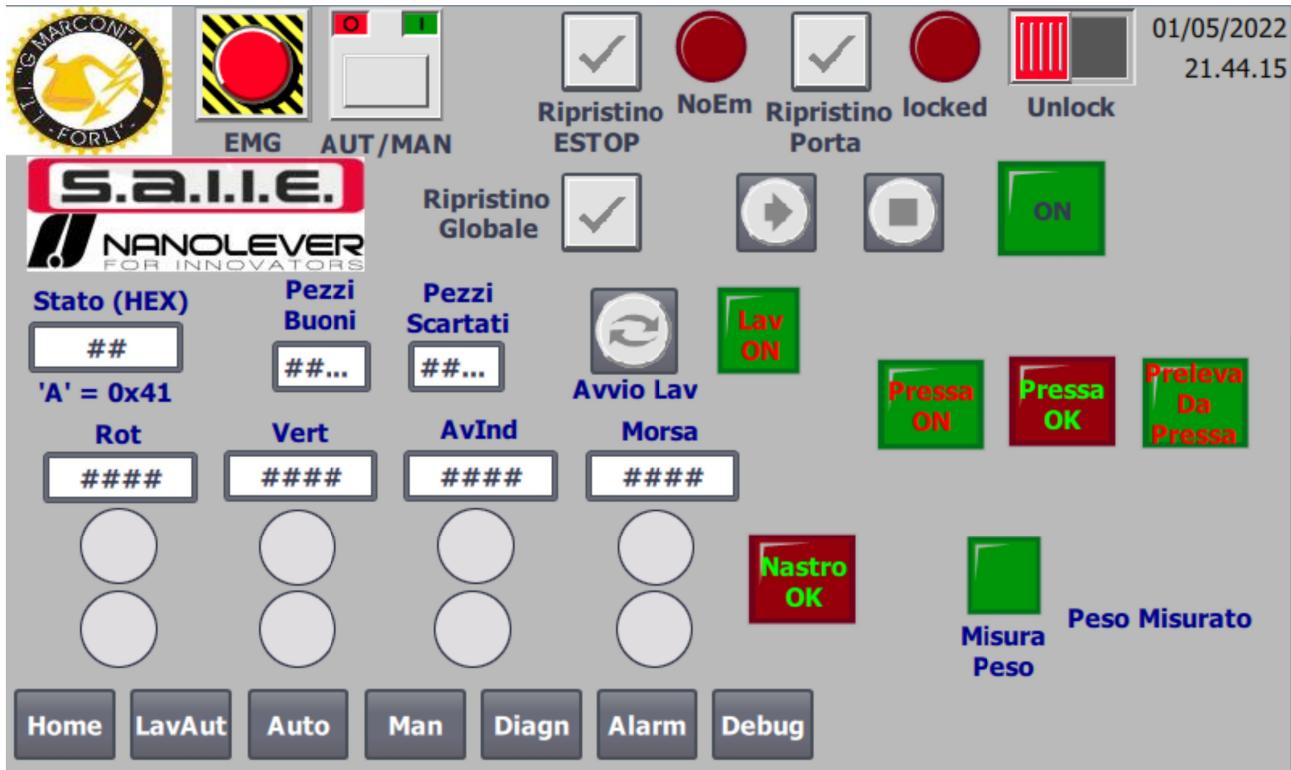
attesa che venga avviato un nuovo processo di lavorazione automatico. Le variabili che memorizzano il numero di pezzi sono le uniche di tutto il programma a essere dichiarate di tipo ritentivo, in modo che anche se viene interrotta l'alimentazione del PLC, alla sua successiva accensione ripartano dall'ultimo valore memorizzato. Se si vuole farle partire da zero è possibile modificarne il valore da pannello operatore, dove sono mappate nella pagina di lavorazione automatica.

Tabella 6: Blocco dati DatiLavorazione con le variabili utilizzate dalla macchina a stati

DatiLavorazione							
	Nome	Tipo di dati	Valore di avvio	Ritenzione	Accessibile ...	Scribi...	Visibile in ...
-	Static						
-	SetPoint1	Array[0..3] o... <input style="width: 20px; height: 20px; border: none; background-color: #f0f0f0;" type="button" value="..."/>		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint2	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint3	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint4	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint5	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint6	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint7	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint8	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint8b	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint9	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint9b	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint10	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint11	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint12	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint10b	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint11b	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPoint	Array[0..3] of Int		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	Delays	Array[0..10] of Time		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	Stato	Char	'A'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	PulsAvvioLav	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	LuceLavInCorso	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	GoToSetPointLav	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	AutoLav	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	SetPointRaggiunto	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	PosAutoAttivoVert	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	PosAutoAttivoRot	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	PosAutoAttivoAv	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
-	PosAutoAttivoMorsa	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Tra le variabili del blocco dati si osservano i set point della macchina a stati, dichiarati come Array di 4 interi, corrispondenti ai 4 set point di ogni asse. Abbiamo poi la variabile Stato di tipo Char e le variabili ritentive PezziBuoni e PezziScartati.

Figura 16: Pagina HMI “LavAut” relativa al processo di lavorazione automatico



Partendo da sinistra verso destra, procedendo per riga dall’alto verso il basso, la pagina HMI per la lavorazione automatica mostra lo stato del sistema (in formato esadecimale), il numero di pezzi buoni e scartati, il pulsante per avviare il processo di lavorazione automatico (il pezzo deve essere posizionato manualmente sulla fotocellula di uscita del nastro), la luce di segnalazione che indica si sta raggiungendo il set point. Nella seconda riga abbiamo la posizione dei vari assi, poi il segnale di accensione della pressa (utile per controllare se la comunicazione M2M funziona), quindi il segnale ProntoDaPrelevare della CPU slave (indicato con Pressa OK) e il corrispondente segnale PrelevaDaPressa della CPU master (utile per controllare se la comunicazione M2M funziona).

Nella terza riga abbiamo lo stato dei motori (si accendono i cerchi corrispondenti alla direzione di marcia di ciascun motore), abbiamo il segnale PezzoDisponibile (indicato con Nastro OK), e infine il segnale che abilita la misura del peso. Il peso viene visualizzato solo nella fase finale della lavorazione, dopo che l’operazione di pesatura si è conclusa.

Main_Safety_RTG1: il blocco funzione Safety

La progettazione della parte Safety si è basata sui tutorial online Siemens [6]. La CPU FailSafe 1214FC può gestire i dispositivi Safety perché è dotata di un modulo di espansione SM1226 F-DI8/16 per 16 ingressi safety 1oo1 (non ridondanti) o 8 ingressi safety 1oo2 (con ridondanza), e un modulo di espansione SM1226 F-DQ4 per 4 uscite safety. Questi moduli di espansione permettono di collegare i dispositivi safety e garantire la sicurezza dell’impianto anche durante la manutenzione.

La sicurezza di un impianto industriale è un obbligo di legge in base alla direttiva macchine 2006/42/CE, recepita in Italia prima dal testo unico sulla sicurezza D.Lgs. 81/2008 e poi, nello specifico, dal Decreto Legislativo 27 gennaio 2010, n. 17.

La direttiva prevede che vada effettuata una valutazione dei rischi associati all'utilizzo della macchina. Sulla base della valutazione dei rischi devono essere prese delle misure per minimizzarne la probabilità e raggiungere un livello di sicurezza adeguato alla commercializzazione della macchina nella Comunità Europea (marchiatura CE) [7].

Per questo la progettazione della parte safety di un macchinario è un elemento imprescindibile.

A seconda del macchinario e del suo utilizzo va raggiunto un certo livello di sicurezza, che può essere misurato secondo due standard, entrambi consentiti in Europa:

- SIL (Safety Integrity Level), con valori da 1 (**SIL 1** è il livello richiesto per macchinari poco rischiosi) a 3 (**SIL 3** è il livello di sicurezza richiesto per i macchinari più rischiosi). Questo standard è stato sviluppato dalla IEC (International Electrotechnical Commission) nella normativa EN 62061.
- PL (Performance Level), con valori da a (**PL a** è il livello richiesto per macchinari poco rischiosi) alla lettera e (**PL e** è il livello di sicurezza richiesto per i macchinari più rischiosi). Questo standard è stato sviluppato dalla ISO (International Standards Organization) nella normativa EN ISO 13849-1.

Nel nostro caso dobbiamo immaginare l'isola robotica circondata da una barriera meccanica che impedisca l'ingresso alle persone. Vi è un solo punto di immissione/prelievo dei pezzi, corrispondente all'ingresso del nastro trasportatore.

Per poter fare manutenzione si può entrare all'interno dell'isola robotica attraverso una porta, normalmente chiusa con un gate interlock switch.

Un gate interlock switch è una serratura che non può essere aperta se non viene messa nello stato di unlock. Per mettere la serratura nello stato di unlock ci sono due modi:

- manuale, attraverso una chiave specifica, che viene data su base nominale ai manutentori dell'impianto
- automatica, attraverso un'uscita safety del PLC

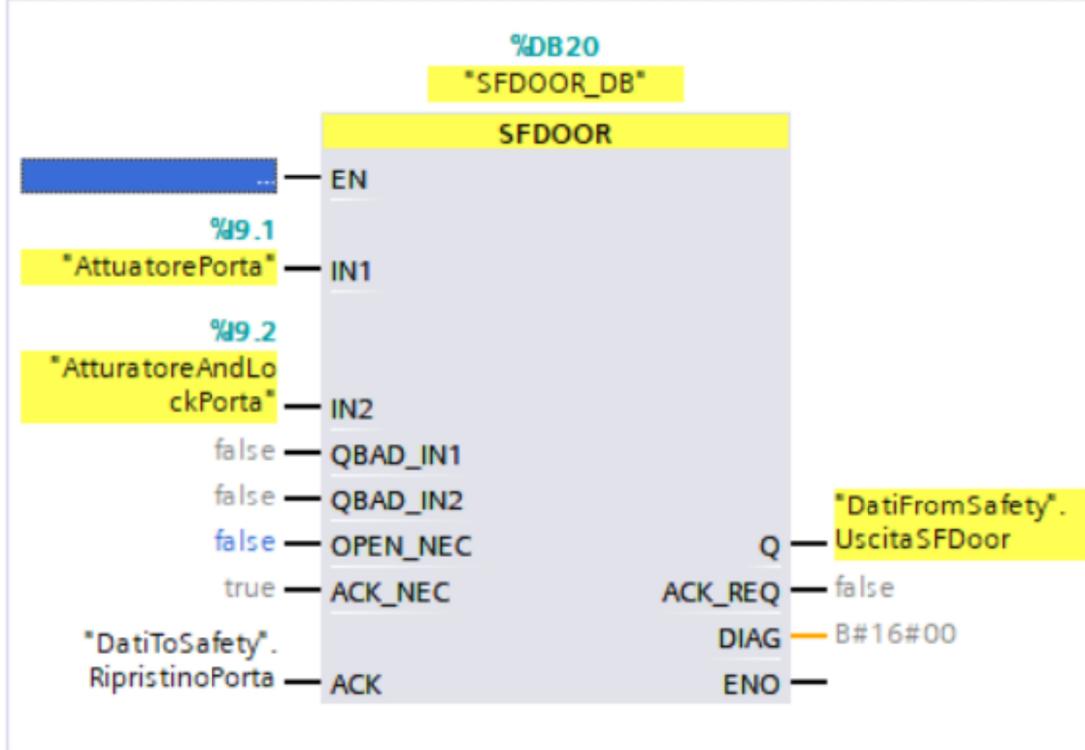
Non appena il gate interlock switch viene messo nello stato di unlock, qualunque sia stata la modalità, la funzione safety ad esso associata segnala la condizione di pericolo, disabilitando il sistema.

Come mostrato in figura 3, il gate interlock switch è dotato di due contatti NC collegati agli ingressi Safety %I9.1 e %I9.2 (variabili PLC AttuatorePorta e AttuatorAndLockporta) e alimentati internamente (entrambi da VS2). Gli ingressi %I9.1 e %I9.2 sono gestiti in modalità 1oo1, ossia separatamente l'uno dall'altro, tramite il blocco funzione safety SFDOOR.

Uno dei due contatti NC è, all'interno del dispositivo safety, posto in serie a un altro contatto NC il cui stato è funzione dello stato di lock/unlock della serratura: in questo modo non appena la serratura viene messa nello stato di unlock l'interruttore si apre e l'ingresso %I9.2 diventa 0, causando la segnalazione di pericolo da parte del blocco safety: la variabile UscitaSFDoor del blocco dati safety DatiFromSafety diventa 0.

Segmento 1: Gestione Gate Interlock Switch

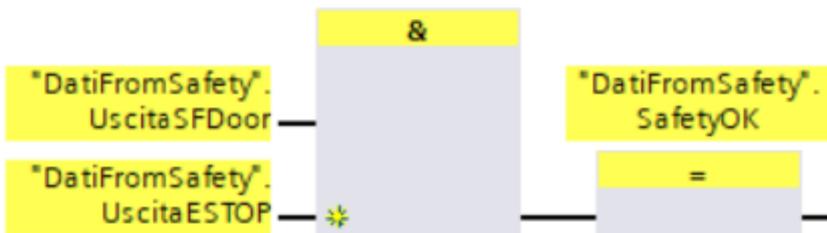
Gestione Gate Interlock Switch



Quando l'uscita del blocco SFDOOR diventa 0, anche l'uscita complessiva del blocco funzione safety diventa 0, segnalando la situazione di pericolo per l'intero sistema.

Segmento 5: Generazione del segnale di abilitazione del sistema

Generazione del segnale di abilitazione del sistema



La variabile SafetyOK del blocco dati safety DatiFromSafety è quella che resetta la variabile SystemOn dell'intero sistema. Finché SafetyOk non torna a 1, il sistema non può essere riavviato.

Per ripristinare la funzione safety SFDOOR si può utilizzare la variabile RipristinoPorta del blocco dati DatiToSafety. Quando gli ingressi del blocco SFDOOR sono ripristinati al loro valore di sicurezza (entrambi a 1, ossia porta chiusa e nello stato locked), l'uscita UscitaSFDoor rimane a 0. Per riportarla a 1 è necessario fornire l'acknowledge premendo il pulsante di ripristino presente nel pannello operatore.

Sperimentalmente abbiamo verificato che il gate interlock switch, una volta sbloccato, può essere ripristinato solo dopo che la serratura è stata prima aperta e poi richiusa. In

sostanza, una volta che il dispositivo è stato messo nello stato di unlock, è possibile ripristinarlo solo dopo che la serratura è stata prima aperta, poi richiusa e, infine, rimessa nello stato di lock. Solo dopo questa sequenza di operazioni la funzione safety SFDOOR può essere ripristinata tramite il pulsante disponibile nel pannello operatore.

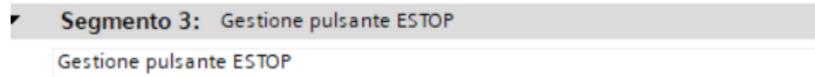
Lo sblocco automatico del gate interlock switch può avvenire tramite uno switch presente nel pannello operatore associato alla variabile IntUnlock del blocco dati DatiToSafety. Quando questa variabile viene posta a 1, l'uscita Safety %Q17.2 (variabile PLC UnlockPorta) si attiva, eccitando la bobina che controlla il contatto NC, aprendolo. Il contatto NC viene richiuso se la variabile IntUnlock ritorna a 0, ma questo non basta a permettere il ripristino del sistema se prima la serratura non è stata aperta e poi richiusa. Il blocco funzione safety gestisce anche un fungo di emergenza: questo è un dispositivo



safety obbligatorio in ogni macchinario che presenti parti in movimento. Il suo intervento causa l'interruzione dell'alimentazione del sistema e la sicurezza che tutti i drive (motori del sistema) si fermino immediatamente.

Come mostrato in figura 3, il fungo di emergenza (ESTOP) ha due contatti NC che sono collegati agli ingressi Safety %I8.0 e %I9.0 e alimentati da VS1 e VS2 rispettivamente (alimentazione interna). Questo permette di abilitare il controllo di corto-circuito sui contatti stessi. I due ingressi %I8.0 e %I9.0 sono gestiti in modalità 1oo2, ossia come un unico contatto %I8.0 ridondato (cui è associata un'unica variabile PLC PESTOP): questo permette di gestire il pulsante di emergenza con il livello massimo di sicurezza SIL 3. Quando il fungo viene premuto entrambi gli ingressi vanno a 0 (sia %I8.0 sia %I9.0) e l'uscita del blocco funzione ESTOP1, ossia il segnale UscitaESTOP del blocco dati safety DatiFromSafety diventa 0, disabilitando tutto il sistema.

A differenza della funzione SFDOOR, il ripristino della funzione ESTOP1 richiede due pulsanti HMI: uno per il ripristino globale della safety tramite il blocco funzione ACK_GL (variabile RipristinoGlobale del blocco dati DatiTosafety), l'altro specifico per il ripristino del blocco funzione ESTOP1 (variabile RipristinoESTOP del blocco dati DatiTosafety).



Nel nostro sistema il fungo di emergenza non interrompe l'alimentazione ma si limita ad agire sulla variabile di accensione SystemOn dell'intero sistema.

Ci si riserva in futuro di aggiungere al sistema un contattore per tagliare l'alimentazione dei drive quando interviene il fungo di emergenza, e monitorare il segnale di feedback del contattore per verificare che l'alimentazione sia effettivamente interrotta, come avviene nei sistemi automatici reali.

Sempre in futuro si aggiungerà al sistema una password per tutti i pulsanti/switch legati al ripristino e gestione dei dispositivi safety, in modo che solo il personale autorizzato possa ripristinare il sistema nel caso sia intervenuta la parte safety.

Figura 17: Pulsanti, switch e luci di segnalazione per la gestione della parte safety



I pulsanti, switch e le luci di segnalazione per la gestione della parte safety sono stati inseriti nella pagina modello, in modo che fossero presenti un tutte le pagine del HMI. All'accensione del sistema le uscite dei blocchi safety sono entrambe a 0, ed è necessario il semplice ripristino di ognuna di esse premendo i pulsanti Ripristino ESTOP e Ripristino Porta. A questo punto le uscite dei due blocchi safety diventano 1 e le luci

NoEm e locked si accendono di colore verde, segnalando che il sistema può essere acceso. L'interruttore di unlock serve per sbloccare/bloccare la serratura da pannello operatore. Per garantire una maggiore sicurezza dell'impianto conviene attivare questi oggetti solo a seguito di autenticazione tramite password, in modo che solo i manutentori esperti siano in grado di ripristinare il sistema in caso di intervento della safety.

Descrizione variabili e segnali del Main del PLC_2

Segmento 1: blocco funzione pressa

Il blocco funzione pressa è stato studiato e progettato durante il corso di Sistemi Automatici. Oltre agli ingressi e uscite fisiche collegate agli ingressi e uscite del PLC_2 (fotocellule, finecorsa e segnali di attivazione dei motori) sono stati definiti tre blocchi dati: DatiPressa, che contiene tre variabili di tipo Time:

Tabella 7: Blocco dati DatiPressa

DatiPressa					
	Nome	Tipo di dati	Valore di avvio	Ritenzione	A
Static					
	TempoAttesaPezzoInserito	Time	T#3S		
	TempoPressata	Time	T#2S		
	TempoAttesaRitornoPezzo	Time	T#400MS		

TempoAttesaPezzoInserito è il tempo che passa da quando il braccio deposita il pezzo sulla fotocellula di ingresso della pressa a quando si avvia il motore del nastro trasportatore della pressa per iniziare la lavorazione del pezzo. TempoPressata è invece la durata temporale durante cui la pressa viene lasciata al fine corsa basso prima di essere rialzata. TempoAttesaRitornoPezzo è invece l'intervallo di tempo che passa da quando la pressa viene rialzata a quando viene attivato il motore del nastro trasportatore che deve riportare indietro il pezzo pressato.

Questi tempi non sono stati parametrizzati perché il blocco funzione pressa è stato preso dal corso di Sistemi Automatici senza modifiche.

Le uniche modifiche apportate sono state quelle necessarie per abilitare la comunicazione M2M tra il PLC_2 che gestisce la pressa, quello slave, e PLC_1, che funge da master.

Come mostrato nella tabella 3, le variabili che vengono lette dal PLC master tramite le funzioni PUT sono state dichiarate nel blocco dati non ottimizzato ReceivedData, che contiene SystemOn, utilizzata per l'abilitazione della pressa, e MotorOn, al momento non utilizzata ma che si prevede di utilizzare in futuro per la gestione delle luci di segnalazione una volta che sarà disponibile un modulo di espansione con altre uscite digitali.

Come mostrato nella tabella 4, le variabili che vengono inviate al PLC master tramite la funzione GET sono state dichiarate nel blocco dati SentData, che contiene ProntoDaPrelevare, utilizzato dalla macchina a stati per prelevare i pezzi dalla pressa

una volta che il PLC_2 ha terminato il processo di lavorazione, e Dato2, al momento non utilizzata e riservata per usi futuri.

Questi blocchi dati sono di tipo non ottimizzato in modo da poter indirizzare le variabili del PLC_2 che ci interessa trasferire tramite i loro indirizzi assoluti.

Il PLC_1 master, infatti, può indirizzare le variabili dei blocchi dati di un altro PLC slave solo tramite i loro indirizzi assoluti.

Funzionalità da aggiungere: non c'è una pagina HMI specifica dedicata alla pressa, in quanto per la sua semplicità non ha richiesto particolari procedure di calibrazione/debug. Nello sviluppo futuro del progetto è prevista però l'aggiunta di una pagina HMI dedicata alla pressa, da utilizzare per il suo azionamento manuale e la configurazione dei suoi parametri/ritardi.

Sitografia

- [1] <https://new.siemens.com/it/it/azienda/webinar/on-demand/programma-scuole/percorso-formativo-didattico-dedicato-al-s7-1200.html>
- [2] <https://support.industry.siemens.com/cs/mdm/109478121?c=72565309451&lc=it-WW>
- [3] <https://assets.new.siemens.com/siemens/assets/api/uuid:933710ab-4f3e-4ade-8b15-c75052227f40/20210222-comunicazionetraplceretivirtuali1-parte01.mp4>
- [4] <https://assets.new.siemens.com/siemens/assets/api/uuid:11143296-0c94-4e83-81b2-798a6bda318e/20210222-comunicazionetraplceretivirtuali1-parte02.mp4>
- [5] https://support.industry.siemens.com/cs/attachments/82212115/82212115_S7_communication_S7-1500_en.pdf
- [6] <https://assets.new.siemens.com/siemens/assets/api/uuid:ed3fd01b-30b2-4f60-8165-55bb001a38da/2801-percorso-demo-safety-1parte.mp4>
- [7] <https://www.youtube.com/watch?v=GnXv52fXxX8>
- [8] <https://www.analog.com/media/en/training-seminars/design-handbooks/designers-guide-instrument-amps-complete.pdf>

Ringraziamenti

Si ringrazia la ditta SAIIE (www.saiie.it) per averci fornito il pannello operatore e i dispositivi safety: il fungo di emergenza e il gate interlock switch.

Ringraziamo inoltre l'ing. Michele Comandini di SAIIE, ex diplomato di Elettronica nel 2017, per aver dedicato parte del suo tempo ad aiutarci nella programmazione del gate interlock switch.

Si ringrazia inoltre la ditta Nanolever (www.nanolever.com) per averci fornito la cella di carico a estensimetri.