

**Università Politecnica Delle Marche**

**FACOLTÀ DI INGEGNERIA ELETTRONICA**



**UNIVERSITÀ  
POLITECNICA  
DELLE MARCHE**

**DII Dipartimento Ingegneria Informazione**

---

**RELAZIONE LABORATORIO**

**Automazione Industriale**

---

**Studente**

	<b>Matricola</b>
Valerio Aquilano	1103267
Marcantonio Luca	1106689
Owabor Vivien	1103872
Maksymilian Pasierb	1114697

**Matricola**

**Docente**

Silvia Maria Zanoli





*Automazione industriale,  
disciplina che studia le metodologie  
e tecnologie che permettono  
il controllo dei flussi  
di energia, di materiali e di informazioni  
necessari alla realizzazione  
dei processi produttivi*



# Indice

<b>1 Relazione esperienza di laboratorio</b>	<b>1</b>
1.1 Obiettivo . . . . .	1
<b>2 Introduzione PLC</b>	<b>1</b>
2.1 Ingressi analogici e digitali . . . . .	2
2.2 Dispositivi virtuali . . . . .	2
2.2.1 Timer - TON . . . . .	2
2.2.2 Timer - TOF . . . . .	3
2.2.3 Timer - TP . . . . .	4
2.2.4 Counter - CTU . . . . .	5
2.2.5 Counter - CTD . . . . .	5
2.2.6 Counter - CTUD . . . . .	6
<b>3 Dispositivi utilizzati nella prova</b>	<b>7</b>
<b>4 Descrizione del processo e dell'impianto</b>	<b>8</b>
4.1 Caricamento . . . . .	8
4.2 Trasporto . . . . .	9
4.3 Lavorazione . . . . .	9
4.4 Dettaglio funzionamento . . . . .	10
<b>5 Tabelle collegamenti interfaccia</b>	<b>11</b>
5.1 Assegnazione indirizzi ingressi . . . . .	11
5.1.1 Associazione input channel . . . . .	12
5.2 Assegnazione indirizzi uscite . . . . .	13
5.2.1 Associazione output channel . . . . .	13
<b>6 Legenda qualificatori - Standard IEC 1131-3</b>	<b>14</b>
<b>7 Programma in Sequential Functional Chart (SFC)</b>	<b>15</b>
7.1 Possibile alternativa tastatore . . . . .	19
<b>8 Programma in Ladder (LD)</b>	<b>21</b>
<b>9 Pannello operatore virtuale</b>	<b>24</b>
9.1 Funzionamento normale . . . . .	24
9.2 Magazzino B4 caricato . . . . .	25
9.3 Baia occupata . . . . .	25
9.4 Controllo campione . . . . .	26
9.5 Controllo altezza . . . . .	26
<b>10 Conclusioni</b>	<b>27</b>



# 1 Relazione esperienza di laboratorio

## 1.1 Obiettivo

Automatizzare una linea di produzione mediante l'ausilio di un controllore a logica programmabile.

## 2 Introduzione PLC

Il Controllore a Logica Programmabile, tradotto dall'inglese **Programmable Logic Controller (PLC)**, è un dispositivo progettato per monitorare e regolare macchine e processi di produzione in ambienti industriali che utilizzano tecnologie di automazione. Esso si è imposto come standard per il controllo logico sequenziale grazie alle sue caratteristiche di affidabilità, espandibilità, semplicità di programmazione, flessibilità, riciclo e costo. Il suo funzionamento è regolato da un'unità centrale (**CPU**), la quale acquisisce i segnali provenienti dall'esterno (**input**), li elabora e determina le operazioni che gli attuatori (**output**) dovranno compiere. L'architettura di un PLC si può riassumere in quanto segue:

- Il **modulo alimentatore**, che provvede a fornire ai circuiti interni le giuste correnti elettriche, mantenendo condizioni di isolamento e di sicurezza.
- Il **modulo processore** (CPU), il quale gestisce le porte ed esegue il programma.
- La memoria **RAM**, volatile, per il salvataggio dei dati in maniera non permanente.
- La memoria **ROM**, non volatile, per l'archiviazione del programma e di tutti i dati permanenti.
- I moduli di ingresso/uscita (**I/O**), che permettono al PLC di interagire con il circuito da controllare.
- L'armadio (o **rack**), contenente i moduli aggiuntivi connessi elettricamente e meccanicamente al PLC.

Generalmente il controllore può trovarsi in due **modalità operative**:

- Esecuzione (**RUN MODE**): si eseguono i programmi utente aggiornando ingressi e uscite;
- Programmazione (**CONFIG MODE**): si può modificare il programma e caricarlo sul dispositivo.

La programmazione del componente avviene tramite software dedicato installato su di un computer; per il caricamento del programma, invece, si effettua un collegamento tramite cavo ethernet tra il computer e il controllore.

Per quanto riguarda i linguaggi di programmazione ne esistono sia di tipo letterale sia di tipo grafico, e vengono regolati dalla norma IEC 1131-3.

Nella prova sono state utilizzati tre tipologie:

- Il Ladder (**LD**), un linguaggio grafico che si pone l'obiettivo di riprodurre in un paradigma di programmazione la terminologia, la simbologia ed il funzionamento di una rete elettrica.
- Il Sequential Functional Chart (**SFC**), un linguaggio grafico di più alto livello, improntato a sottolineare la sequenzialità del processo globale, scomponendolo in "passi" o "fasi".
- Lo Structured Text (**ST**), consistente in una serie di istruzione simili alla sintassi di un linguaggio ad alto livello per la programmazione. Vengono implementate scelte come ("IF..THEN..ELSE"), oppure loops come (WHILE..DO).

## 2.1 Ingressi analogici e digitali

In generale, si possono avere due tipi di input nel controllore: segnali di tipo analogico e segnali di tipo digitale.

Gli ingressi analogici rappresentano grandezze continue, come temperatura, pressione, livello di fluidi, velocità; offrono, dunque, una maggiore precisione e risoluzione. Necessitano, però, di essere convertiti in formato digitale, attraverso un convertitore analogico-digitale, in virtù del funzionamento su base bit del calcolatore.

Per quanto riguarda gli ingressi digitali, invece, non necessitano di essere convertiti, in quanto segnali binari che possono assumere due stati: "ON" o "OFF", "1" o "0", "aperto" o "chiuso". Sono comunemente utilizzati per rilevare la presenza o l'assenza di un oggetto, per monitorare interruttori, finecorsa, pulsanti, fotocellule, e così via; vengono interpretati dal PLC come livelli logici.

## 2.2 Dispositivi virtuali

Uno dei tanti vantaggi del PLC, sono i timer digitali, i contatori e le variabili virtuali di appoggio che si possono creare. Infatti essi non sono associati a dispositivi fisici realmente esistenti, ma semplicemente vanno ad emularne il funzionamento a livello software.

### 2.2.1 Timer - TON

Questo blocco funzione è utilizzato per attivare una risposta in uscita dopo un certo intervallo di tempo. Il temporizzatore si presenta graficamente come in figura 1.

Gli ingressi sono:

- IN : BOOL (avvia il conteggio rilevando un fronte di salita, lo resetta rilevando un fronte di discesa);
- PT : TIME (tempo trascorso prima che Q vada in set);

Le uscite sono:

- Q : BOOL (è allo stato alto PT secondi dopo che IN ha rilevato un fronte di salita);

- ET : TIME (tempo trascorso);



Figura 1: TON

Se IN è FALSO, Q è FALSO ed ET è 0. Non appena IN diventa VERO, il tempo inizierà a essere contato in millisecondi in ET finché il suo valore non sarà uguale a PT. In seguito rimarrà costante. Q è VERO quando IN è VERO ed ET è uguale a PT. Altrimenti è FALSO. Pertanto, Q ha un fronte di salita quando il tempo indicato in PT in millisecondi è terminato. L'andamento temporale è mostrato in figura 2.

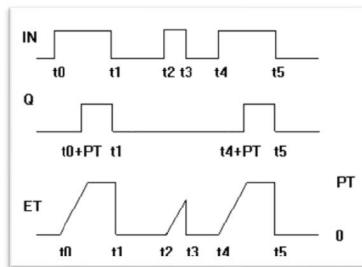


Figura 2: Andamento temporale - TON

### 2.2.2 Timer - TOF

Questo blocco funzione è usato per disattivare la risposta in uscita dopo un dato intervallo di tempo. Il temporizzatore si presenta graficamente come in figura 3.

Gli ingressi sono:

- IN : BOOL (avvia il conteggio rilevando un fronte di discesa, lo resetta rilevando un fronte di salita);
- PT : TIME (tempo trascorso prima che Q vada in set);

Le uscite sono:

- Q : BOOL (è allo stato alto PT secondi dopo che IN ha rilevato un fronte di discesa);
- ET : TIME (tempo trascorso);

Quando IN è VERO, Q è VERO ed ET è 0. Non appena IN diventa FALSO, il tempo inizia a essere contato in millisecondi in ET fino a quando il suo valore è uguale a quello di PT. In seguito rimarrà costante. Q è FALSO quando IN è FALSO ed ET è uguale a PT. Altrimenti è VERO. Pertanto, Q ha un fronte di discesa quando il tempo indicato in PT in millisecondi è terminato. L'andamento temporale è mostrato in figura 4.

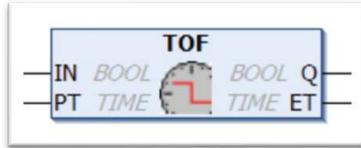


Figura 3: TOF

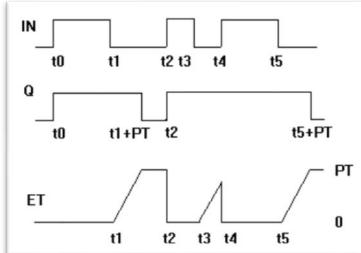


Figura 4: Andamento temporale - TOF

### 2.2.3 Timer - TP

Questo blocco funzione è utilizzato per generare impulsi di durata definita. Il temporizzatore si presenta graficamente come in figura 5.

Gli ingressi sono:

- IN : BOOL (innesco per l'inizio del segnale);
- PT : TIME (durata, in ms, del segnale in uscita allo stato alto);

Le uscite sono:

- Q : BOOL (impulso in uscita);
- ET : TIME (stato del segnale alto);

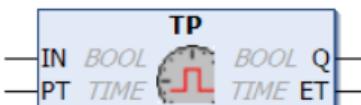


Figura 5: TP

Se IN è FALSO, le uscite sono FALSE o 0. Non appena IN diventa VERO, anche Q diventa VERO e resta VERO per la durata dell'impulso PT. Finché Q è VERO, il tempo viene incrementato in millisecondi in ET, finché il valore non raggiunge PT. Il valore rimane quindi costante. L'uscita Q rimane VERA fino allo scadere della durata dell'impulso, indipendentemente dallo stato dell'ingresso IN. Q fornisce quindi un segnale per l'intervallo specificato in PT. L'andamento temporale è mostrato in figura 6.

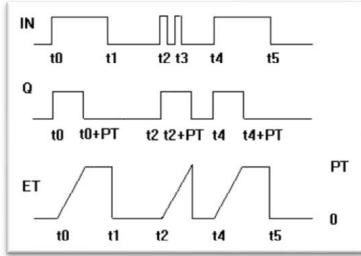


Figura 6: Andamento temporale - TP

#### 2.2.4 Counter - CTU

Il contatore si presenta graficamente come in figura 7.

Gli ingressi sono:

- CU : BOOL (conteggio ascendente);
- RESET : BOOL (ripristino del contatore a zero);
- PV : WORD (limite del contatore);

Le uscite sono:

- Q : BOOL (uscita del contatore);
- CV : WORD (valore corrente del conteggio);

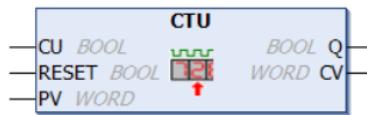


Figura 7: CTU

La variabile contatore CV viene inizializzata con 0 se RESET è VERO. Se CU ha un fronte di salita da FALSE a TRUE, il blocco funzione CV verrà incrementato di 1, a condizione che CV sia inferiore a PV (cioè che non causi un overflow). Q restituirà VERO quando CV è maggiore o uguale al limite superiore PV.

#### 2.2.5 Counter - CTD

Il contatore si presenta graficamente come in figura 8.

Gli ingressi sono:

- CD : BOOL (conteggio discendente);
- LOAD : BOOL (ripristino del valore iniziale);
- PV : WORD (valore iniziale del contatore);

Le uscite sono:

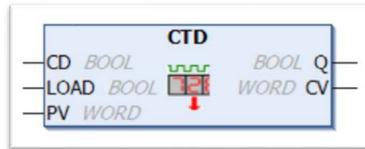


Figura 8: CTD

- Q : BOOL (uscita del contatore);
- CV : WORD (valore corrente del conteggio);

Quando LOAD è VERO, la variabile del contatore CV viene inizializzata con il limite superiore PV. Se CD ha un fronte di salita da FALSE a TRUE, CV viene abbassato di 1, a condizione che CV sia maggiore di 0 (cioè che non faccia scendere il valore sotto lo 0). Q restituisce TRUE quando CV è minore o uguale a 0.

#### 2.2.6 Counter - CTUD

Il contatore si presenta graficamente come in figura 9.

Gli ingressi sono:

- CU : BOOL (conteggio ascendente);
- CD : BOOL (conteggio discendente);
- RESET : BOOL (ripristino del contatore a zero);
- LOAD : BOOL (valore dal quale iniziare il conteggio)
- PV : WORD (limite del contatore);

Le uscite sono:

- QU : BOOL (uscita quando il contatore raggiunge il limite);
- QD : BOOL (uscita quando il contatore raggiunge zero);
- CV : WORD (valore corrente del conteggio);

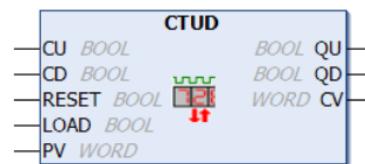


Figura 9: CTUD

Se RESET è valido, la variabile del contatore CV viene inizializzata con 0. Se LOAD è valido, CV viene inizializzata con PV. Se CU ha un fronte di salita da

FALSE a TRUE, CV viene incrementato di 1, a condizione che CV non causi un overflow. Se CD ha un fronte di salita da FALSE a TRUE, CV verrà abbassato di 1, a condizione che ciò non provochi una caduta del valore al di sotto di 0. QU restituisce TRUE quando CV è diventato maggiore o uguale a PV. QD restituisce VERO quando CV è diventato minore o uguale a 0.

### 3 Dispositivi utilizzati nella prova

- PLC **Beckhoff - CX8090** (Figura 10), è un sistema di controllo con una porta Ethernet commutata. Supporta protocolli come Ethernet in tempo reale, ADS UDP/TCP, client/server Modbus TCP o comunicazione aperta TCP/IP-UDP/IP. I terminali K-bus o E-bus possono essere collegati secondo necessità; il CX8090 riconosce automaticamente il tipo di sistema I/O collegato durante la fase di avvio. Il sistema di controllo viene programmato con TwinCAT tramite l'interfaccia bus di campo o l'interfaccia Ethernet aggiuntiva. Per ulteriori informazioni di carattere tecnico, è possibile consultare il **datasheet** relativo al prodotto.
- Cavo **Ethernet** per garantire la trasmissione e lo scambio di dati tra personal computer e PLC.
- Software **TwinCAT 2** - TX1200, versione 2.11. Sviluppato da Beckhoff Automation, è un ambiente di sviluppo per il controllo e l'automazione industriale.
- La stazione è poi costituita da componenti come cilindri a semplice effetto, braccio pneumatico, sensori di rilevamento, fine corsa ed attuatori, come riportato anche nel paragrafo 5, con le tabelle 1 e 2.

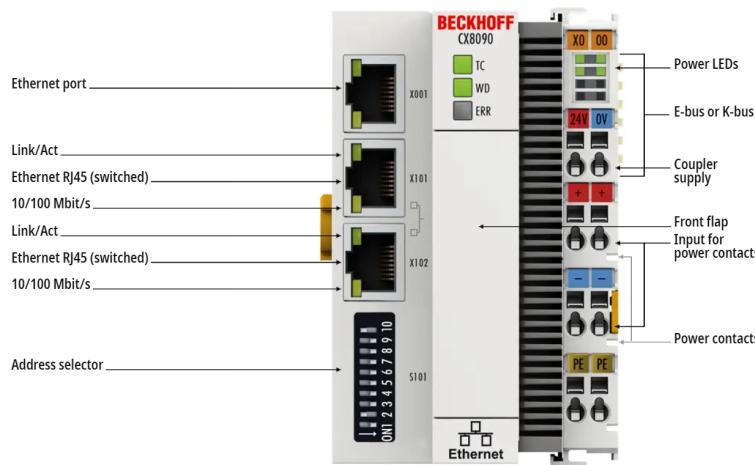


Figura 10: Beckhoff CX8090

## 4 Descrizione del processo e dell'impianto

Nella sua interezza, la stazione si mostra come in (fig. 11). Si compone poi di sotto processi, descritti nei paragrafi successivi.

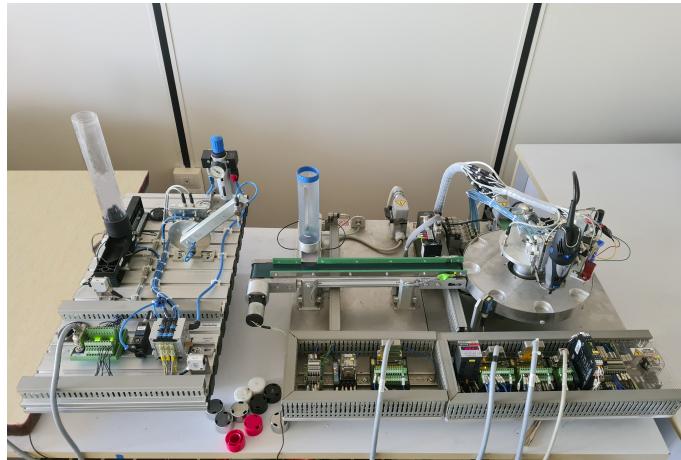


Figura 11: Stazione completa

### 4.1 Caricamento

Questa prima parte è composta da un **magazzino gravitazionale** ed un **cilindro a semplice effetto**; si comportano come buffer in ingresso. È inoltre presente un braccio pneumatico (fig. 12) per il caricamento ed il successivo scarico del pezzo grezzo sul nastro trasportatore. Il cilindro a semplice effetto viene movimentato andando ad agire su di una elettrovalvola che permette il passaggio di aria.

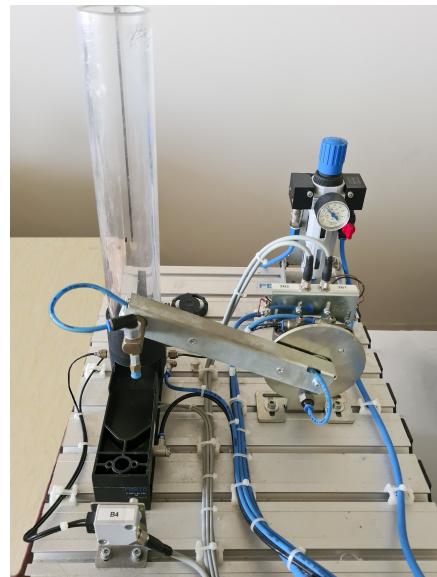


Figura 12: Braccio pneumatico

## 4.2 Trasporto

In quest'area, si trova il **nastro trasportatore** ed il secondo magazzino gravitazionale con accesso diretto sul nastro mediante **cilindro a semplice effetto** (figura 13). I pezzi grezzi dal buffer vengono spinti direttamente sul nastro, senza l'ausilio del braccio pneumatico, come nel caso precedente (paragrafo 4.1).

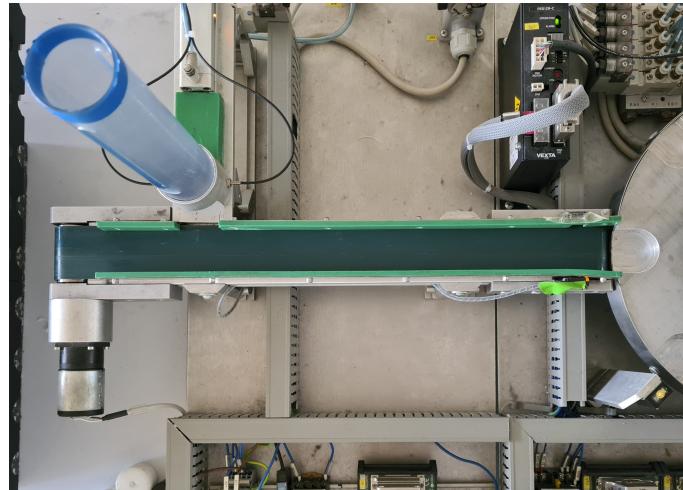


Figura 13: Nastro trasportatore

## 4.3 Lavorazione

Nella terza ed ultima zona dell'assieme, è presente la **giostra multi-lavorazione**, in grado di effettuare operazioni di riconoscimento colore, foratura, controllo altezza e marcatura indelebile (fig. 14). Inoltre, essa prevede anche lo scarico del pezzo lavorato in uno dei due depositi.

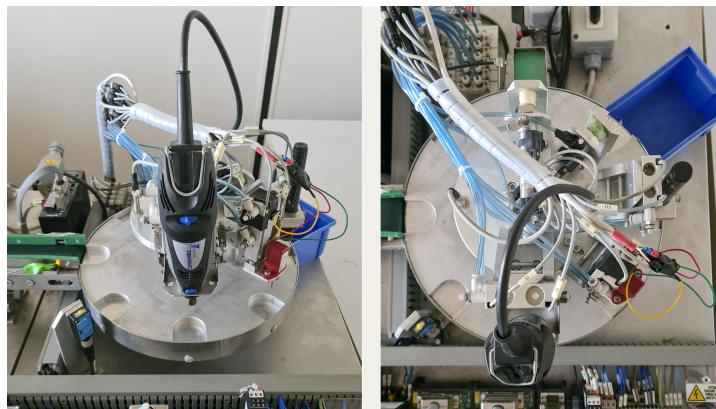


Figura 14: Giostra multi-lavorazione

## 4.4 Dettaglio funzionamento

Il controllore deve essere in grado di gestire la flessibilità di prodotto, vale a dire **caricare** i diversi pezzi grezzi sul nastro trasportatore da entrambi i buffer iniziali. Dal primo buffer, dopo aver azionato il **cilindro a semplice effetto**, il **braccio pneumatico** prende il pezzo e lo porta sul nastro, sfruttando un **vacuostato** con una ventosa. Invece, il secondo buffer, viene svuotato azionando il cilindro a semplice effetto direttamente sul nastro.

Caricati i pezzi, viene azionato un motore che muove il **nastro** e trasporta i pezzi grezzi, in attesa di lavorazione, sino all'ingresso della giostra.

In virtù della posizione sulla giostra, il pezzo grezzo subirà diverse lavorazioni. Dapprima, grazie ad un **sensore** di colore, verranno distinti i pezzi rossi dai pezzi di altro colore.

Dovranno poi essere svolte le seguenti operazioni:

- **Pezzi rossi:** foratura e deposito nella baia;
- **Pezzi non rossi:** misurazione dell'altezza a campione con un tastatore (simulato), marcatura con un pennarello e deposito nel buffer.

Terminate le lavorazioni, i pezzi verranno depositati nel rispettivo buffer finale grazie all'azionamento di un cilindro a semplice effetto. Se un pezzo finisce nella **baia**, l'intero sistema si blocca e, sul **pannello virtuale**, compare l'allarme che indica la necessità di un intervento da parte dell'operatore per liberare la baia stessa. Inoltre, nell'intero sistema sono presenti pulsanti e indicatori luminosi virtuali che permettono l'**avvio del ciclo**, l'interruzione in caso di **emergenza**, l'azionamento del **pistone 1B1** se è presente un pezzo nel magazzino gravitazionale (paragrafo 4.1) ed infine la verifica della conformità rispetto all'altezza dei pezzi a campione. Per il pannello operatore si veda il capitolo 9, in cui vengono riportate nel dettaglio tutte le possibili situazioni in cui può trovarsi il sistema.

## 5 Tabelle collegamenti interfaccia

Il programma caricato sul controllore opera direttamente su indirizzi di ingressi ed uscite collegate al terminale. Agli indirizzi è possibile associare un nome di una variabile che permetta di ricordare quale elemento reale è associato all'indirizzo virtuale e di descriverne la funzione svolta. Questa associazione indirizzo/variabile viene descritta nelle tabelle 1 e 3:

### 5.1 Assegnazione indirizzi ingressi

Tabella 1: INDIRIZZI VIRTUALI INGRESSI

Nome	Indirizzo	Variabile	Contatto
Presenza pezzo	%IX0.0	BOOL	N.O.
Colore giostra	%IX0.1	BOOL	N.O.
Pistone avanti	%IX0.2	BOOL	N.O.
Trapano alto	%IX0.3	BOOL	N.O.
Pistone indietro	%IX0.4	BOOL	N.O.
Trapano basso	%IX0.5	BOOL	N.O.
Fine corsa nastro	%IX0.6	BOOL	N.C.
Tastatore alto altezza	%IX0.7	BOOL	N.O.
Scarto giostra avanti post 6	%IX1.0	BOOL	N.O.
Scarto giostra indietro post 6	%IX1.1	BOOL	N.O.
Presenza pezzo baia	%IX1.2	BOOL	N.O.
Tastatore basso altezza	%IX1.3	BOOL	N.O.
Pennarello alto	%IX1.4	BOOL	N.O.
Pennarello basso	%IX1.5	BOOL	N.O.
Scarto giostra avanti post 7	%IX1.6	BOOL	N.O.
Scarto giostra indietro post 7	%IX1.7	BOOL	N.O.
Giostra in movimento	%IX2.0	BOOL	N.O.
Sensore vacuostato	%IX2.5	BOOL	N.O.
Fine corsa braccio magazzino	%IX2.6	BOOL	N.O.
Presenza pezzo magazzino B4	%IX2.7	BOOL	N.C.
Pistone indietro 1B2	%IX3.0	BOOL	N.O.
Pistone avanti 1B1	%IX3.1	BOOL	N.O.
Fine corsa braccio nastro	%IX4.1	BOOL	N.O.

### 5.1.1 Associazione input channel

Per far sì che il controllore comunichi con i sensori, a livello software, si effettua l'associazione delle variabili (usate nel main del programma) con i canali dove fisicamente sono connesse le periferiche.

Tabella 2: ASSEGNAZIONE CANALI - INPUT

Nome	Terminale	Canale
Presenza pezzo	2	1
Colore giostra	2	2
Pistone avanti	2	3
Trapano alto	2	4
Pistone indietro	2	5
Trapano basso	2	6
Fine corsa nastro	2	7
Tastatore alto altezza	2	8
Scarto giostra avanti post 6	5	1
Scarto giostra indietro post 6	5	2
Presenza pezzo baia	5	3
Tastatore basso altezza	5	4
Pennarello alto	5	5
Pennarello basso	5	6
Scarto giostra indietro post 7	5	7
Scarto giostra avanti post 7	5	8
Giostra in movimento	6	2
Sensore vacuostato	6	6
Fine corsa braccio magazzino	6	7
Presenza pezzo magazzino B4	6	8
Pistone indietro 1B2	9	1
Pistone avanti 1B1	9	2
Fine corsa braccio nastro	11	2

## 5.2 Assegnazione indirizzi uscite

Tabella 3: INDIRIZZI VIRTUALI USCITE

Nome	Indirizzo	Variabile	Contatto
Nastro	%QX0.0	BOOL	/
Pistone giostra post 6	%QX0.1	BOOL	/
Pistone nastro	%QX0.2	BOOL	/
Pennarello down	%QX0.3	BOOL	/
movimento giostra 1 posto	%QX0.4	BOOL	/
Pistone giostra post 7	%QX0.5	BOOL	/
Trapano down	%QX0.6	BOOL	/
Trapano ON	%QX0.7	BOOL	/
Tastatore down	%QX1.0	BOOL	/
Pistone orizzontale	%QX2.1	BOOL	/
Vacuostato	%QX2.2	BOOL	/
Soffio aria ventosa	%QX2.3	BOOL	/
Braccio pneumatico indietro	%QX2.4	BOOL	/
Braccio pneumatico avanti	%QX2.5	BOOL	/

### 5.2.1 Associazione output channel

Per far sì che il controllore comunichi con gli attuatori, a livello software, si effettua l'associazione delle variabili (usate nel main del programma) con i canali dove fisicamente sono connesse le periferiche.

Tabella 4: ASSEGNAZIONE CANALI - OUTPUT

Nome	Terminale	Canale
Nastro	3	1
Pistone giostra post 6	3	2
Pistone nastro	3	3
Pennarello down	3	4
movimento giostra 1 posto	3	5
Pistone giostra post 7	3	6
Tastatore down	3	7
Trapano ON	3	8
Trapano down	7	1
Pistone orizzontale	8	2
Vacuostato	8	3
Soffio aria ventosa	8	4
Braccio pneumatico indietro	8	5
Braccio pneumatico avanti	8	6

## 6 Legenda qualificatori - Standard IEC 1131-3

Nel linguaggio SFC, ad ogni passo, oltre ad una entry o exit action, si possono attribuire delle azioni ben precise, chiamate "IEC actions", in virtù della norma IEC 1131-3 che definisce i linguaggi di programmazione. I passi vengono etichettati da **qualificatori**, i quali descrivono la modalità di esecuzione dell'azione associata al passo stesso. Ognuno dei qualificatori presenta un diverso significato; vengono descritti nella tabella 5 seguente.

Tabella 5: QUALIFICATORI - IEC

Indicatore	Legenda
N	L'azione viene eseguita per tutta la durata dello step
S	L'azione viene eseguita e rimane attiva fino ad eventuale azione di Reset (R)
R	L'azione eseguita con SET viene disattivata
L	L'azione eseguita termina dopo un certo intervallo di tempo che può durare al massimo fino a che lo step è attivo
D	L'azione viene eseguita dopo un certo tempo e poi rimane attiva per tutta la durata dello step
P	L'azione viene eseguita solo una volta se lo step è attivo
SD	L'azione viene eseguita come azione SET dopo un certo intervallo di tempo dall'attivazione dello stato e rimane attiva fino ad un Reset (R)
DS	L'azione viene eseguita come azione SET se lo stato rimane attivo per un certo intervallo di tempo e rimane attiva fino ad un Reset (R)
SL	L'azione viene eseguita come SET, dopo un certo tempo l'azione viene terminata con un'azione di Reset (R)

Graficamente, si presenta nel seguente modo (figura 15):

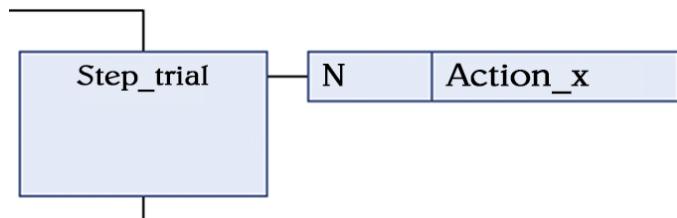


Figura 15: IEC 1131-3, step actions

## 7 Programma in Sequential Functional Chart (SFC)

La programmazione in SFC è stata utilizzata per la gestione dei pezzi nella giostra, le eventuali lavorazioni e le operazioni di inizio e fine del processo. La logica utilizzata prevede che, attraverso il sensore di colore, si abbiano due differenti linee di lavorazione:

- I pezzi rilevati dal sensore, ovvero i pezzi rossi, subiscono l'operazione di foratura e vengono depositati nella baia, dalla quale un operatore deve rimuovere manualmente il pezzo per far proseguire il funzionamento;
- I pezzi non rilevati dal sensore vengono marcati dal pennarello, scaricati dal pistone “*post\_6*” e controllati a campione, ogni tre pezzi che passano in corrispondenza della posizione del tastatore; il check viene effettuato con un pulsante virtuale (paragrafo 9.4).

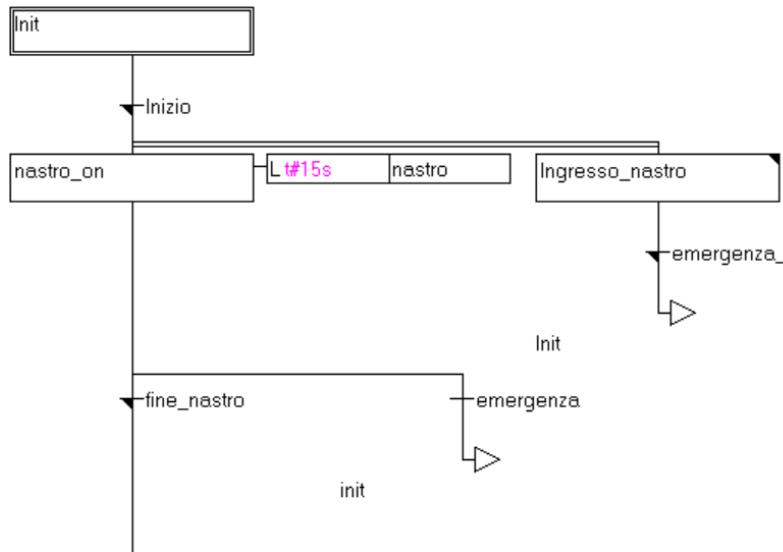


Figura 16: Inizializzazione lavorazione giostra - SFC

All'avvio del programma il token esce dal posto “init” (figura 16) se il controllo iniziale ha dato esito positivo, vale a dire se:

- Il pulsante virtuale “start” è attivato (figura 28);
- Il pulsante virtuale ”stop di emergenza” è disattivato (figura 28);
- La “baia” è libera.

Uscendo da “init” si avviano le seguenti azioni:

- Il nastro inizia a scorrere in attesa dell’arrivo di un pezzo (se esso non arriva entro quindici secondi, il nastro stesso si arresta automaticamente);
- Il token resta su “*ingresso\_nastro*” (nel quale l’azione eseguita è l’attivazione di tutta la parte discussa nel successivo capitolo 8) fino a quando non viene attivato lo stop di emergenza o viene rilevato un pezzo sulla baia.

Dal passo “*nastro\_on*” il programma esce in caso di emergenza oppure quando arriva un pezzo pronto per essere caricato nella giostra (rilevato dal fine corsa); in quest’ultimo caso si va nello step “*pezzo\_attesa*” (figura 17).

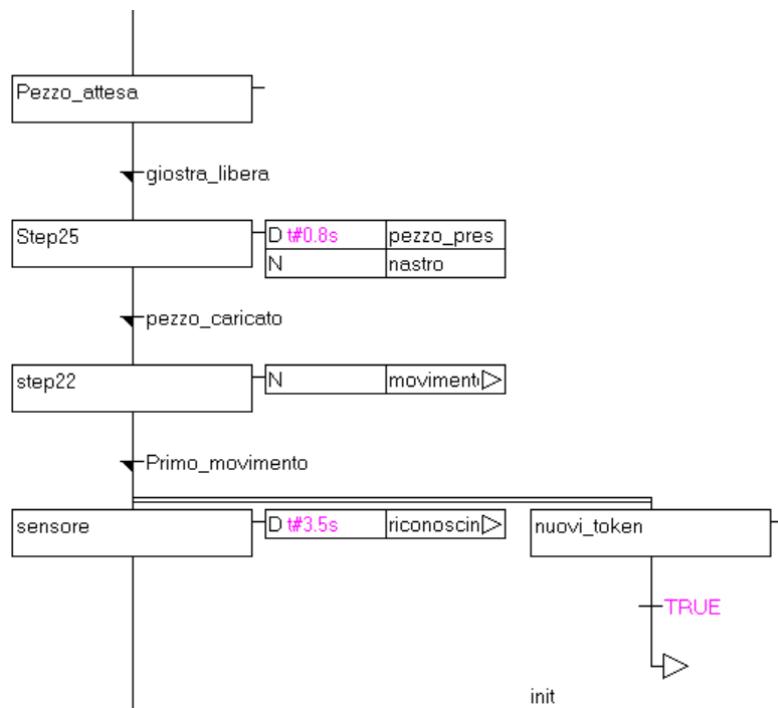


Figura 17: Primo movimento girostra - SFC

A questo punto se la girostra è in lavorazione il nastro si ferma per attendere la fine della lavorazione precedente prima di caricare il pezzo; se invece non è in lavorazione, si passa allo “*step25*” nel quale si attendono 0,8 secondi grazie ai quali il pezzo dal fine corsa del nastro viene effettivamente caricato sulla girostra. Arrivando ora allo “*step22*” la girostra si muove di un posto mandando sia il pezzo in corrispondenza del sensore di colore e sia facendo tornare un token all’inizio per aspettare l’arrivo di un nuovo pezzo.

Come mostrato in figura 18, se il sensore rileva la presenza di un rosso il token va nel ramo di sinistra (“*rosso\_rilevato*”); nel caso contrario, in quello di destra (“*non\_rosso*”).

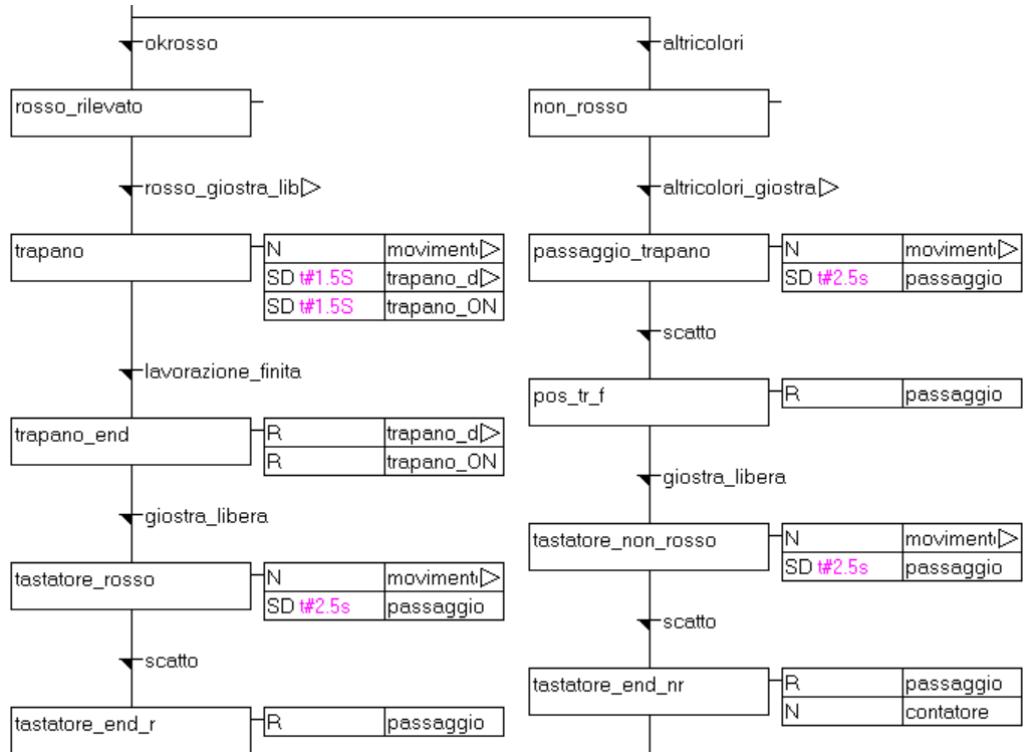


Figura 18: Selezione per colore - SFC

Ora si attende che la giostra sia libera o che un eventuale nuovo pezzo sia arrivato al sensore per iniziare le effettive lavorazioni. Il movimento avviene se viene soddisfatta la condizione “*giostra\_libera*”; come mostrato in figura 19, la condizione è vera se:

- non è presente un pezzo nella baia;
- la giostra non sta effettuando alcun tipo di lavorazione (il trapano, il pennarello, i pistoni di scarico ed il tastatore sono alti);
- la giostra non è in movimento;
- non vi è un pezzo in caricamento nella giostra (rappresentato da “*PEZZO\_ATTESA.x*” e “*STEP25.x*”);
- non è stato premuto il pulsante stop di emergenza;
- non è il momento di effettuare un controllo a campione.

Figura 19: Condizione ”*giostra\_libera*” - LD

A questo punto, nel caso di pezzo rosso, una volta arrivati su “*trapano*” la giostra si muove per portare il pezzo dal sensore al trapano, il quale si attiva dopo 1,5 secondi (tempo necessario al movimento della giostra) e scende per effettuare la foratura. Quando il trapano è completamente basso (“*lavorazione\_finita*”) il token passa allo step “*trapano\_end*” nel quale si resettano le variabili attivate per far scendere e avviare il trapano. Da questo passo il sistema esce solo quando tutte le lavorazioni in tutti i posti della giostra sono terminate (“*giostra\_libera*”). Questo approccio mira a sincronizzare i movimenti della giostra ponendo l’attenzione su eventuali altre lavorazioni ancora in corso nella giostra.

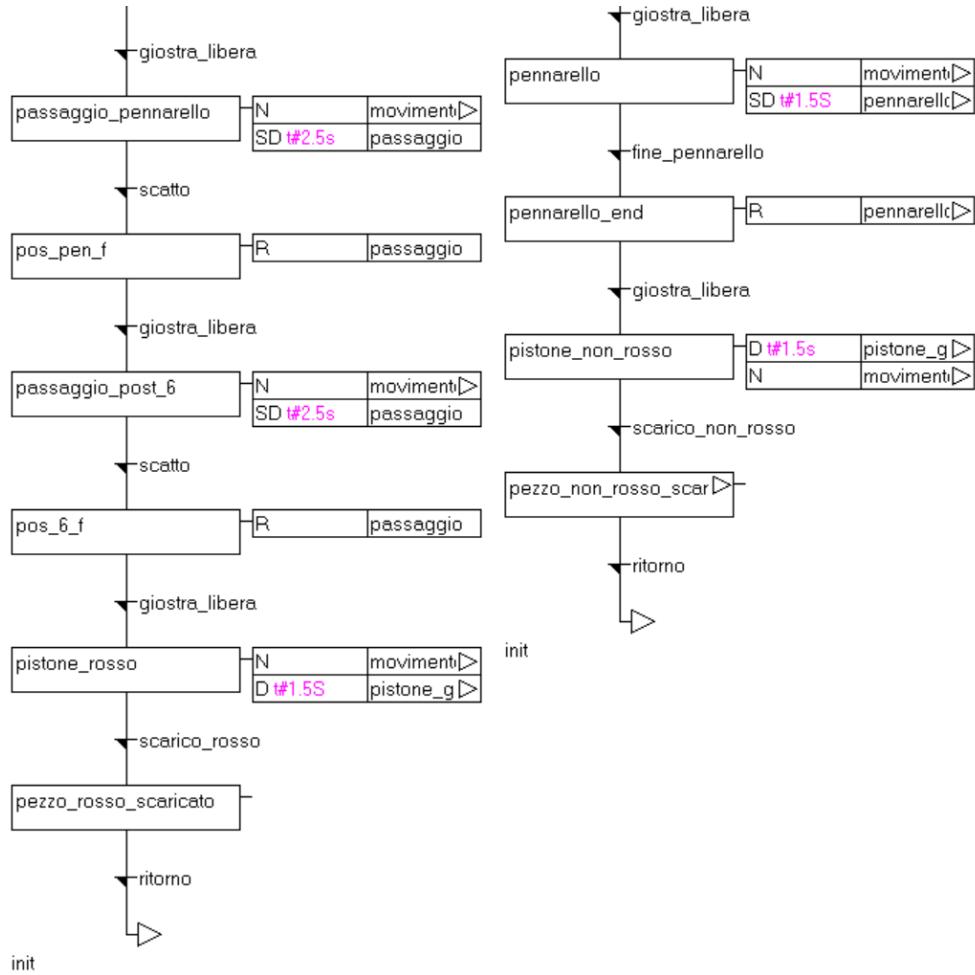


Figura 20: Lavorazioni giorstra - SFC

Con la stessa ottica sono state implementate tutte le altre lavorazioni da fare nel caso di pezzo non rosso (ramo di destra in figura 20), sfruttando una variabile di appoggio “*passaggio*” per i passaggi da fare senza lavorazioni.

Inoltre, anche il tastatore è stato simulato sfruttando la suddetta variabile, in virtù del suo malfunzionamento che non permetteva l’esecuzione corretta del processo. Ciò nonostante, nella transizione “*giostra.libera*” è stato inserito un contatto “*TRUE*” al posto del contatto “*tastatore.alto*”, il quale andrebbe sostituito nel caso di effettivo utilizzo del sensore analogico. In particolare, la simulazione del tastatore prevede che, ogni qualvolta che un pezzo non rosso passa sotto il tastatore, viene incrementato di una unità un contatore che, se arrivato a satu-

razione (cioè 3 incrementi), fa accendere il led virtuale “*controllo a campione*” (si veda paragrafo 9.4). A questo punto l’operatore dovrà controllare il pezzo e premere il pulsante “*pezzo ok*” se il controllo è andato a buon fine, oppure il bottone di stop di emergenza in caso contrario. In figura 19 è mostrata la logica con cui si effettua il conteggio.

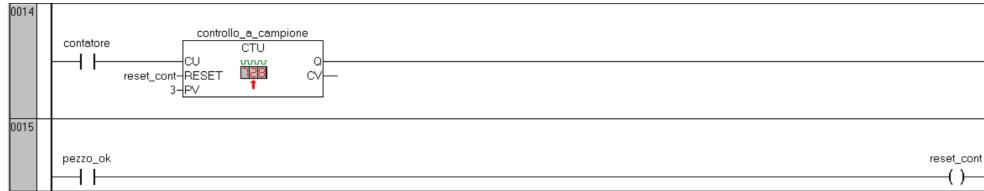


Figura 21: CTU controllo a campione - LD

Per concludere, una volta che i pezzi sono arrivati in corrispondenza della postazione del cilindro “*post\_6*”, esso viene attivato per scaricare il pezzo dalla giostra; quando il pistone è tutto avanti, l’attivazione del rispettivo finecorsa fa sì che il token, attraverso una transizione jump, torni all’inizio del diagramma pronto per rilevare un nuovo pezzo in arrivo.

## 7.1 Possibile alternativa tastatore

Nel programma principale, la simulazione del tastatore prevede che, a campione, un pezzo venga semplicemente ispezionato. Un’altra possibile realizzazione, viene descritta di seguito. Il controllo a campione viene declassato e, al suo posto, per ogni pezzo non rosso che transita sotto il tastatore si pone il sistema davanti ad una scelta:

- se il pezzo è alto, allora subisce la marcatura dal pennarello nella postazione seguente;
- se il pezzo è basso, la marcatura non viene effettuata.

Chiaramente, chi decide l’altezza del pezzo è l’operatore, attraverso il pannello operatore. Infatti, nel paragrafo 9.5 è riportata la condizione in cui l’indicatore luminoso è attivo ed il sistema, per procedere, deve attendere che l’operatore faccia la sua scelta. L’estratto di SFC in figura 22 mostra l’implementazione della suddetta logica (la diramazione è effettuata immediatamente dopo l’ultimo step nel ramo di destra in figura 18).

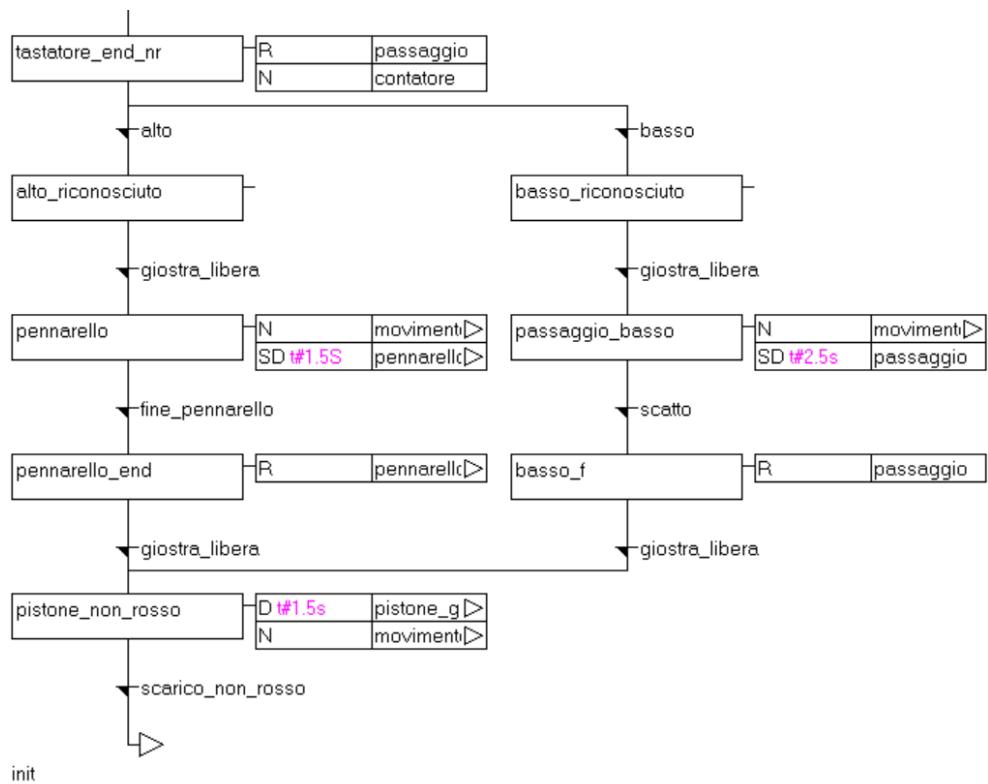


Figura 22: Selezione per altezza - SFC

## 8 Programma in Ladder (LD)

Nella realizzazione del progetto si è voluto proseguire col gestire entrambi i magazzini mediante il linguaggio ladder. In particolare, la logica implementata prevede che il magazzino con accesso diretto nel nastro effettui la spinta dei pezzi verso il nastro ad intervalli di 1.8 secondi; tuttavia, tale azione viene inibita in presenza di un caricamento imminente (sul nastro) da parte del braccio pneumatico. Questo garantisce la precedenza, nel caricamento, del magazzino B4, evitando collisioni e sovrapposizioni durante il trasporto.

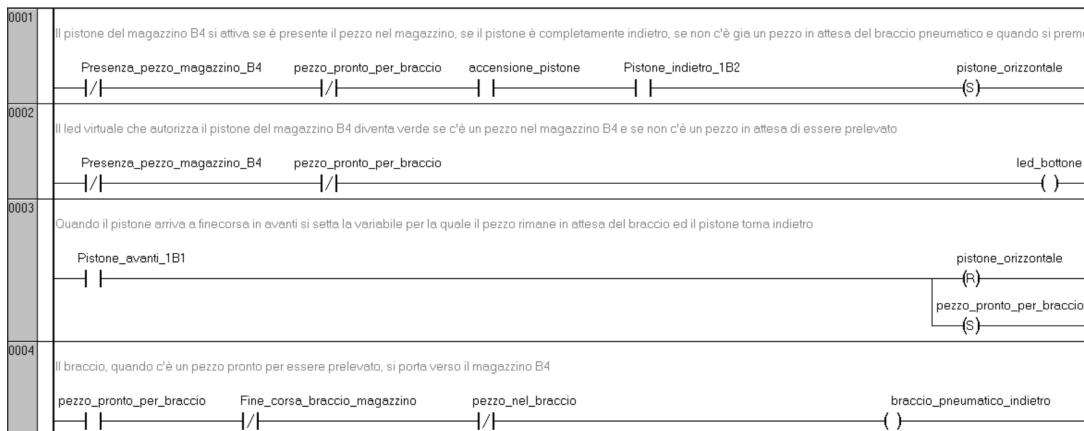


Figura 23: Gestione magazzino B4 - LD

In questa parte di codice (figura 23) si gestisce il magazzino con braccio pneumatico (B4); alla pressione di un pulsante virtuale sul pannello operatore (figura 29), il pistone orizzontale si attiva se si verificano queste condizioni:

- Magazzino gravitazionale occupato, cioè *"presenza\_pezzo\_magazzino\_B4"* deve trovarsi allo stato alto. Da notare che il contatto è negato, perché vi è un sensore che lavora a logica inversa (si veda tabella 1);
- Posizione in attesa del braccio pneumatico libera (gestita con la variabile *"pezzo\_pronto\_per\_braccio"*, variabile che si attiva con un set quando il pistone ha portato il pezzo in posizione e si resetta una volta che tale pezzo viene prelevato dal braccio).

Soddisfatte le condizioni, si gestisce il braccio portandolo indietro fino ad arrivare al finecorsa verso il magazzino, a patto che non sia già pieno (gestito da *"pezzo\_nel\_braccio"*). In quest'ultima posizione, si attiva il vacuostato che permetterà al braccio di movimentare il pezzo e verrà settata una variabile di appoggio *"pezzo\_nel\_braccio"*, usata per evitare conflitti; tale variabile si resetta ogni qual volta che il pezzo ha superato, lungo il nastro, la postazione del magazzino orizzontale in modo da lasciare spazio e liberare per un nuovo caricamento (figura 24).

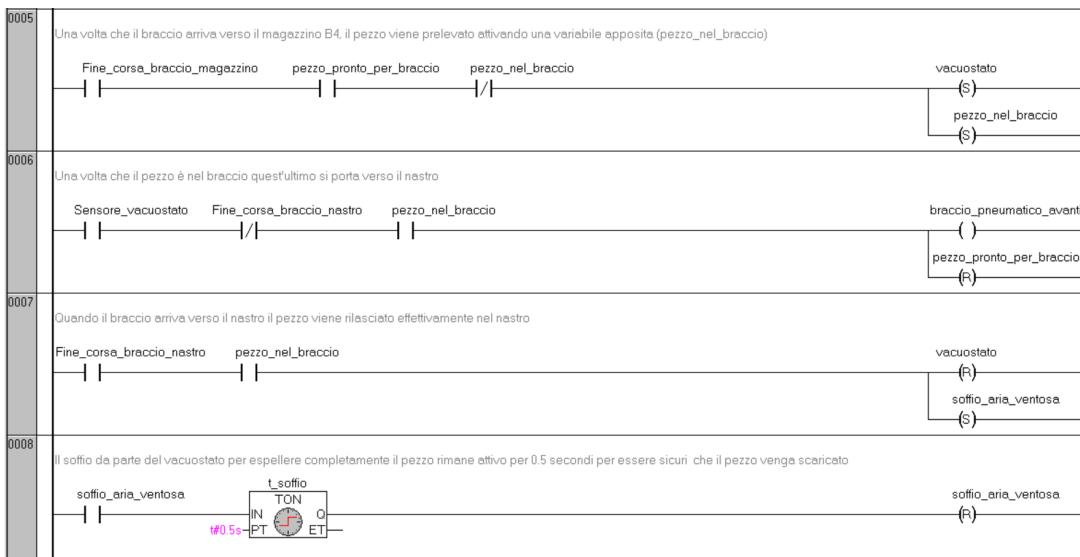


Figura 24: Sgombero ventosa con regolazione soffio - LD

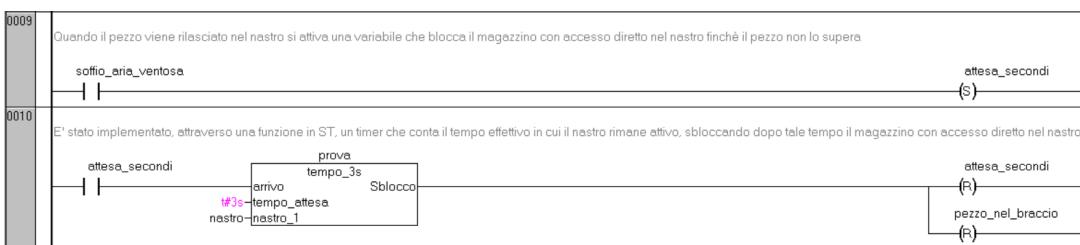


Figura 25: Chiamata function block - LD

Resta ora da gestire il timer che blocca il secondo magazzino (quello con il pistone ad accesso diretto sul nastro) per un totale di 3 secondi dopo il rilascio del pezzo nel nastro dal braccio pneumatico. L'attivazione della variabile "attesa\_secondi" segnala l'inizio del periodo di attesa. Durante questo periodo, il secondo magazzino resta inattivo fintanto che il pezzo non ha superato la sua posizione sul nastro. Non avendo sensori disponibili, per implementare questa logica occorre temporizzare e contare il tempo in cui il nastro è effettivamente attivo, essendo il movimento del nastro stesso gestito dalla parte SFC della giostra. Si è scelto di usare un TON (Timer On-Delay, paragrafo 2.2.1) creando una function block (figura 25) che implementa la logica voluta di contare il tempo in cui il nastro resta effettivamente attivo.

La funzione (figura 26) prende in ingresso:

- "arrivo", il quale rappresenta la variabile che blocca il pistone del magazzino con accesso sul nastro;
- "tempo\_attesa", che rappresenta il tempo effettivo da attendere;
- la variabile "nastro".

```

0001 FUNCTION_BLOCK tempo_3s
0002 VAR_INPUT
0003   arrivo: BOOL;
0004   tempo_attesa: TIME;
0005   nastro_1: BOOL;
0006 END_VAR
0007 VAR_OUTPUT
0008   Sblocco:BOOL:= FALSE;
0009 END_VAR
0010 VAR
0011   MyTimer: TON;
0012   TotalET: TIME:= T#0s;
0013   prova: TIME:= T#0s;
0014 END_VAR
0015

0001 MyTimer(N := (nastro_1 AND arrivo), PT := tempo_attesa);
0002 IF nastro_1 = TRUE
0003 THEN
0004   THEN
0005     prova := MyTimer.ET;
0006 END_IF
0007
0008 IF nastro_1 = FALSE
0009 THEN
0010   TotalET := TotalET + prova;
0011   prova := #0s;
0012 END_IF
0013
0014 IF ((TotalET + prova) >= tempo_attesa)
0015 THEN
0016   sblocco := TRUE;
0017   TotalET := #0s;
0018   prova := #0s;
0019 ELSE
0020   sblocco := FALSE;
0021 END_IF
0022
0023
0024

```

Figura 26: Estratto ST della funzione "timer con memoria"

In uscita si ha la variabile “*sblocco*” che andrà a dare il via libera per il magazzino con accesso sul nastro al tempo opportuno. Il tempo viene calcolato attraverso il TON e memorizzato, nel caso in cui il nastro si spenga mentre il timer sta contando, nella variabile “*TotalET*”. Lo sblocco del magazzino avviene ora quando la combinazione dei tempi (*TotalET* + *prova*) raggiunge il tempo assegnato; si specifica che (*TotalET* è il tempo totale memorizzato, invece *prova* è il tempo corrente).

In maniera del tutto analoga è stata gestita la parte relativa al secondo cilindro, cioè quello del magazzino con accesso diretto sul nastro; è stata implementata una logica temporizzata che regola la frequenza di caricamento a 1.8 secondi tra un pezzo e l’altro (riga 13 in figura 27). Si noti come, anche in questo caso, l’utilizzo della function block sopracitata “*tempo\_3s*”, ha permesso di realizzare il comportamento desiderato.

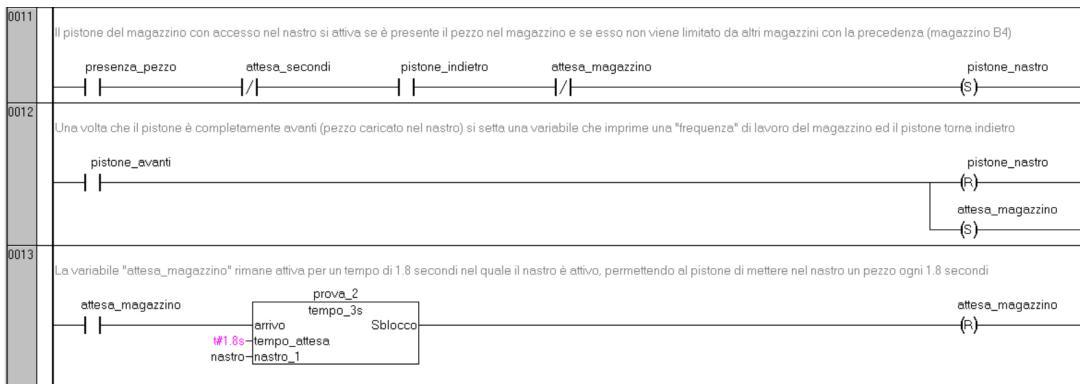


Figura 27: Gestione cilindro diretto - LD

## 9 Pannello operatore virtuale

Per sopperire alla mancanza di pulsanti fisici per l'azionamento del processo, l'arresto in caso di emergenza e l'avvio del cilindro B4, sono stati realizzati dei pulsanti virtuali. Essi non fanno altro che, via software, emulare un impulso elettrico proveniente da un pulsante esterno. Di seguito, verranno mostrate configurazioni in cui è possibile incorrere.

### 9.1 Funzionamento normale

In figura 28 è mostrato il pannello operatore virtuale, in una situazione di normale funzionamento. Si può notare:

- Pulsante **start**, per avviare il sistema;
- Pulsante **stop**, per arrestare in caso di emergenza;
- Pulsante **magazzino B4**, per azionare il cilindro a semplice effetto del primo buffer;
- Indicatore luminoso **"pezzo in attesa"**, se rosso indica che nel magazzino B4 non è presente il pezzo e il cilindro non può partire, anche se azionato;
- Indicatore luminoso **"liberare baia"**, se bianco non vi è la necessità di un intervento da parte dell'operatore, in quanto la baia è libera;
- Indicatore luminoso **"controllo a campione"**, se bianco non vi è la necessità di controllare la conformità del pezzo campione.

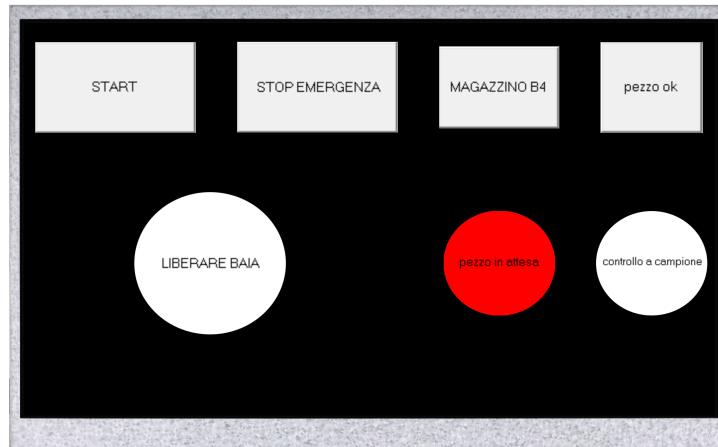


Figura 28: Pannello operatore - normale funzionamento

## 9.2 Magazzino B4 caricato

In figura 29 è mostrato il pannello operatore virtuale quando nel magazzino B4 ci sono pezzi pronti per essere espulsi dal cilindro a semplice effetto. Si può notare come l'indicatore luminoso **"pezzo in attesa"** sia divenuto verde, cambiando il suo colore da rosso.



Figura 29: Pannello operatore - magazzino B4 caricato

## 9.3 Baia occupata

In figura 30 è mostrato il pannello operatore virtuale quando nella baia è presente un pezzo. In base a com'è stato gestito il sistema, questo avviso richiede l'intervento da parte di un operatore per rimuovere il pezzo (rosso) dalla baia stessa. Se attivo, dunque, l'intero sistema si arresta e l'indicatore **"liberare baia"** diventa di colore giallo, fino a intervento ultimato.



Figura 30: Pannello operatore - baia occupata

## 9.4 Controllo campione

In figura 31 è mostrato il pannello operatore virtuale quando il contatore ha raggiunto il valore preimpostato per il controllo della conformità di un pezzo campione casuale. Il rispettivo indicatore **"controllo a campione"** diventa blu e torna bianco solo dopo che viene premuto il pulsante **"pezzo ok"**, indicando che il pezzo è stato effettivamente controllato. Durante questa fase, il sistema si blocca e riprende alla pressione del medesimo tasto.



Figura 31: Pannello operatore - controllo campione

## 9.5 Controllo altezza

In figura 32 è mostrato il pannello operatore virtuale quando il pezzo rosso è arrivato nella postazione del tastatore. Il rispettivo indicatore **"tastatore"** diventa blu e torna bianco alla pressione di uno dei due tasti, alto o basso. Durante questa fase, il sistema si blocca e riprende solo dopo aver effettuato la scelta (si veda paragrafo 7.1 per dettaglio).

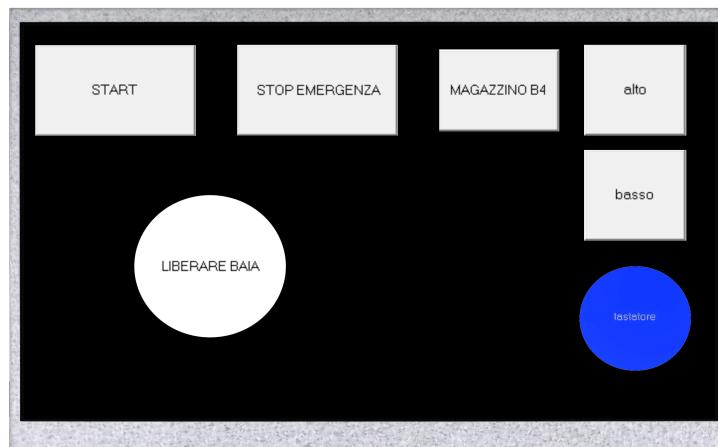


Figura 32: Pannello operatore - controllo altezza

## 10 Conclusioni

L'obiettivo iniziale della prova era quello di automatizzare una linea di produzione mediante l'impiego di un controllore a logica programmabile. Attraverso l'implementazione di un sistema efficace e flessibile, siamo riusciti ad ottimizzare il processo produttivo, gestire la sincronizzazione tra le diverse lavorazioni, aumentare l'efficienza e ridurre i tempi stessi di lavorazione. L'utilizzo del controllore ha consentito una gestione più precisa ed accurata delle attività, garantendo una maggiore affidabilità e riducendo al minimo gli errori umani.

Inoltre, la flessibilità offerta dal sistema ha permesso di adattarci rapidamente ad eventuali cambiamenti nelle esigenze produttive; infatti, sono stati implementati tre linguaggi di programmazione distinti, ognuno in grado di gestire in modo efficace e ottimale un compito ben preciso. Ad esempio, il sincronismo delle lavorazioni nella giostra è stato gestito con l'SFC, oppure il timer "con memoria", per evitare conflitti nel caricamento del nastro, è stato gestito con l'ST.

Per migliorare ulteriormente l'efficienza, questa volta in termini di consumo di potenza, è stato implementato un meccanismo che permette al nastro di fermarsi automaticamente dopo 15 secondi di inattività. Questa funzionalità non solo contribuisce a ridurre il consumo energetico durante i periodi di pausa o di bassa attività, ma aiuta anche a preservare la durata del sistema nel lungo termine. La combinazione di questa caratteristica con il controllore a logica programmabile ha consentito di massimizzare l'efficienza complessiva della linea di produzione, garantendo un impatto positivo sia sulle prestazioni che sull'ecosostenibilità dell'intero processo.

Altro aspetto degno di nota è l'implementazione di un pannello operatore; esso ha giocato un ruolo cruciale nel migliorare l'accessibilità della comunicazione tra l'operatore ed il processo produttivo. Infatti, questo strumento, fornisce un'interfaccia intuitiva ed user-friendly attraverso cui gli operatori possono monitorare e controllare le operazioni in tempo reale. Mediante poi l'uso di indicatori visivi e allarmi, il pannello operatore ha facilitato la tempestiva identificazione e risoluzione di eventuali anomalie, problemi o semplici segnalazioni nel processo produttivo.

In conclusione, questa esperienza di laboratorio, oltre che le capacità nella gestione di una linea di produzione, ci ha fornito soft skills come:

- capacità organizzative, rispetto delle tempistiche e delle scadenze, flessibilità ed adattabilità;
- autonomia;
- capacità di astrazione ed immaginazione;
- precisione e cura dei dettagli;
- problem solving;

- capacità di esprimere le proprie idee in modo chiaro e di ascoltare attivamente i membri del team.

L'obiettivo è stato raggiunto, la prova può dichiararsi conclusa.

## Elenco delle figure

1	TON . . . . .	3
2	Andamento temporale - TON . . . . .	3
3	TOF . . . . .	4
4	Andamento temporale - TOF . . . . .	4
5	TP . . . . .	4
6	Andamento temporale - TP . . . . .	5
7	CTU . . . . .	5
8	CTD . . . . .	6
9	CTUD . . . . .	6
10	Beckhoff CX8090 . . . . .	7
11	Stazione completa . . . . .	8
12	Braccio pneumatico . . . . .	8
13	Nastro trasportatore . . . . .	9
14	Giostra multi-lavorazione . . . . .	9
15	IEC 113-1, step actions . . . . .	14
16	Inizializzazione lavorazione giostra - SFC . . . . .	15
17	Primo movimento giostra - SFC . . . . .	16
18	Selezione per colore - SFC . . . . .	17
19	Condizione " <i>giostra.libera</i> " - LD . . . . .	17
20	Lavorazioni giostra - SFC . . . . .	18
21	CTU controllo a campione - LD . . . . .	19
22	Selezione per altezza - SFC . . . . .	20
23	Gestione magazzino B4 - LD . . . . .	21
24	Sgombero ventosa con regolazione soffio - LD . . . . .	22
25	Chiamata function block - LD . . . . .	22
26	Estratto ST della funzione "timer con memoria" . . . . .	23
27	Gestione cilindro diretto - LD . . . . .	23
28	Pannello operatore - normale funzionamento . . . . .	24
29	Pannello operatore - magazzino B4 caricato . . . . .	25
30	Pannello operatore - baia occupata . . . . .	25
31	Pannello operatore - controllo campione . . . . .	26
32	Pannello operatore - controllo altezza . . . . .	26

## Elenco delle tabelle

1	INDIRIZZI VIRTUALI INGRESSI . . . . .	11
2	ASSEGNAZIONE CANALI - INPUT . . . . .	12
3	INDIRIZZI VIRTUALI USCITE . . . . .	13
4	ASSEGNAZIONE CANALI - OUTPUT . . . . .	13
5	QUALIFICATORI - IEC . . . . .	14