



UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea triennale in INGEGNERIA ELETTRONICA

**RETROFIT DI UNA SPAZZOLATRICE PER STATORI: NUOVA
IMPLEMENTAZIONE SOFTWARE PLC**

**RETROFIT STATOR'S SWEEPER: NEW PLC SOFTWARE
IMPLEMENTATION**

Relatore: Chiar.mo/a

Prof. Silvia Maria Zanoli

Correlatore: Chiar.mo/a

Ing. Massimiliano Altobelli

Tesi di Laurea di:

Mattia Forzin

Matr. 1081683

A.A. 2019 / 2020

INDICE

Indice.....	1
Introduzione.....	2
Capitolo I Scopo del Lavoro.....	5
Capitolo II Il PLC.....	9
Capitolo III L'Automazione Esistente.....	31
Capitolo IV La Programmazione	36
Conclusioni.....	66
Bibliografia.....	68
Sitografia.....	69

INTRODUZIONE

L'Automazione Industriale è uno degli ambiti in cui l'ingegneria è fortemente radicata, può essere definita come la disciplina che studia le tecniche che portano al controllo dei flussi di energia, materiali e informazioni necessari ed utili per la realizzazione di processi produttivi.

L'obiettivo dell'Automazione Industriale può esprimersi come l'intento di creare un processo produttivo privo dell'errore umano, che possa essere ripetitivo e abbia costi ridotti, garantendo standard di qualità sul prodotto.

L'Automazione industriale, inoltre, ha subito notevoli sviluppi nel tempo tramite l'avvento di tecnologie in grado di garantire gli standard sempre più elevati a cui oggi si è abituati ad esigere. Nei primi decenni del '900 questa disciplina poteva avvalersi di strumenti ben più limitanti di quelli odierni, le automazioni erano fondamentalmente comandate da immensi quadri elettrici con logica cablata a relè, con costi immensi di manutenzione e riadattabilità dell'impianto. Un passo decisivo è stato segnato negli anni sessanta con la comparsa dei PLC, controllori logici programmabili, che hanno totalmente soppiantato l'impiego dei relè per come si era abituati, da allora i relè hanno perso il ruolo di controllori logici e spesso sono usati a valle dei PLC come interruttori comandati e in ultimo come protezione dei contatti di uscita del PLC. Anche i PLC hanno avuto enormi evoluzioni permettendo una programmazione sempre più agevole e ad alto livello, ma anche una maggiore interazione con l'uomo attraverso messaggi di allarme

precisi. L'avvento di Internet, poi, ha ulteriormente ampliato le possibilità comunicative delle macchine, agevolando così gli operatori in tutte le attività di produzione o di manutenzione delle macchine, permettendo lo scambio di dati con sistemi supervisivi o la storicizzazione di immagini su cloud. Negli ultimi anni si stanno verificando enormi cambiamenti dal punto di vista del miglioramento delle condizioni lavorative dell'uomo attraverso l'impiego di tecnologie improntate sulla collaborazione delle macchine. Esempi di questo processo sono i co-bot, l'impiego dell'intelligenza artificiale, l'impiego dell'Internet of Things.

Dal punto di vista ambientale però trova posto anche una forte politica del riuso, ove possibile, di automazioni esistenti tramite le tecniche del Revamping o del Retrofit.

In particolare, in questo lavoro di tesi, verrà trattato il Retrofit di una spazzolatrice per statori, ossia il riammodernamento software di una automazione già esistente in uno stabilimento industriale.

La Tesi è stata svolta in collaborazione con la Denso Manufacturing Italia S.p.A. che ha sede a San Salvo (CH) in Abruzzo.

A seguito di una modifica al layout, per far spazio a nuove automazioni è stato richiesto un ammodernamento software di una spazzolatrice per statori già esistente in stabilimento.

Tale ammodernamento ha riguardato l'introduzione di un nuovo sistema di controllo dell'usura della mola di spazzolatura per far fronte all'ammontare di pezzi di scarto dati da abrasione eccessiva sul pacco statorico in alluminio e cercare di allungare la vita alle mole impiegate in lavorazione. Oltre ciò la modifica prevede una riorganizzazione dell'architettura software e del linguaggio di programmazione per poter rientrare all'interno delle regole dettate dal capitolato aziendale.

La tesi è strutturata nel seguente modo:

- un primo capitolo in cui si delinea lo scopo della tesi, per cui viene introdotto il componente da lavorare e la richiesta da parte dell'azienda;
- un secondo capitolo in cui si ripercorre un pò di storia dell'Automazione Industriale per introdurre le caratteristiche dell'hardware che verrà utilizzato: il PLC, con descrizione delle funzionalità, dei componenti che lo costituiscono e l'importanza che ha assunto in quest'ambito.
- un terzo capitolo in cui viene descritta l'automazione presente in stabilimento ed esposta la modalità di funzionamento attuale con le problematiche da superare;
- un quarto capitolo in cui si mostra il processo che ha portato alla costituzione del nuovo software, evidenziando le tecniche in cui si è analizzato il ciclo di lavoro: flowchart e Reti di Petri; seguono poi le scelte di programmazione effettuate per riuscire a soddisfare le richieste aziendali;
- in ultimo sono state tratte delle conclusioni in cui si ricapitola il lavoro eseguito e si lascia aperta la strada a possibili migliorie future.

Capitolo I

Scopo del lavoro

La Denso Manufacturing Italia SpA ha sede a San Salvo (CH) in Abruzzo, è uno stabilimento che conta circa 1200 dipendenti tra operai e impiegati ed è una azienda del gruppo Denso, colosso giapponese, che nella sede abruzzese produce componenti per l'Automotive, in particolare: Alternatori e Motorini di Avviamento, per diversi Brand, sono i prodotti di punta.

Nello stabilimento principale di San Salvo, che copre circa 60000mq dove vi sono tutte le linee di produzione, il layout segue il flusso dei componenti, dall'arrivo dei grezzi, alla produzione dei sottogruppi, all'assemblaggio finale sulle linee di montaggio.

A seguito di una modifica al layout, per far spazio a nuove automazioni, è stato richiesto un ammodernamento software di una spazzolatrice per statori; della macchina attualmente in produzione, non sarà possibile fornire dettagli specifici per motivi di privacy, né allegare foto della stazione o del componente, le foto che verranno inserite sono reperite online e sono di componenti simili.

Uno statore è un componente dell'alternatore¹, in particolare è formato da:

- un pacco statorico in alluminio a lamelle con apposite cave;
- un avvolgimento plurifilare, a più fasi, in filo di rame;
- una serie di zeppe isolanti nelle cave del pacco statorico;

ed è la parte statica dell'alternatore, al suo interno ha sede il rotore che, invece, è la parte mobile.



Una volta che lo statore viene avvolto da macchine specifiche e l'avvolgimento assestato, di norma esso viene poi resinato e ricotto; a questo punto, una volta raffreddato, deve essere spazzolato.

¹ L'Alternatore è una macchina elettrica in grado di lavorare sia come motore che come generatore, all'interno di un'automobile è montato sulla cinghia di distribuzione. È composto da supporti di ancoraggio, statore, rotore, puleggia e ponte raddrizzatore con regolatore. È in grado di trasformare l'energia meccanica del motore endotermico di un'automobile in energia elettrica per il funzionamento della macchina.

La macchina in questione, all'interno dello stabilimento, nasce proprio a questo scopo e prevede la spazzolatura del pezzo sia esternamente che internamente. Essa in origine nasceva come stazione singola caricata manualmente da un operatore, negli anni, con l'incalzante processo di automatizzazione intrapreso dall'azienda, è stata provvista di un'automazione esterna per l'asservimento automatico e, quindi, inquadrata all'interno di una catena di montaggio. L'automazione è formata da un buffer di carico e uno di scarico motorizzati, un robot con la funzione di "pick&place" ed un buffer di scarto non motorizzato; essa, in ogni caso, non esclude la possibilità di utilizzare la stazione singolarmente come era stata pensata originariamente, soprattutto allo scopo di ripassare i semilavorati.

La stazione è controllata da un PLC² Siemens S7 attraverso un software scritto con un linguaggio di programmazione noto come AWL.

Il Retrofit a cui sarà sottoposta ha lo scopo di introdurre funzionalità di controllo sulla spazzolatura del pezzo e di riportare la macchina all'interno del capitolato aziendale, che prevede l'utilizzo del solo linguaggio di programmazione KOP o LADDER; il capitolato, inoltre, impone uno standard sulla struttura del programma che analizzeremo più avanti nel corso della trattazione.

Le motivazioni alla base di questa richiesta sono legate alla qualità del prodotto che l'azienda vuole garantire, attualmente la stazione produce degli scarti di lavorazione legati alla mancanza di controllo sull'usura della mola.

² Un PLC è un Controllore Logico Programmabile usato ampiamente in ambito industriale per la gestione automatizzata di macchinari poiché è uno strumento solido efficace ed efficiente, formato da una CPU per la gestione del programma e una serie di schede di ingressi e di uscite per l'acquisizione dello stato della macchina e la movimentazione delle componenti mobili della stessa.

Un'altra motivazione è da ricercarsi nel risparmio economico che si otterrebbe monitorando lo stato della mola in modo da cercare di allungare la sua vita lavorativa, ad oggi il consumo delle mole è dell'ordine di una al giorno. È intenzione dell'azienda, appunto, cercare di sfruttare meglio questi dispositivi diminuendo la frequenza con cui vengono sostituiti.

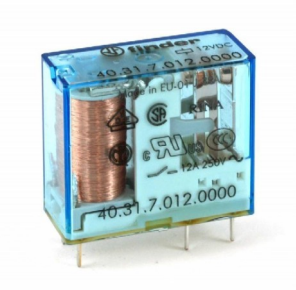
Capitolo I

Il PLC

In questo capitolo si proporrà una panoramica del PLC, dalla sua prima comparsa nel mondo dell'automazione industriale, cercando di analizzare le caratteristiche di funzionamento, capire quali sono i componenti di cui è formato e i vari linguaggi di programmazione comparsi nel corso del tempo. Per la dissertazione si citano le fonti [2], [3],(1),(2)

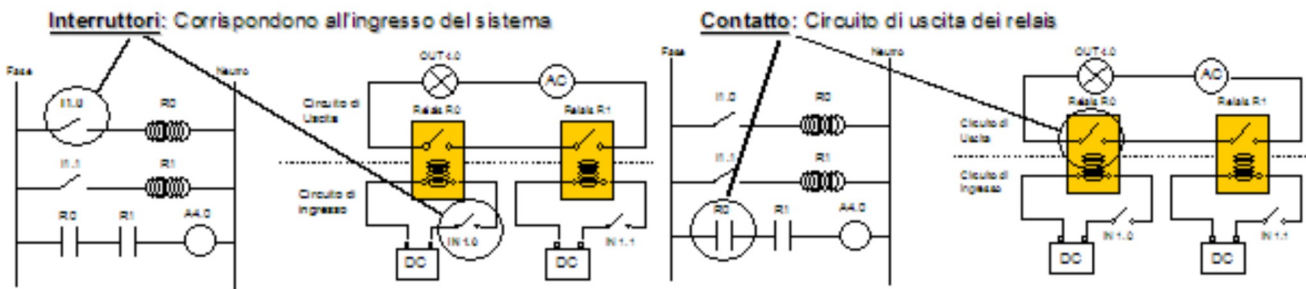
UN PO' DI STORIA

L'elemento su cui si basava la realizzazione della logica nei sistemi automatici, prima degli anni '60, era il "relè".

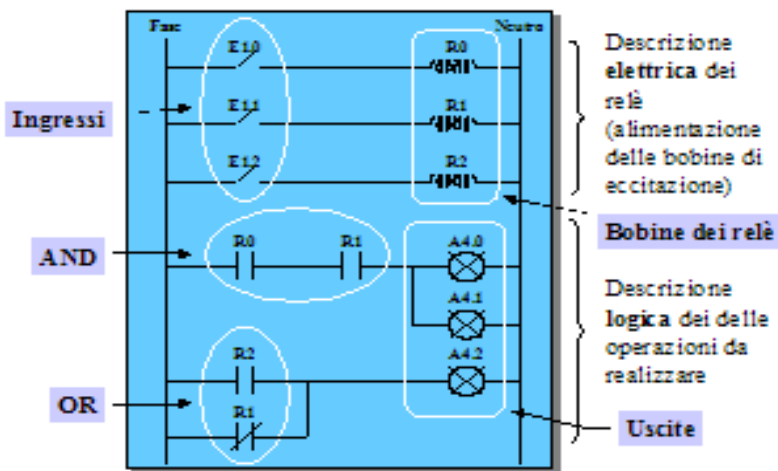


Le loro dimensioni, di certo non ridottissime, facevano sì che gran parte dei controlli per l'automazione industriale richiedessero, per essere implementati, l'utilizzo di interi armadi.

Di seguito un esempio di cablaggio impianti a relè:



Controllori cablati: *Quadri a relais*



La forza di questa schematizzazione sta nel riuscire ad evidenziare la logica delle operazioni, pur utilizzando notazioni di immediata interpretazione elettrica e quindi facilmente comprensibili per un qualsiasi tecnico. Il diagramma precedente va quindi interpretato proprio come un circuito elettrico, in cui la tensione d'alimentazione è applicata tra i due nodi principali del circuito, contrassegnati con "Fase" e "Neutro", dai quali dipartono una serie di rami o "pioli".

Dalle associazioni appena presentate, si evince quanto segue:

- Gli interruttori sono utilizzati per simulare lo stato degli ingressi.

- I “condensatori” sono utilizzati per rappresentare i contatti dei relè.
- Le sigle alfanumeriche associate a condensatori e bobine, hanno lo scopo di segnalare le reali corrispondenze tra questi due aspetti di un unico elemento fisico: il relè.
- Le bobine rappresentano, dal punto di vista logico, degli elementi in grado di memorizzare il passaggio di corrente, attraverso il posizionamento di un interruttore.
- Le lampadine stanno a segnalare le uscite, o le periferiche a loro collegate, e saranno da considerare attivate solo nel caso in cui siano attraversate da corrente.

È facile intuire come l'utilizzo dei controllori a relè richiedesse l'accettazione implicita di tutta una serie di grossi inconvenienti:

Debugging difficoltosi: soprattutto nei sistemi di una certa complessità, riuscire ad individuare un relè rotto, o mal collegato, all'interno di un enorme circuito contenuto in un armadio di sei metri, può non essere un'operazione semplice da fare.

Manutenzioni onerose: legata all'utilizzo di una gran quantità di strumenti elettromeccanici, sensibili, per loro stessa natura, all'usura ed all'invecchiamento.

Ingombro elevato: per quanto in origine i relè fossero considerati dispositivi di ridotte dimensioni, la necessità di controlli sempre più complessi nel campo dell'automazione industriale, ha portato gli impianti di maggiore dimensione a misurarsi con la necessità di realizzare circuiti fisicamente molto grandi.

Consumo elevato: l'alimentazione della bobina di eccitazione di un relè comporta consumi non indifferenti.

DAL RELE' AL PLC

Lo sviluppo dei PLC iniziò nel 1968 in risposta alle necessità della General Motors di ridurre gli alti costi di cablaggio. Le specifiche dei controlli GM richiedevano sistemi a stato solido con la flessibilità di un computer, ma che potessero essere programmati e mantenuti dagli ingegneri e tecnici dello stabilimento. Tali apparecchiature dovevano, inoltre, resistere alla sporcizia presente nell'aria, alle vibrazioni, alle interferenze elettriche, umidità ed a temperature estreme, condizioni tipiche dell'ambiente industriale.

I primi PLC vennero installati nel 1969 ed ebbero subito un grande successo, la possibilità di riprogrammarli ne aumentò considerevolmente la flessibilità al momento della modifica degli schemi di controllo.

I controllori logici programmabili (PLC) sono, quindi, dei dispositivi elettronici che comandano macchine e processi. Sono dotati di memoria programmabile per la memorizzazione delle istruzioni e l'esecuzione di funzioni specifiche, quali il comando di attivazione e disattivazione, la temporizzazione, i conteggi, le funzioni sequenziali e matematiche, e la gestione dei dati.

Forse la caratteristica che più di tutte favorì l'introduzione dei PLC nell'industria fu il fatto che il linguaggio di programmazione iniziale era basato sugli schemi Ladder e sui simboli elettrici comunemente usati dagli elettricisti. La logica Ladder riveste a tutt'oggi un ruolo essenziale nella programmazione e nell'individuazione dei problemi, anche se nel frattempo sono stati sviluppati linguaggi di programmazione più "avanzati". Sostanzialmente un PLC può, nel caso più semplice, essere assimilato ad un dispositivo che, leggendo un idoneo programma dalla memoria di cui è dotato ed eseguendo le istruzioni in esso contenute, controlla lo stato dei

suoi ingressi e rende attive o meno le uscite di cui è fornito. Normalmente agli ingressi sono connessi apparati di comando come interruttori, deviatori, sensori, finecorsa, dispositivo fotoelettrici, interruttori induttivi ecc., e alle uscite vari tipi di attuatori come motori, connessi al PLC per mezzo di teleruttori o amplificatori di potenza allo stato solido, relè, lampade.

All'interno di un PLC sono presenti sostanzialmente gli stessi dispositivi contenuti in un computer: CPU, memoria RAM ed EPROM, interfacce di I/O. Il Controllore Programmabile ha, però, caratteristiche sia strutturali che funzionali completamente diverse. Esso presenta sempre una struttura compatta, robusta e facilmente maneggevole.

Sostanziali sono anche le differenze dal punto di vista funzionale: un PLC svolge ciclicamente sempre lo stesso programma, ovvero arrivato all'ultima istruzione torna a svolgere la prima. Almeno fin quando non viene variata l'applicazione per cui è utilizzato. Un PLC non deve eseguire complesse elaborazioni su grosse quantità di dati ma svolgere controlli sullo stato dei segnali presenti ai suoi ingressi, effettuare elaborazioni di tipo logico o matematico elementari, ed infine attivare o meno alcune uscite portandole a livello alto o basso in base al programma in esecuzione. Poiché i PLC vengono utilizzati essenzialmente in ambienti industriali, i livelli logici dei segnali d'ingresso e d'uscita non sono del tipo TTL o CMOS, ma ben diversi (in genere 0 / 24V) per avere alta immunità ai disturbi.

Normalmente il set di istruzioni disponibile non è molto vasto. È da precisare poi che Controllori Logici Programmabili realizzati da case

costruttrici diverse hanno metodi di programmazione diversi anche se sostanzialmente simili.

Oltre ai risparmi di denaro, i PLC forniscono molti altri benefici:

Affidabilità: Una volta scritto e messo a punto, un programma può essere facilmente trasferito e caricato in altri PLC. In questo modo si riducono i tempi di programmazione e di messa a punto, e si aumenta l'affidabilità. Con la logica interamente ospitata nella memoria dei PLC, non si corre il rischio di fare errori logici di cablaggio. Gli unici collegamenti necessari sono quelli dell'alimentazione, degli ingressi e delle uscite.

Flessibilità: Le modifiche ai programmi possono essere effettuate semplicemente battendo un paio di tasti su una tastiera. Gli utenti finali possono modificare il programma nello stabilimento oppure, con la stessa semplicità, il produttore può impedire che gli utenti manomettano i programmi, una funzione di sicurezza molto importante.

Funzioni avanzate: I PLC possono effettuare una grande varietà di comandi, da un'azione semplice e ripetitiva a complesse manipolazioni di dati. La standardizzazione dei PLC apre molte porte ai progettisti e semplifica la vita al personale di manutenzione.

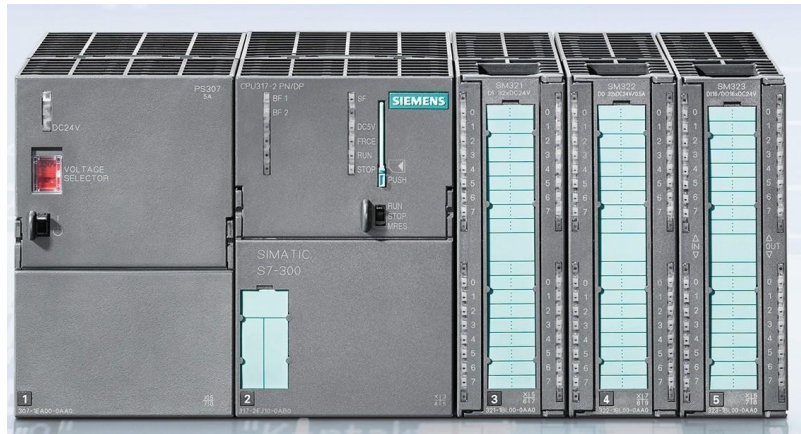
Comunicazioni: La comunicazione con interfacce dell'operatore, altri PLC o altri computer facilita la raccolta dei dati e lo scambio di informazioni.

Rapidità: Poiché alcuni macchinari automatizzati lavorano migliaia di articoli al minuto, e gli oggetti sostano solo per una frazione di secondo davanti ad un sensore, molte applicazioni di automazione richiedono l'installazione dei PLC per la loro rapidità di risposta.

Diagnostica: Le funzioni di individuazione dei problemi dei dispositivi di programmazione e la diagnostica incorporata nei PLC consentono agli utenti di identificare e di risolvere facilmente i problemi software e hardware.

Di seguito alcune immagini di PLC moderni:

S7-300 SIEMENS



MITSHUBISHI serie Q



NX7 OMRON



TRATTAZIONE GENERALI DEI COMPONENTI:

Tutti i PLC, dai micro ai grandissimi, sono dotati degli stessi componenti di base e hanno strutture simili. I sistemi PLC sono composti da:

- INGRESSI
- USCITE
- CPU
- MEMORIA, PER LA CONSERVAZIONE DI PROGRAMMI E DATI
- ALIMENTATORE
- INTERFACCIA OPERATORE

INGRESSI

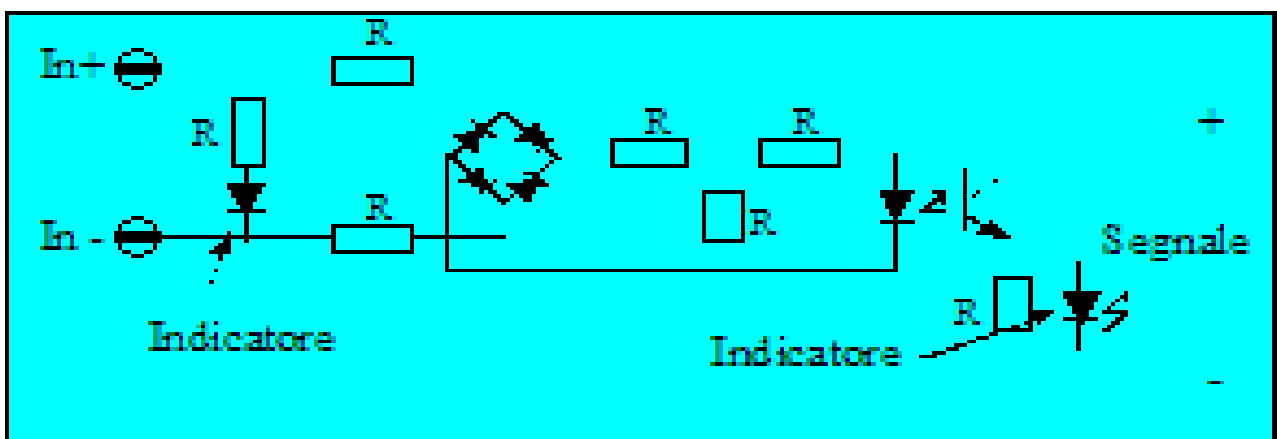
Gli ingressi includono componenti quali i pulsanti, i selettori rotativi, gli interruttori di finecorsa, i selettori manuali, i sensori di prossimità e i sensori fotoelettrici. Attraverso schede ingresso analogiche sono in grado di accettare valori analogici, segnali di tensione o di corrente variabili, come ad esempio da sensori di temperatura o di pressione.

I segnali elettrici che i dispositivi di campo inviano al PLC sono normalmente a 24Vc.c. (corrente continua) non filtrati. I circuiti di ingresso del PLC "manipolano" queste tensioni di campo in modo che il PLC le possa utilizzare. Tale conversione è necessaria in quanto i componenti

interni del PLC funzionano a 5Vc.c, il che li protegge dagli sbalzi di tensione riducendo i rischi di danni. Per isolare elettricamente i componenti interni dai morsetti di ingresso, i PLC sono dotati di isolatori ottici che utilizzano la luce per accoppiare i segnali da un dispositivo elettrico all'altro.

I circuiti di ingresso del PLC "filtrano", inoltre, i segnali di campo per determinare se sono validi, come ad esempio i segnali provenienti da un sensore, o se non lo sono, come i disturbi elettrici ad alta frequenza o dovuti alla statica. I filtri degli ingressi determinano la validità del segnale in base alla sua durata; in altre parole, "aspettano" prima di confermare la provenienza del segnale da un dispositivo di ingresso piuttosto che da un disturbo elettrico. L'intervallo tipico di filtraggio è di 8 ms, ma in alcuni PLC tale intervallo può essere regolato.

Esempio di un tipico circuito di **OPTOISOLAMENTO** utilizzato in ingresso ai PLC



Il vantaggio di questo circuito sta nel riuscire a trasferire un segnale dall'ingresso del PLC, alla CPU interna, senza trasmettere tensione, in isolamento galvanico. Questo è un grosso vantaggio, perché, in questo modo, si isola la CPU da eventuali sovraccarichi di corrente e/o di tensione, si realizza quindi il migliore isolamento possibile.

USCITE

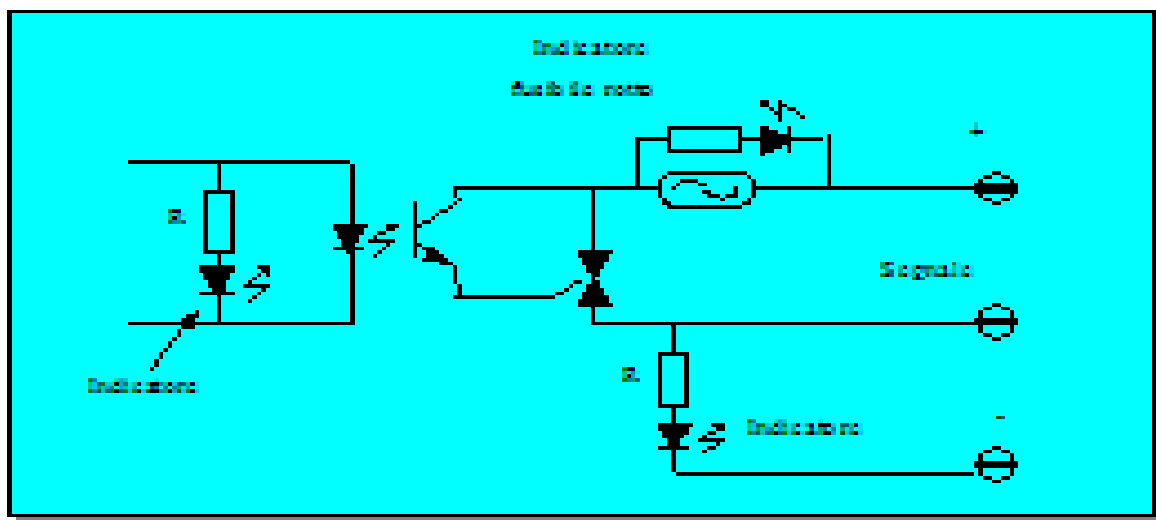
Ai morsetti di uscita del PLC sono collegati dei dispositivi quali solenoidi, relè, contattori, avviatori, spie luminose, valvole e allarmi.

I circuiti d'uscita funzionano in modo analogo a quelli d'ingresso: i segnali dalla CPU passano attraverso una barriera isolante prima di eccitare i circuiti di uscita.

I PLC utilizzano diversi tipi di circuiti per dare energia ai propri morsetti d'uscita:

- relè, usati per corrente alternata o continua. La portata dei relè elettromagnetici per PLC tradizionali è normalmente di pochi ampere. Tuttavia sono relativamente lenti e soggetti ad un'usura nel tempo.
- transistor, per commutare la corrente continua, sono silenziosi e non contengono parti mobili soggetti ad usura. Inoltre sono veloci e possono ridurre i tempi di risposta, ma sono in grado di sopportare solo carichi di 0.5 A o meno. Tipi speciali di transistor, quali FET possono sopportare potenze superiori, normalmente fino a 1A.
- triac, prettamente per lo scopo di commutare corrente alternata. Analogamente ai transistor le uscite triac sono silenziose, non contengono parti mobili soggette ad usura, sono veloci e sono in grado di sopportare carichi di 0.5 A o meno.

Schema di optoisolamento in uscita dal PLC.



Per quanto riguarda le uscite invece, è certamente indispensabile che le relative schede prevedano una protezione contro i corto-circuiti.

CPU: UNITÀ CENTRALE DI ELABORAZIONE

La CPU, che consta di un microprocessore e di un sistema di memoria, rappresenta il componente principale dei PLC. La CPU legge i segnali in ingresso, esegue la logica indicata dal programma applicativo, esegue i calcoli e controlla le uscite di conseguenza. Gli utenti dei PLC lavorano con due aree della CPU: i file dei programmi e i file dei dati. I file dei programmi contengono i programmi applicativi dell'utente, i file delle subroutine e il file di errore. I file dati contengono i dati relativi al programma, quali lo stato degli I/O, i valori preimpostati ed accumulati dei contatori e dei temporizzatori nonché altre costanti e variabili memorizzate. Collettivamente, queste due aree prendono il nome di memoria applicativa o memoria dell'utente. Nella CPU si trova anche un programma esecutivo, o memoria del sistema, che impartisce comandi ed esegue "operazioni"

quali l'esecuzione del programma dell'utente e la coordinazione delle scansioni degli ingressi e degli aggiornamenti delle uscite. L'utente non può accedere alla memoria del sistema, che è programmata dal produttore.

I primi PLC avevano un set di istruzioni molto ridotto (AND, OR, NOT, IN, OUT, SET, RESET), al fine di consentire l'esecuzione di un'istruzione in un solo ciclo di clock; infatti, con queste semplici istruzioni è possibile implementare qualsiasi logica combinatoria o sequenziale. Il vantaggio di quest'approccio, era ovviamente quello di ottenere una grande efficienza di esecuzione ed elevate prestazioni. In questo contesto molti PLC utilizzavano CPU ad 1 bit, per ottenere migliori prestazioni.

Le CPU a più bit diventano indispensabili in tutti i casi in cui c'è la necessità di:

- **Elaborare operandi non binari:** operazioni aritmetiche e di confronto, per variabili non booleane.
- **Introdurre salti di programma.**
- **Realizzare conteggi.**
- **Gestire ingressi e/o uscite di tipo puramente analogico.**

I controllori a logica programmabile più moderni implementano l'uso delle "subroutine": blocco funzionale che realizza una funzione logica e che può essere richiamato ovunque serva nel programma.

TEMPORIZZATORI E CONTATORI

I timer interni svolgono funzioni fondamentali, permettendo al programmatore di temporizzare il verificarsi di determinati eventi. I

temporizzatori, tutti del tipo non ritentivo, hanno indirizzi nel campo 0 – 127.

È importante notare che una volta attivato il temporizzatore, il conteggio procede in modo asincrono rispetto all'esecuzione del programma. Ovvero le istruzioni vengono svolte una dopo l'altra fino alla fine per poi ripartire dalla prima mentre il tempo scorre. Quindi, il temporizzatore si comporta come un elemento indipendente che una volta avviato fa scorrere il tempo fino al valore impostato, se non intervengono elementi che lo azzerano.

Le Temporizzazioni ed i Conteggi sono gestite mediante delle strutture che contengono sostanzialmente 3 tipi di informazione:

uno Stato (valore ad 1 bit) che indica se il tempo od il conteggio sono scaduti (1) oppure no (0), un Valore corrente (word) che contiene il tempo trascorso dall'abilitazione o i conteggi effettuati ed il Setpoint (word), cioè il valore impostato. Quando esso viene raggiunto dal valore corrente lo stato va a 1.

DATI, MEMORIA E INDIRIZZAMENTO

Mentre la memoria è uno spazio fisico i dati sono le informazioni conservate in tale spazio. La CPU funziona esattamente come un calcolatore, in quanto manipola i dati per mezzo di cifre binarie, ossia i bit. Un bit è una posizione discreta entro un chip di silicone nella quale o la tensione è presente, nel qual caso il valore rilevato è 1 (eccitato), o non è presente, per cui il valore è 0 (diseccitato).

Il bit è l'unità di memoria più piccola a disposizione. Generalmente, le CPU elaborano e memorizzano i dati in gruppi di 16 bit, comunemente chiamati "parole" o "word".

Ciascuna parola di dati ha una posizione fisica specifica nella CPU, chiamata "indirizzo" o "registro". Ad ogni elemento nel programma dell'utente viene fatto riferimento mediante un indirizzo, che indica dove vengono memorizzati i dati di quell'elemento.

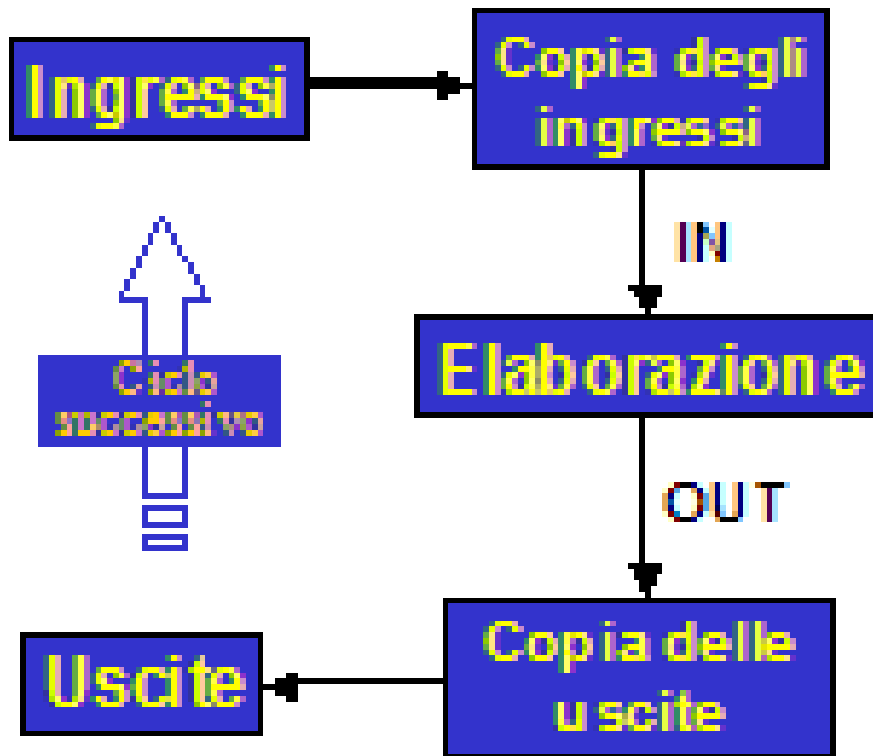
Quando si assegnano gli indirizzi agli I/O in un programma, bisogna tenere a mente che tali indirizzi sono relativi ai morsetti ai quali i dispositivi di ingresso o di uscita sono collegati.

CICLO OPERATIVO

Tutti i componenti del sistema PLC entrano in azione durante il ciclo operativo, che consiste in una serie di operazioni eseguite in sequenza e ripetitivamente.

Gli elementi principali del ciclo operativo sono i seguenti:

1. *Scansione degli ingressi.* Durante la scansione degli ingressi, il PLC esamina i dispositivi di ingresso esterni per controllare se vi è tensione o meno, ossia controllare se i circuiti sono aperti o chiusi. Lo stato degli ingressi viene memorizzato temporaneamente in un file di memoria di "immagine degli ingressi".
2. *Scansione del Programma.* Durante la scansione del programma, il PLC legge le istruzioni nel programma in logica Ladder, usa lo stato degli ingressi contenuto nel relativo file immagine e determina quali uscite verranno eccitate e quali no. Il conseguente stato delle uscite viene scritto nel file di memoria di "immagine delle uscite".
3. *Scansione delle Uscite.* Il PLC eccita o diseccita i suoi circuiti d'uscita, in base ai dati contenuti nei file delle uscite, controllando così i dispositivi esterni.



ALIMENTATORI

L'alimentatore fornisce l'energia ai componenti elettronici all'interno del controllore, converte la tensione in entrata in modo che possa essere utilizzata e protegge i componenti del PLC dagli sbalzi di tensione. Si deve considerare il fatto che la maggior parte degli stabilimenti sono soggetti a sbalzi di tensione. Gli alimentatori del PLC sono progettati in modo da non far alterare il funzionamento del dispositivo anche quando la tensione varia dal 10 al 15%.

INTERFACCE OPERATORE

Per poter inviare i dati relativi allo stato del macchinario, il pannello anteriore del PLC è dotato di una serie di spie luminose che indicano l'alimentazione, il funzionamento, eventuali guasti o lo stato degli I/O. Per comunicare con il PLC, ossia per immettere dei dati oppure per seguire e controllare lo stato della macchina, le interfacce operatore tradizionali includono dei pulsanti, dei selettori rotativi, delle spie e dei visualizzatori numerici a LED.

Per migliorare l'interfaccia fra l'operatore e il PLC, è stata creata una nuova generazione di dispositivi elettronici di interfaccia operatore (o periferiche). Non si tratta di dispositivi di programmazione ma di visualizzatori grafici o alfanumerici e di pannelli di comando.

ASSEGNAZIONE DEGLI I/O ESTERNI

Il modo in cui gli I/O esterni sono assortiti e debbono essere pilotati può essere molto diverso. Ad esempio si possono avere degli I/O, raggruppati in un rack, pilotati ognuno da una propria scheda oppure degli I/O, distribuiti lungo un Fieldbus, e pilotati da un'unica scheda di interfaccia, oppure, ancora, si può avere un mix di queste due situazioni.

È compito dello "I/O Driver" gestire l'immagine Virtuale ed assegnarla ai dispositivi fisici. Sebbene le combinazioni possano essere molteplici l'indirizzo Virtuale può' essere univocamente assegnato ad un Indirizzo fisico utilizzando 2 sole informazioni aggiuntive e ponendo la limitazione di trattare gli I/O digitali a gruppi minimi di 16 Bit (1 word). Per ogni Word di I/O Virtuale è quindi possibile abbinare un'indicazione di Tipo e

un Indirizzo. I Tipi e gli Indirizzi leciti per ognuno di essi sono definiti, di volta in volta, dal I/O Driver utilizzato.

OPERAZIONE DI MEMORIZZAZIONE

Si possono eseguire tre operazioni di memorizzazione denominate di SET (S), RESET (R), e ASSEGNAZIONE (=). Con l'operazione di SET si assegna all'operando il valore 1 e con RESET il valore 0 e i valori assegnati agli operandi non vengono modificati anche se le condizioni di SET e RESET vengono a mancare. Con l'ASSEGNAZIONE (=) viene invece attribuito il valore 1 all'operando, dinamicamente fintanto che resta vera la condizione di assegnazione.

PROGRAMMAZIONE DEL PLC: STANDARD "IEC 1131-3"

Un altro grosso problema dei PLC, rimasto tuttora parzialmente irrisolto, è quello della loro standardizzazione; infatti, ogni produttore di PLC ha introdotto un marchio ed una tipologia proprietari (interpretabili solo da loro tipo di PLC). Per avere una significativa riduzione delle piattaforme in commercio bisogna attendere gli anni '90, anno in cui, attraverso lo standard "IEC 1131-3", si definirono dei linguaggi standard per la realizzazione dei programmi per PLC, attualmente molti PLC supportano tutti questi linguaggi:

Function block diagram (**FBD**) o diagrammi a blocchi funzionali.

Instruction list (**IL**) o lista di istruzioni (**AWL**).

Linguaggio **C**: si usa lo stesso standard impiegato anche nei PC, ma nei PLC è applicato solamente nei suoi fondamentali

Structured text (**ST**) o letterale strutturato o ancora testo strutturato.

Ladder diagram (**LD**) o **KOP**, o linguaggio a contatti: è una schematizzazione del linguaggio grafico originariamente utilizzato dai tecnici per progettare i “quadri a relè”.

Lo standard “IEC 1131-3” ha tracciato anche una descrizione di PLC, definendolo un sistema elettronico a funzionamento digitale, destinato all'uso in ambito industriale, che utilizza una memoria programmabile per l'archiviazione interna di istruzioni orientate all'utilizzatore, per l'implementazione di funzioni specifiche come quelle logiche.

LINGUAGGIO LADDER/KOP

Questo linguaggio di programmazione è sicuramente stato tra i primi ad essere introdotto e si può anche dire che sia nato assieme ai PLC, perché presenta una struttura che deriva direttamente da quella definita per i primi controllori programmabili. Il “Ladder/kop” fa uso della simbologia dei relè, ed è sicuramente il “*linguaggio base*” oggi più utilizzato.

Un programma in Ladder/kop viene costruito con una serie di contatti aperti o chiusi, in combinazione logica. Sulla base dello stato logico degli ingressi, ed in funzione della combinazione logica prevista, si determinano le attivazioni delle uscite. I motivi del successo di questo linguaggio sono da ricercare nella sua perfetta aderenza con la descrizione di una rete mediante logica a relè, caratteristica questa che in passato ha certamente incontrato il gradimento del personale abituato a lavorare con gli armadi a relè.

Il simbolo di contatto normalmente aperto rappresenta una condizione “vera” quando l'ingresso è presente o la cella di memoria corrispondente vale ‘1’.

Il simbolo di contatto normalmente chiuso rappresenta una condizione “vera” quando l'ingresso non è presente o la cella di memoria corrispondente vale ‘0’.

Il simbolo di uscita “setta” la cella di memoria corrispondente, cioè la pone ad ‘1’, se c'è almeno un percorso in cui sussiste la continuità circuitale nel “rung” (piolo) che lo contiene.

PECULIARITÀ CHE HANNO CONSENTITO AL PLC DI RESISTERE AL DILAGARE DEI PC

Nonostante i PC abbiano avuto, ed abbiano tuttora, capacità di calcolo sempre maggiori a prezzi sempre più bassi, permangono alcune barriere che impediscono la completa sostituzione dei PLC.

I limiti che impediscono ai PC di sostituire completamente i PLC sono i seguenti:

Inadeguatezza all'utilizzo in ambiente industriale. Non essendo nati per l'utilizzo in ambienti estremi, i PC si rivelano particolarmente vulnerabili a tutta una serie di sollecitazioni molto frequenti in ambiente industriale, quali ad esempio:

Escursioni termiche. Il ristretto range di temperature in cui è garantito il corretto funzionamento dei moderni PC, è forse il limite più grosso per l'applicazione di questi dispositivi in ambienti estremi. L'aumento della

potenza di calcolo procede, infatti, di pari passo con i problemi legati alla dissipazione del calore generato dai circuiti interni ai microprocessori.

Alimentazione di rete soggetta a brusche variazioni.

Vibrazioni meccaniche. Un esempio rappresentativo di come i comuni PC non siano adatti sopportare sollecitazioni di tipo meccanico, è l'hard disk. Un hard disk sottoposto alle vibrazioni meccaniche tipicamente presenti in ambiente industriale, rischia di bruciare progressivamente tutti i settori del suo disco, divenendo ben presto inutilizzabile.

Disturbi elettromagnetici di varia natura ed intensità. Un circuito elettrico è sempre potenzialmente vulnerabile ai disturbi di natura elettromagnetica, ma questa vulnerabilità è ancora più marcata nel caso dei PC commerciali. Il crescere della potenza di calcolo, ha portato i microprocessori di questi dispositivi a lavorare con delle tensioni sempre più basse, al punto tale da essere divenute ormai paragonabili con quelle che potrebbero essere indotte, nonostante la presenza di eventuali schermature, dai forti campi magnetici generati in ambiente industriale.

In aggiunta a quanto sopra, possiamo dire che un'altra grossa differenza tra PC e PLC risiede nel fatto che, mentre i primi lavorano su “parole” di uno o più byte, tutte le informazioni sono memorizzate ed elaborate sotto forma di stringhe di bit a lunghezza standard, i PLC lavorano su singoli bit, che rappresentano degli stati. Così come i relè potevano descrivere due soli stati: ‘0’ ed ‘1’; anche nei PLC tutte le elaborazioni sono “orientate al bit”.

In conclusione quindi, quando si parla di PLC ci si riferisce a dispositivi che, pur avendo una struttura logica simile a quella dei PC, sono nati, e si sono sviluppati, proprio per operare in piena efficienza anche in prossimità dei sistemi che vanno a controllare. Così anche i microprocessori utilizzati sono scelti tra quelli adatti al particolare tipo di applicazione cui sono

destinati; vale a dire che sono scelti tra quelli che ad esempio garantiscono una dissipazione del calore adeguata alle temperature che caratterizzano l'ambiente di lavoro: in quest'ambito è possibile trovare i famosissimi processori Intel '486', ormai obsoleti per i PC, ma ancora utilizzati dai PLC, perché realizzati con una densità d'integrazione che ne consenta l'impiego entro range di temperatura ambiente sufficientemente ampi.

Capitolo II

L'Automazione Esistente

La macchina presenta due stazioni, una di spazzolatura esterna e l'altra interna.

Le due stazioni sono del tutto speculari, ognuna è formata da:

- un mandrino con una pinza a tre griffe per la rotazione dello statore;
- una mola di spazzolatura;
- un cilindro oleodinamico per il movimento verticale della mola;
- un cilindro pneumatico per il movimento orizzontale della mola;
- un aspiratore per gli sfridi di alluminio;
- una centralina idraulica;
- due inverter³ per la gestione della rotazione del pezzo e della mola;
- riparistica;
- elettrovalvole⁴ pneumatiche bistabili e monostabili;

³ L'inverter è un convertitore statico di potenza in grado di modificare la frequenza di rotazione di un motore elettrico a piacimento, diversa da quella di rete.

⁴ L'elettrovalvola è un dispositivo meccanico di commutazione comandato in corrente per un attuttore movimentato con un fluido, in genere aria o olio. Ne esistono di varie tipologie distinte: per fluido, per metodo di attivazione, per funzionamento, per numero di vie e posizioni. Si dicono monostabili se una volta azionate tornano nella posizione precedente rimuovendo il comando, bistabili se per tornare indietro hanno bisogno di un comando diverso dal precedente.

- elettrovalvole idrauliche bistabili;
- sensoristica fotosensibile ed induttiva;
- barriere fotoelettriche;
- sistema di controllo di usura mola;

l'unica differenza tra le due stazioni è data dal tipo di lavorazione e quindi dalla tipologia di pinza usata per bloccare il pezzo, nella stazione per la spazzolatura esterna il pezzo viene bloccato dall'interno, nell'altra il pezzo verrà bloccato sull'esterno poiché deve essere spazzolato internamente.

L'automazione esterna si presenta alla vista come una gabbia in cui vi è posto un Robot che provvede a prelevare gli statori sul buffer di carico contenente i pezzi caldi usciti da un forno e depositarli nella stazione di spazzolatura esterna, dopo di che il pezzo verrà posto in quella di spazzolatura interna ed infine il robot depositerà il componente sul buffer di scarico che porterà i pezzi verso altre macchine della linea di produzione. Se il pezzo dovesse risultare di scarto durante la lavorazione esso verrà posto dal Robot sul buffer di scarto, qui sarà poi oggettivato e decretato scarto a rottame o rilavorabile.

L'automazione esterna è però gestita da un altro PLC, poiché arrivata in un secondo momento rispetto alla prima installazione della spazzolatrice, ed è stata quindi prevista la possibilità di escluderla per poter effettuare cicli di ricarica manuale dei pezzi semilavorati o degli scarti rilavorabili; dato che l'impianto originale era stato pensato come impianto singolo con caricamento manuale.

Ogni stazione è provvista di una finestra di carico, ai lati delle quali sono montate le due barriere fotoelettriche, con lo scopo di arrestare immediatamente ogni movimentazione se invase, per la sicurezza

dell'operatore. Queste barriere sono sempre attive durante il ciclo di spazzolatura ed inibite in fase di caricamento pezzi sia da parte del Robot che dell'eventuale operatore.

La lavorazione avviene tramite la movimentazione cartesiana della mola azionando il cilindro verticale e quello orizzontale in modo che la spazzola in rotazione sul proprio asse tocchi lo statore sul mandrino, in rotazione anch'esso, ed asporti il sedimento in eccesso della resina dovuto al processo precedente, facendo diverse passate che sono customizzabili in base alla tipologia di statore in lavorazione.

La stazione, inoltre, è provvista di porte di accesso per facilitarne la manutenzione, ogni porta è munita di blocca-porte elettrici gestiti da relè di sicurezza a doppio canale; un altro di questi relè è previsto per la gestione dei vari funghi di emergenza montati nei punti cruciali della macchina. Questi relè garantiscono l'accesso alle parti interne delle stazioni in totale sicurezza rimuovendo le possibili pressioni residue e garantendo che le parti rotanti gestite da motori con inverter non entrino accidentalmente in rotazione.

All'esterno dell'automazione esterna vi sono due pannelli operatore, uno per la gestione del Robot, l'altro per i movimenti manuali della spazzolatrice, la calibrazione dei trasduttori e la scelta delle passate in relazione alla ricetta in macchina.

Ad oggi la macchina lavora in maniera non controllata, ovvero la spazzolatura degli statori avviene senza il corretto monitoraggio della pressione di spazzolatura. Il cilindro orizzontale, infatti, avanza con una pressione limitata da un regolatore di pressione in ingresso tarato sempre allo stesso valore senza tener conto dell'usura della mola. Ciò porta ad un continuo cambio delle mole (circa una al giorno) poiché non vengono

sfruttate al meglio, inoltre, è frequente la presenza di scarti a rottame per sovraspazzolatura dell'alluminio del pacco statorico (con le mole nuove) e la presenza di scarti da riprocessare per sottospazzolatura (dovuta alle mole a fine vita). Un altro difetto riscontrato attualmente è dato dal fatto che, avendo la pressione di spazzolatura costante, la frequenza di rotazione del motore della mola è costante.

Il Retrofit ha lo scopo di ripristinare il sistema di controllo di spazzolatura attraverso il monitoraggio dell'usura della mola e garantendo a seconda dello stato di quest'ultima, che la velocità tangenziale di spazzolatura rimanga costante al variare del diametro. Ciò sarà possibile grazie all'impiego di schede PLC Analogiche per la gestione di un trasduttore lineare⁵, una cella di carico⁶ ed una valvola proporzionale⁷.

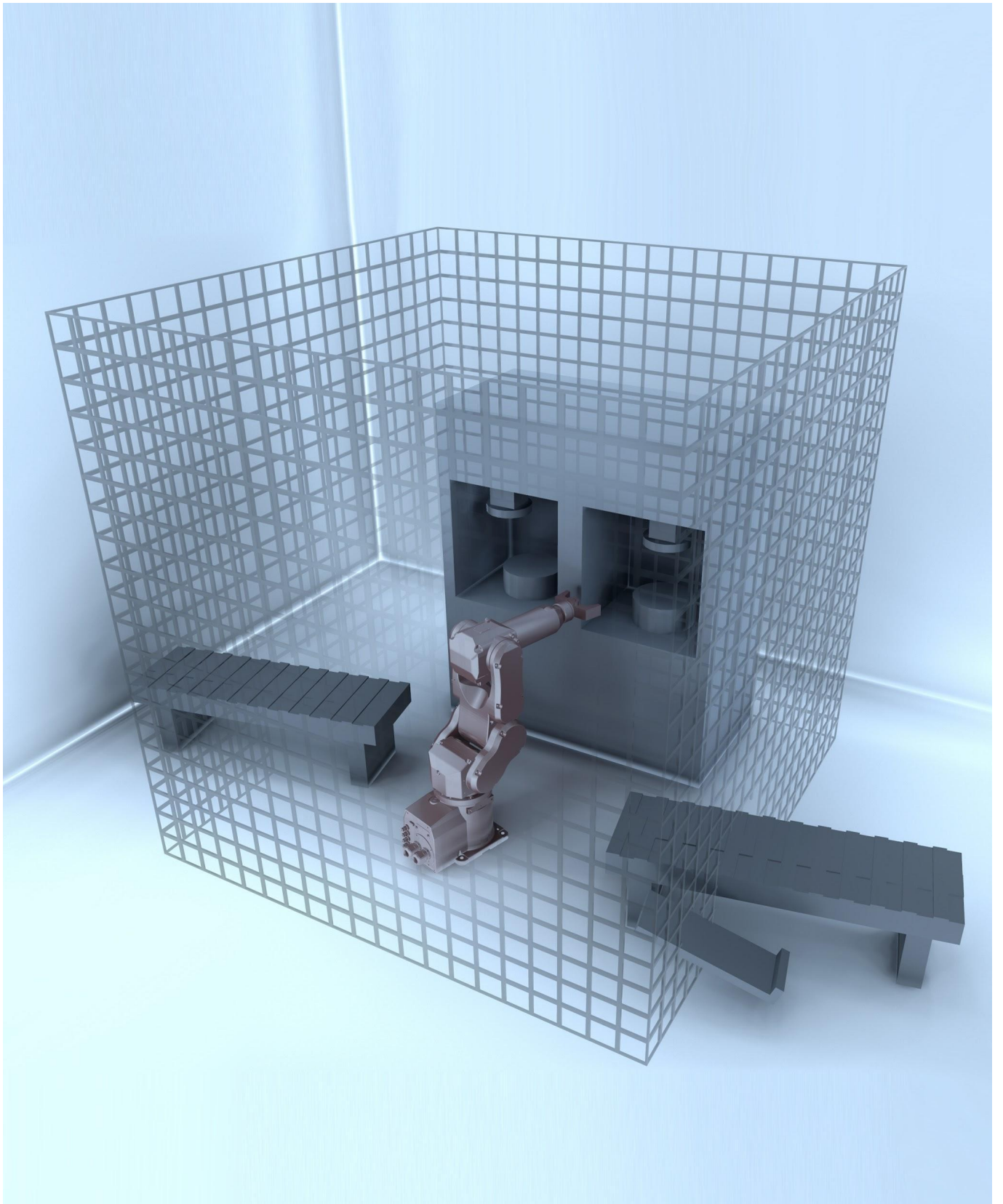
Il programma di controllo dovrà quindi acquisire, tramite la movimentazione orizzontale del cilindro di accostamento, l'escursione di quest'ultimo attraverso il trasduttore lineare a seconda di quando la spazzola abbia toccato il pezzo, informazione notificata dalla cella di carico che è montata sullo stesso cilindro, controllando quindi, la pressione dell'aria compressa per la sua movimentazione attraverso la valvola proporzionale.

Dal valore derivato dal trasduttore, poi, sarà possibile ottenere la misura del raggio della mola nel tempo e in questo modo gestire la frequenza di rotazione della mola stessa attraverso l'inverter dedicato e allo stesso tempo garantire una spazzolatura efficace poiché sarà possibile sfruttare la mola al meglio, fino alla massima usura.

5 Trasduttore in grado di convertire la propria escursione in un segnale analogico, corrente o tensione.

6 Trasduttore in grado di convertire il carico avvertito in un segnale analogico, corrente o tensione.

7 Valvola in grado di controllare la portata o la velocità di un cilindro fornendo al solenoide un segnale analogico di corrente.



Capitolo III

La Programmazione

In questo capitolo si analizzerà il programma di controllo della spazzolatrice sulla base delle istruzioni ricevute dall'azienda, quindi ci si rifarà al capitolato aziendale, a valle della revisione software e si confronteranno le modifiche al programma con la vecchia gestione, ove possibile e nel rispetto della privacy aziendale.

La Denso, infatti, in fase di acquisto di nuovi impianti impone delle regole alle ditte costruttrici di macchinari industriali, ma anche in questo caso, poiché si tratta di un retrofit di un impianto esistente, in fase di programmazione software bisogna far capo a quelle stesse regole.

L'Ambiente di Programmazione

Per la programmazione della macchina si è usato l'ambiente Simatic Manager di Siemens. Il programma sarà strutturato in una serie di sottoprogrammi che coesistono in esso e che hanno il compito di gestire ciascuno un aspetto del funzionamento della macchina.

Un sottoprogramma nell'ambiente Siemens può essere un FC, un FB o un OB ed avere dei richiami a delle DB che a loro volta possono essere strutturate tramite UDT.

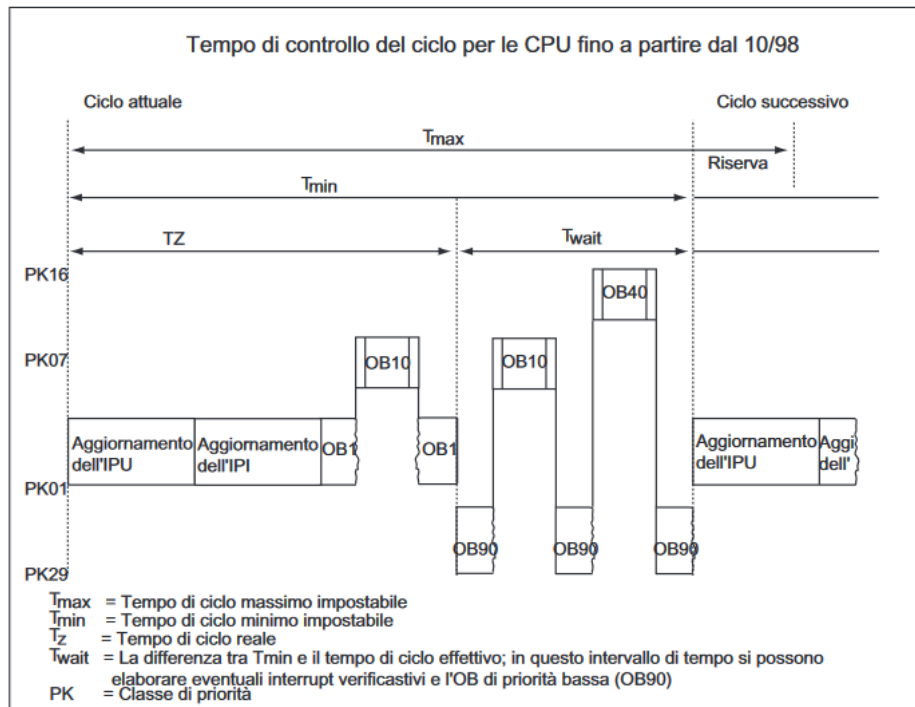
[1] **OB**, blocchi organizzativi, rappresentano l'interfaccia tra il sistema operativo e il programma utente. Essi vengono richiamati dal sistema operativo e comandano l'elaborazione ciclica del programma su interrupt, il comportamento di avvio del sistema di automazione e la gestione degli errori. Programmando i blocchi organizzativi è possibile determinare il comportamento della CPU. I blocchi organizzativi determinano la sequenza in cui verranno elaborate le singole parti del programma. L'elaborazione di un OB può essere interrotta dal richiamo di un altro OB. È la priorità a stabilire quale OB può essere interrotto da un altro OB, gli OB con priorità più alta interrompono quelli con priorità più bassa.

L'elaborazione ciclica del programma è l'elaborazione "normale" nei controllori programmabili. Il sistema operativo richiama ciclicamente l'OB 1 e con esso avvia l'elaborazione ciclica del programma utente.

Esecuzione dell'elaborazione ciclica del processo

La seguente tabella riporta le fasi dell'elaborazione ciclica del programma.

Fase	Esecuzione nelle CPU fino a 10/98	Esecuzione nelle CPU a partire da 10/98
1	Il sistema operativo avvia il tempo di controllo del ciclo.	Il sistema operativo avvia il tempo di controllo del ciclo.
2	La CPU legge lo stato degli ingressi nelle unità di ingresso, e aggiorna l'immagine di processo degli ingressi.	La CPU scrive i valori dall'immagine di processo delle uscite nelle unità di uscita.
3	La CPU elabora il programma utente ed esegue le operazioni specificate nel programma.	La CPU legge lo stato degli ingressi nelle unità d'ingresso, e aggiorna l'immagine di processo degli ingressi.
4	La CPU scrive nelle unità di uscita i valori dell'immagine di processo delle uscite.	La CPU elabora il programma utente ed esegue le operazioni specificate nel programma.
5	Alla fine di un ciclo il sistema operativo esegue i compiti previsti, p. es. caricamento e cancellazione di blocchi, ricezione e invio di dati globali.	Alla fine di un ciclo il sistema operativo esegue i compiti previsti, p. es. caricamento e cancellazione di blocchi, ricezione e invio di dati globali.
6	A questo punto, la CPU ritorna all'inizio del ciclo e avvia nuovamente il tempo di controllo del ciclo.	A questo punto, la CPU ritorna all'inizio del ciclo e avvia nuovamente il tempo di controllo del ciclo.



[1] FC, le funzioni fanno parte dei blocchi programmati dall'utente. La funzione è un blocco di codice "privo di memoria". Le variabili temporanee dell'FC vengono memorizzate nello stack dei dati locali e dopo l'elaborazione dell'FC, questi dati vanno perduti. Per la memorizzazione dei dati non temporanei, le funzioni possono utilizzare blocchi dati globali, DB. Poiché a una FC non è abbinata alcuna memoria, è necessario indicarne sempre i parametri attuali.

Una FC contiene un programma che viene eseguito ogni qualvolta che l'FC viene richiamata da un altro blocco di codice. Le funzioni possono essere utilizzate per:

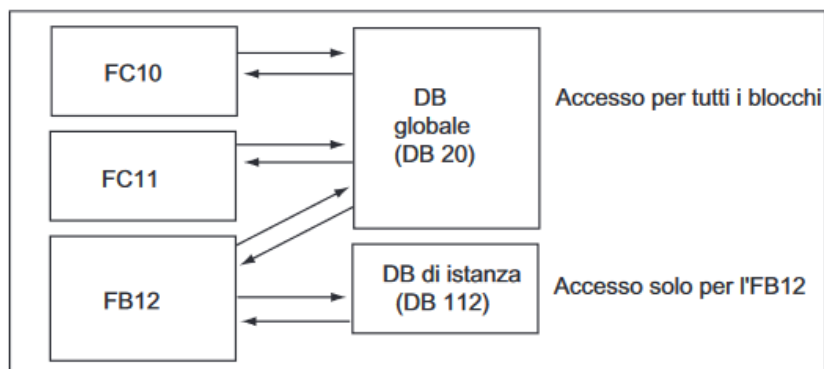
- restituire un valore di funzione al blocco richiamante (per esempio funzioni matematiche).
- eseguire una funzione tecnologica (per esempio controllo singolo con combinazione binaria).

[1] **FB**, blocchi funzionali, fanno parte dei blocchi programmati dall'utente. Un blocco funzionale è un blocco "con memoria". Esso dispone di un blocco dati correlato come memoria (blocco dati di istanza). Sia i parametri che vengono trasmessi all'FB, sia le variabili statiche vengono memorizzati nel blocco dati di istanza, mentre le variabili temporanee vengono memorizzate nello stack dei dati locali. Al termine dell'elaborazione dell'FB, i dati memorizzati nel DB di istanza non vanno perduti, come invece accade a quelli memorizzati nello stack dei dati locali. Un FB contiene un programma che viene eseguito ogni qualvolta che l'FB viene richiamato da un altro blocco di codice. A ogni richiamo di un blocco funzionale che trasferisce parametri viene assegnato un blocco dati di istanza. Nel DB di istanza vengono memorizzati i parametri attuali e i dati statici dell'FB. Le variabili dichiarate nell'FB determinano la struttura del blocco dati di istanza. Con il termine istanza si definisce il richiamo di un blocco funzionale. Per esempio, se un blocco funzionale viene richiamato cinque volte nel programma utente S7, di tale blocco esisteranno cinque istanze. Per poter creare un blocco dati di istanza, deve già esistere l'FB a cui il blocco deve essere assegnato.

Con il richiamo di diverse istanze di un FB si possono comandare più apparecchiature con un solo FB. Per esempio, usando dati di istanza differenti per i vari motori, l'FB per un solo tipo di motore può comandare più motori. I dati per ogni singolo motore (per esempio numero di giri, tempo di rampa, ore di funzionamento accumulate, ecc.) possono essere memorizzati in uno o più blocchi dati di istanza.

[1] **DB**, blocchi dati globali, a differenza dei blocchi di codice, i blocchi dati non contengono istruzioni STEP 7, ma servono alla registrazione dei dati utente. Nei blocchi dati sono quindi compresi i dati con cui opera il

programma utente. I blocchi dati globali servono alla registrazione di dati utente che possono essere usati da tutti gli altri blocchi. La dimensione dei DB è variabile e la struttura dei blocchi dati globali può essere stabilita liberamente. Quando un blocco di codice (FC, FB, OB) viene richiamato, esso può temporaneamente occupare spazio in memoria nell'area dei dati locali (L-Stack) oppure può aprire un'area di memoria sotto forma di un DB. Al contrario dei dati dell'area di dati locali, i dati contenuti in un DB non vengono cancellati quando si chiude il DB o al termine dell'elaborazione del relativo blocco di codice. Ogni FB, FC o OB ha accesso di lettura/scrittura a un DB globale. È possibile aprire contemporaneamente un DB globale e un DB di istanza. La figura seguente mostra i diversi accessi ai blocchi dati.



[1] **UDT**, tipi di dati definiti dall'utente, sono speciali strutture di dati che l'utente crea e può utilizzare in tutto il programma utente S7 secondo la definizione che ne ha dato.

- Gli UDT possono essere usati come tipi di dati semplici o tipi di dati composti nella dichiarazione di variabili dei blocchi di codice (FC, FB, OB), oppure come tipo di dati delle variabili di un blocco dati (DB). Ciò rappresenta un notevole vantaggio per l'utente, il quale, una volta definita una particolare struttura di dati utilizzabile più volte, la può assegnare a un numero infinito di variabili.

- Gli UDT possono essere usati come modello per creare blocchi dati con uguale struttura; è cioè possibile realizzare una volta la struttura e creare in seguito i blocchi dati semplicemente assegnando l'UDT (esempio "ricette": la struttura del DB resta uguale, cambiano solo le quantità indicate).

Architettura del Software

Per quanto riguarda la parte software il capitolato impone:

- L'utilizzo del solo linguaggio di programmazione noto come KOP/Ladder;
- L'utilizzo di una specifica nomenclatura per ogni sottoprogramma legato alla singola stazione;
- Una programmazione del ciclo di lavoro a gradini/step;
- Un utilizzo specifico dei nomi delle variabili usate nel programma;

Ed altre regole più rigide per una standardizzazione dei programmi, che però non è possibile analizzare nel dettaglio, per privacy.

Questi punti sono frutto di anni di sperimentazione in cui si è notato che nel far rispettare queste semplici regole, si hanno benefici nel lungo termine legati soprattutto alla ricerca guasto da parte dell'ente manutenzione. Si è notato, infatti, che con una standardizzazione su larga scala i tempi di identificazione dei guasti software scendono drasticamente, anche se si tratta del primo approccio ad un nuovo impianto. Questi benefici si tramutano poi in efficienza, poiché il tempo risparmiato sull'identificazione del guasto e quindi sul totale del fermo macchina per la risoluzione del guasto, va a sommarsi al tempo in cui la macchina è in produzione a

regime, per cui quell'efficienza viene trasferita all'impianto in termini di maggiore produttività.

Per quanto riguarda il secondo punto della lista precedente, per specifica nomenclatura, si intende che il programma avrà un numero N di sottoprogrammi numerati a seconda di quante stazioni formano l'impianto: ogni stazione ha a disposizione 10 sottoprogrammi numerati da 10 a 19 per la prima stazione, incrementando la decina per le successive (da 20 a 29 per la seconda e così via). Ad ogni numero corrisponde un nome specifico come ad esempio: Gestione Automatico o Gestione Manuale o Uscite o Pannello Operatore... in cui ha sede la gestione di ogni singola stazione in quella modalità di funzionamento o comunque viene gestito un componente specifico. I sottoprogrammi, poi, verranno richiamati di volta in volta dal PLC a seconda delle modalità di lavoro della macchina.

La prima fase della programmazione è stata la definizione dell'hardware installato per il controllo, partendo dal Rack, scegliendo quindi, l'alimentatore, la CPU le schede digitali ed analogiche ed infine definendo i due protocolli di comunicazione previsti: MPI e Profibus, indicando i componenti collegati con essi alla CPU scegliendo opportunamente gli indirizzi di questi sulla rete.

In profibus avviene la comunicazione con gli inverter, ovvero lo scambio tra la CPU e questi ultimi, delle command e status word per il loro controllo e diagnosi. In MPI, invece, avviene la comunicazione con l'automazione esterna della macchina attraverso una DB di interfaccia in cui, ad esempio il robot scambia le informazioni di fuori ingombro.

Il passaggio successivo alla configurazione hardware è la creazione della Tabella dei Simboli, essa contiene la definizione di tutte le variabili globali del progetto ed il relativo indirizzamento, agli ingressi fisici vengono

associate gli indirizzi E, alle uscite fisiche gli indirizzi A e alle merker gli indirizzi M rispettivamente. Ad ogni variabile è possibile apporre un commento, cura del programmatore è quella di essere il più fedele possibile allo schema elettrico e dettagliato nel commentare una variabile, in modo da rientrare nel capitolato come visto nel punto quattro della lista precedente.

Buona norma, è riportare sul commento della variabile:

- la tipologia di ingresso o uscita: pulsante, selettore, sensore, elettrovalvola, relè;
- riportare il nome presente sullo schema elettrico: PL, S, FC, EV... con la relativa numerazione utilizzata;
- fornire un'argomentazione chiara e concisa;

accorgimenti di questo tipo aiutano in fase di diagnosi e ricerca guasto ad una immediata individuazione del componente, riducendo i tempi di intervento in caso di anomalia.

La tabella dei simboli inoltre contiene la definizione anche delle subroutine del programma: FC, FB, DB o UDT; in essa, poi, ci sono anche gli eventuali contatori e temporizzatori utilizzati.

Il programmatore nella prima compilazione della tabella, quindi, potrà inserire gli ingressi e le uscite previste nello schema elettrico, dopodichè andando a scrivere il programma l'ambiente di programmazione permette di creare le variabili senza che siano state definite a priori, esse una volta indirizzate e commentate verranno inserite automaticamente nella Tabella dei simboli dall'software stesso.

Stilata la Tabella dei simboli si passa alla creazione dei Blocchi, ovvero si inseriscono tutti i sottoprogrammi necessari al funzionamento della macchina.

Nel caso specifico sono stati inseriti questi sottoprogrammi:

Nome oggetto	Nome simbolico	Linguaggio di progra...	Dimensione della me...	Tipo
Dati di sistema	---	---	---	SDB
OB1		KOP	52	Blocco organizzativo
FC10	Generale	KOP	516	Funzione
FC13	Pannello Operatore	KOP	38	Funzione
FC18	Anomalie	KOP	80	Funzione
FC19	Utenze Comuni	KOP	52	Funzione
FC21	Manuale Stazione A	KOP	230	Funzione
FC22	Automatico Stazione A	KOP	688	Funzione
FC26	Analogaica Stazione A	KOP	998	Funzione
FC27	Diagnostica Stazione A	KOP	106	Funzione
FC28	Anomalie Stazione A	KOP	538	Funzione
FC29	Utenze stazione A	KOP	348	Funzione
FC31	Manuale Stazione B	KOP	38	Funzione
FC32	Automatico Stazione B	KOP	38	Funzione
FC36	Analogaica Stazione B	KOP	38	Funzione
FC37	Diagnostica Stazione B	KOP	38	Funzione
FC38	Anomalie Stazione B	KOP	38	Funzione
FC39	Utenze stazione B	KOP	218	Funzione
FC50	DP_Ref_Freq_en	AWL	282	Funzione
FC51	Inverter 1	KOP	240	Funzione
FC52	Inverter 2	KOP	222	Funzione
FC53	Inverter 3	KOP	222	Funzione
FC54	Inverter 4	KOP	222	Funzione
FC105	SCALE	AWL	244	Funzione
FC106	UNSCALE	AWL	324	Funzione
FC110	Interfaccia MPI	KOP	510	Funzione
DB13	PLC <-> DP7B	DB	48	Blocco dati
DB16	DB Analogiche	DB	120	Blocco dati
DB28	Segnalazioni Stazione A	DB	52	Blocco dati
DB38	Segnalazioni Stazione B	DB	52	Blocco dati
DB51	PLC <-> Inverter 1	DB	50	Blocco dati
DB52	PLC <-> Inverter 2	DB	50	Blocco dati
DB53	PLC <-> Inverter 3	DB	50	Blocco dati
DB54	PLC <-> Inverter 4	DB	50	Blocco dati
DB55	Appoggi e Risultati	DB	284	Blocco dati
DB110	rete_MPI	DB	96	Blocco dati
UDT11	MM4_Control_Word_1	AWL	---	Tipo di dati
UDT33	MM4_Status_Word_1	AWL	---	Tipo di dati

Nelle FC dalla 10 alla 19 sono state gestite tutte le parti comuni alle due stazioni, come ad esempio l'avvio della centralina idraulica e definite le condizioni iniziali della macchina, come le condizioni di inizio ciclo e lo stato di Automatico o Manuale.

Dalla FC 21 alla 29 c'è la gestione della singola stazione di spazzolatura

interna in tutte le configurazioni di lavoro, dalla 31 alla 39 è stata gestita l'altra stazione.

L'FC 50 è un sottoprogramma Siemens in cui viene configurata la comunicazione con gli inverter, essa in particolare prende in input e in output i dati dalle DB 51, 52, 53 e 54 a loro volta strutturate secondo le UDT 11 e 33, questa FC viene richiamata nelle FC 51, 52, 53 e 54 in cui vi sono le condizioni di lavoro dell'inverter corrispondente. Sarà, quindi, compito della FC 50 inviare in profibus la command word e ricevere la status word e storicizzarle nelle rispettive DB a seconda delle condizioni imposte dal programmatore nelle FC in cui questa viene richiamata.

Le FC 105 e 106 sono funzioni di sistema, in quanto protette in lettura e scrittura, in grado di effettuare manipolazioni su grandezze analogiche.

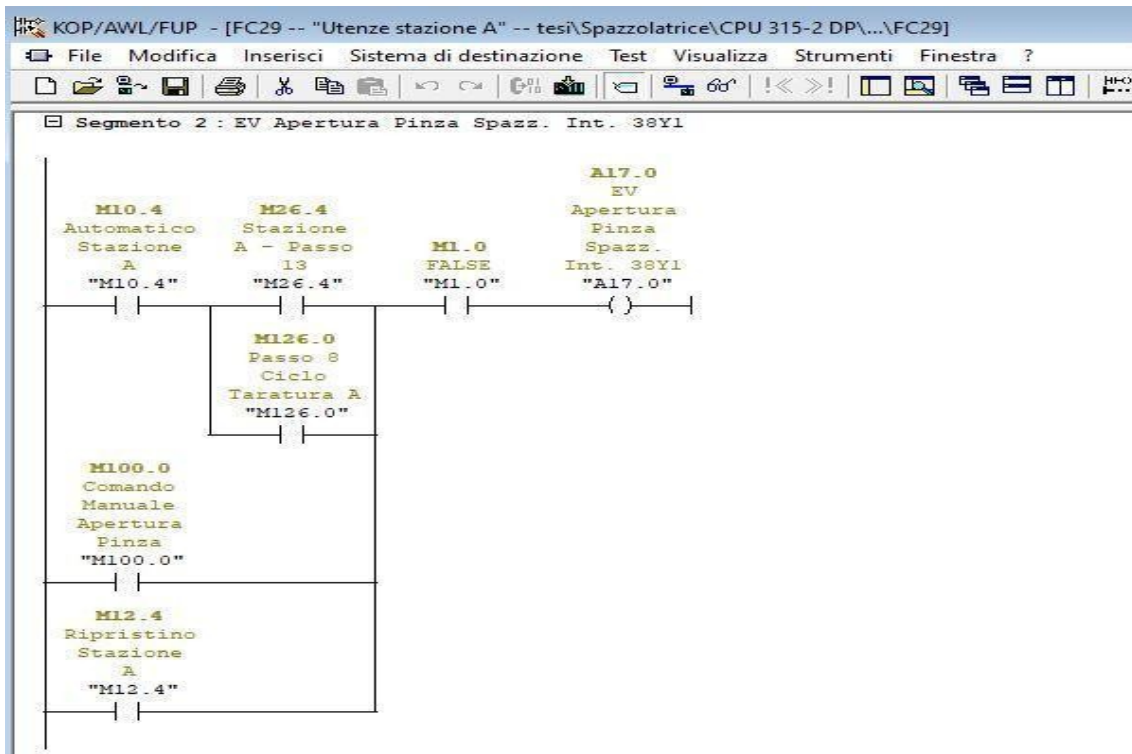
In particolare l'FC 105 "scale" prende in ingresso una grandezza analogica e la scala entro dei limiti massimi e minimi, definiti dal programmatore o acquisibili tramite calibrazione del segnale analogico di riferimento, ritornando il valore della grandezza stessa scalata sotto forma di "Real" storicizzabile ed utilizzabile nel programma. L'FC 106 "unscale", invece, si comporta al contrario. Quindi in questa applicazione con l'FC scale sono stati acquisiti il segnale del trasduttore e della cella di carico tramite schede di ingresso analogiche e appoggiati i valori limite e istantanei delle grandezze sulla DB 16, mentre con la FC "unscale" è stata comandata la valvola proporzionale tramite la scheda uscita analogica.

Nelle FC definite Utenze avviene l'assegnazione delle variabili di uscita (A), ovvero ci sono tutte le condizioni affinché una uscita vada dallo stato 0 allo stato 1 o viceversa. Questo può accadere in tre maniere:

- la condizione di chiamata in Automatico

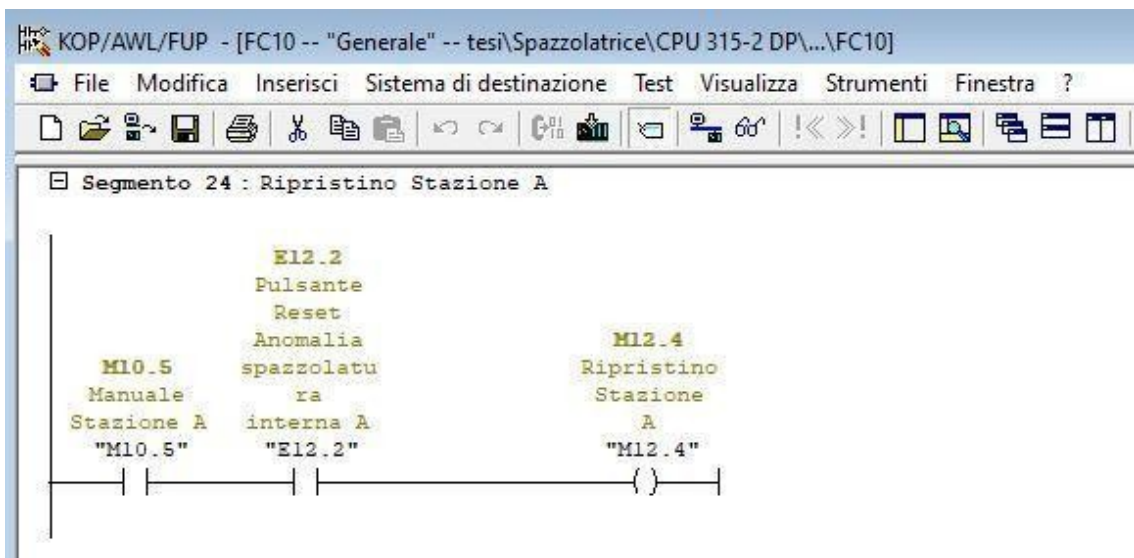
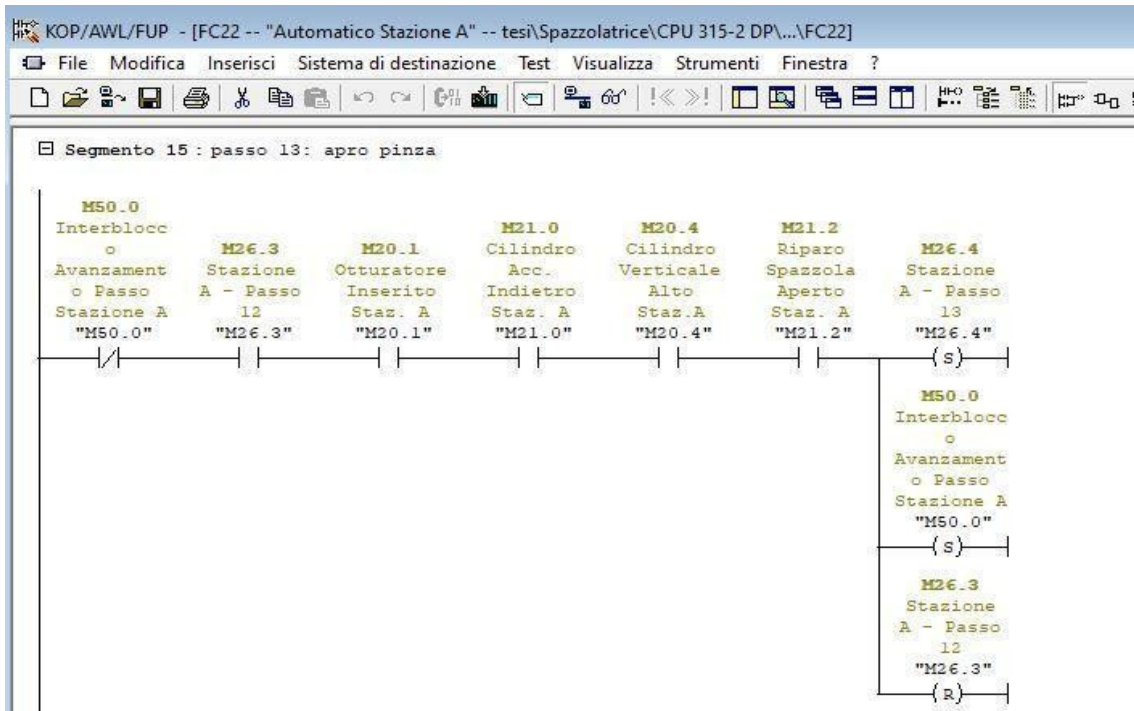
- la condizione di chiamata in Manuale
- la condizione di chiamata durante il Riposizionamento

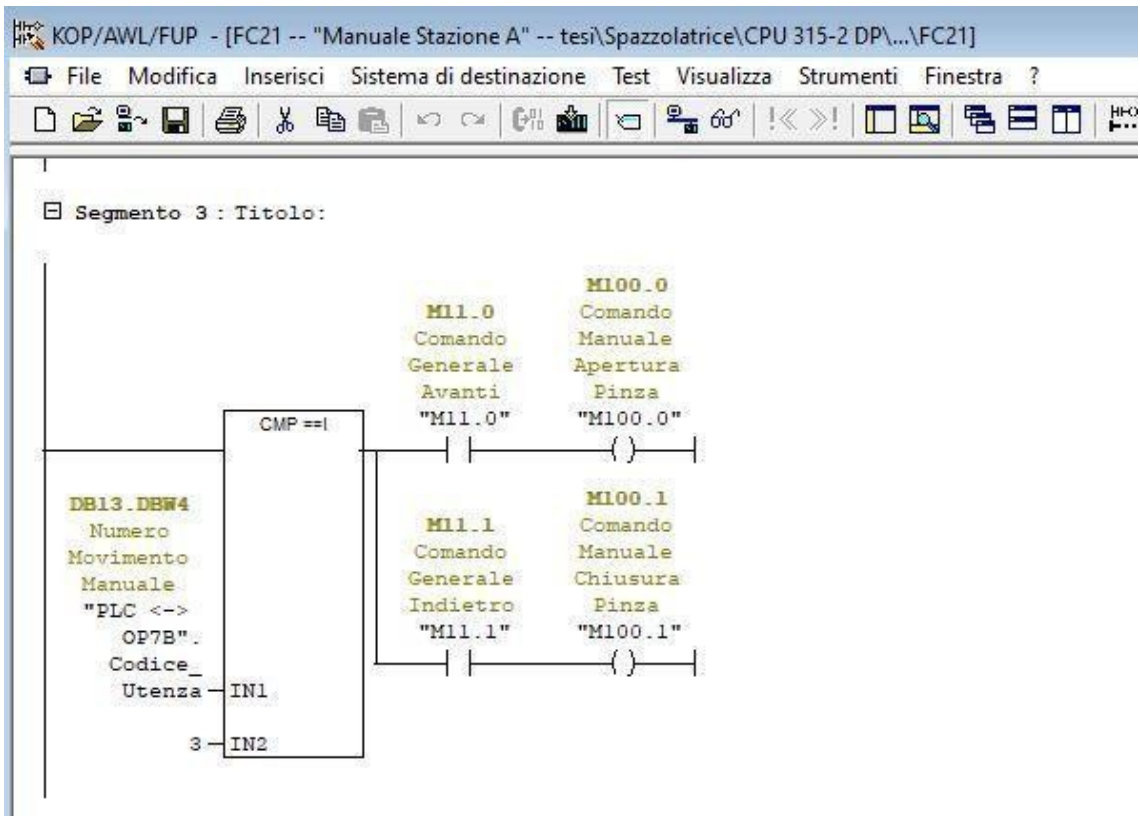
tutte programmate in modo da creare un OR logico, ossia la singola uscita può essere comandata solo da uno dei tre rami a seconda dello stato della macchina.



Dalla figura si evince come il verificarsi di uno dei tre rami dipenda da una merker specifica, l'assegnazione della quale avviene in altri sottoprogrammi; chiaramente la merker per l'automatico verrà comandata nell'FC 22 o 32 a seconda della stazione, quella per il manuale nell'FC 21 o 31, venendosi così a creare una matrioska che dai segnali fisici (gli ingressi) tramite condizioni logiche avveratesi in alcuni sottoprogrammi porta al cambio stato delle uscite in queste FC.

Per una migliore comprensione sono state inserite le immagini di seguito in cui sono presenti le merker sopra indicate e dove avviene la rispettiva assegnazione.

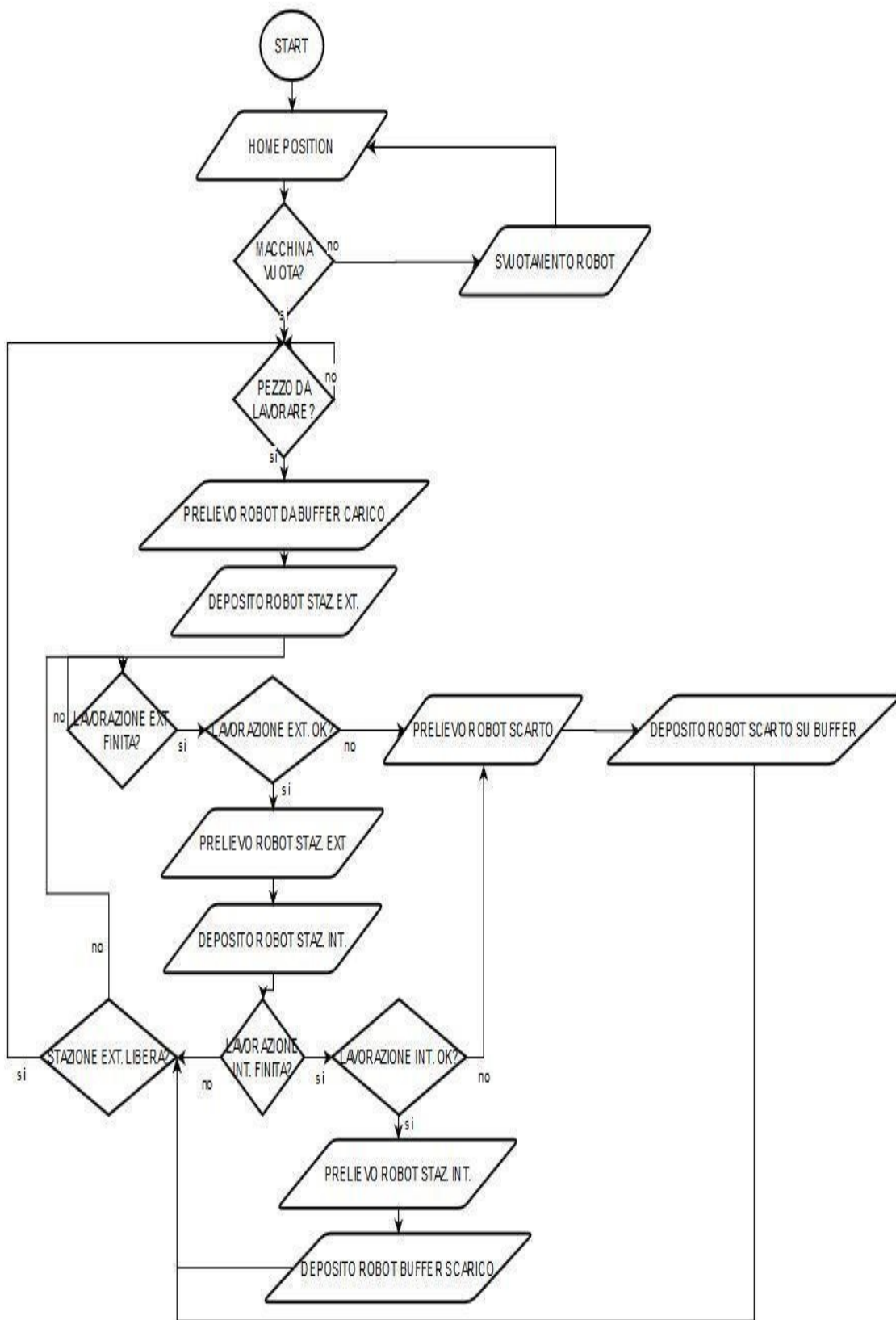


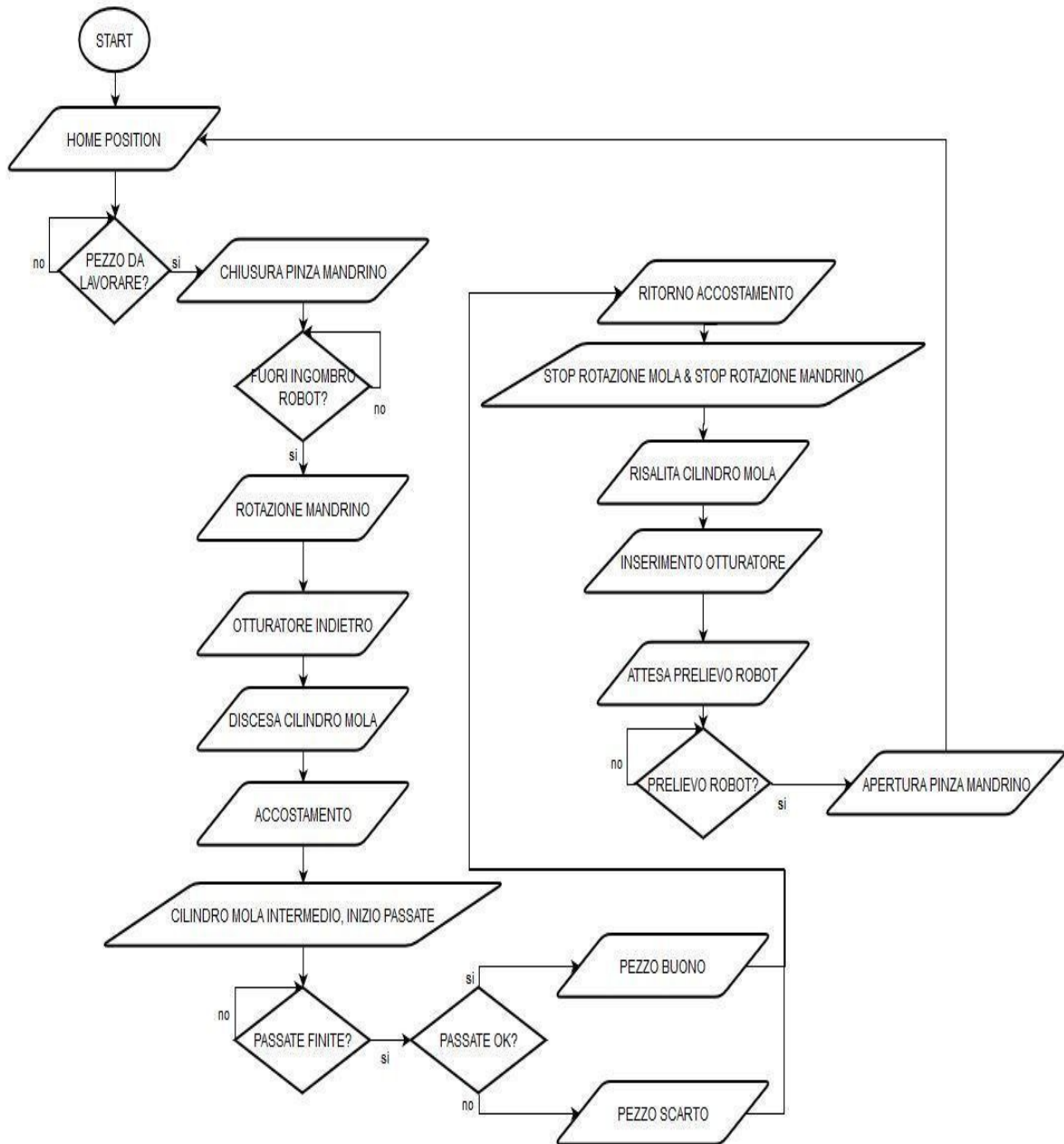


Tutte le FC per poter funzionare però vengono richiamate dall'OB 1 tramite la funzione "call", se così non fosse il PLC non le comprenderebbe nel monitoraggio e quindi non si avrebbe la giusta sincronia tra i sottoprogrammi.

Ulteriori Strumenti di Analisi

Di seguito si analizzerà il ciclo di lavoro attraverso flowchart e rete di petri, valutando le criticità a cui si è dovuto far fronte e le scelte di programmazione per superarle.





I precedenti flowchart rappresentano il ciclo della macchina, il primo è il ciclo della spazzolatrice all'interfaccia con l'automazione esterna, in cui si mostra come il robot di asservimento carica sempre la stazione esterna per prima e poi sposta il semilavorato sulla stazione interna, a meno di scarto; nel frattempo che la stazione interna è in lavorazione è possibile ricaricare un grezzo sulla stazione esterna, quindi verrà liberata la stazione interna appena questa avrà terminato le operazioni e potrà così ricevere il nuovo

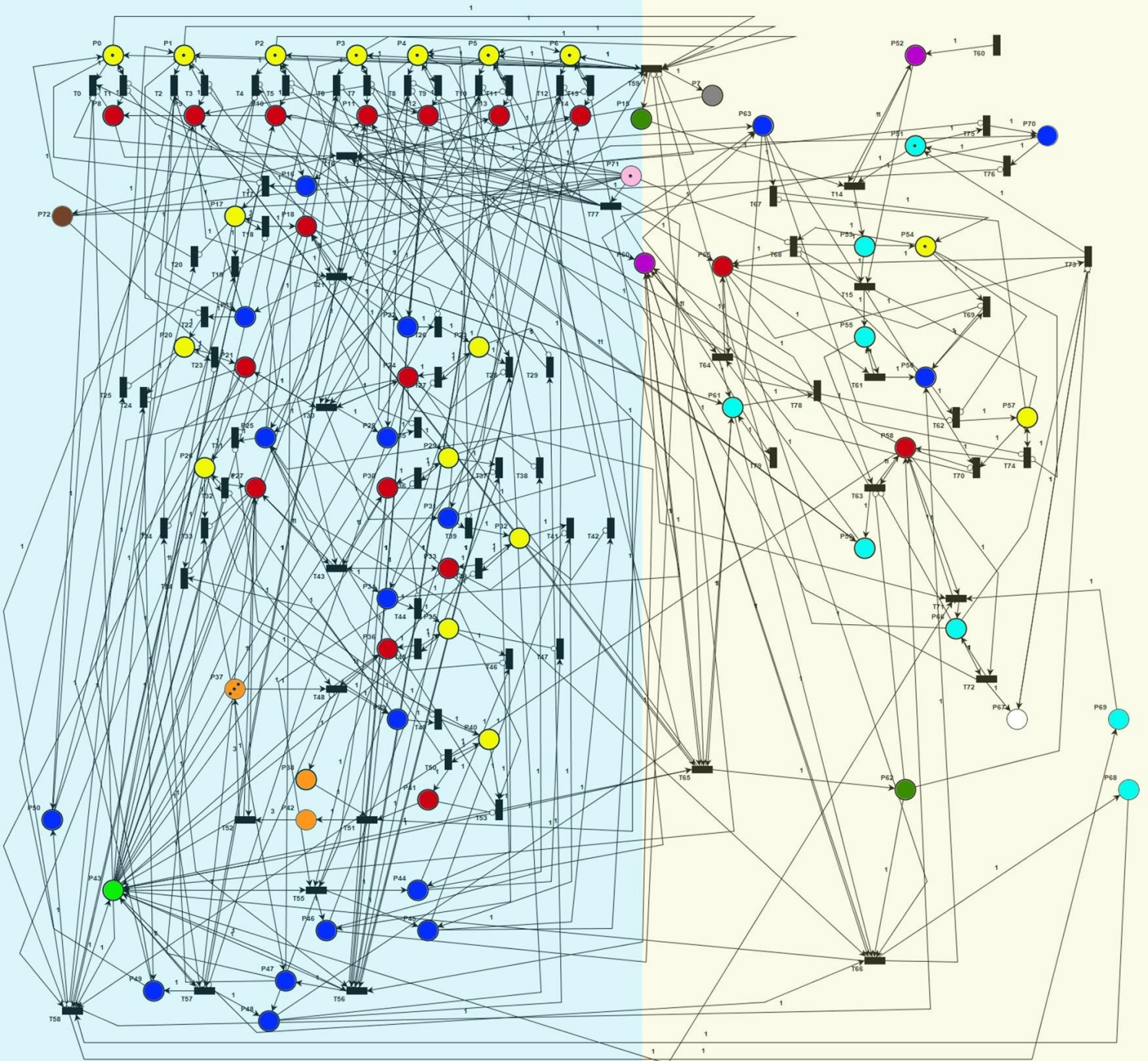
semilavorato dalla stazione esterna. Si evince anche che in caso di interruzione brusca del ciclo, al riavvio se le stazioni non sono vuote i pezzi in esse vengono depositati sul buffer di scarto dal robot in via preventiva per poi essere discriminati.

Nel secondo flowchart, invece, viene esaminato il ciclo della singola stazione dal momento in cui il robot finisce il pick&place del pezzo e fornisce il fuori ingombro, il ciclo comincia con l'apertura della pinza del mandrino e la rotazione dello stesso, avviene la discesa della mola, cominciando a ruotare, quindi l'accostamento e da qui si iniziano una serie di passate della mola in verticale sul pezzo tra il micro in basso del cilindro stesso e uno intermedio. Il numero delle passate dipende dal componente in lavorazione ed è customizzabile da pannello operatore. Durante questa fase viene monitorata la pressione di accostamento del cilindro, lo sfioro viene notificato dalla cella di carico e la corsa del cilindro di accostamento misurata da un trasduttore lineare, dalla misura di quest'ultimo per sottrazione si otterrà la misura del raggio della mola con la quale comandare la frequenza di rotazione della mola stessa tramite inverter, affinché rimanga costante la velocità tangenziale sulla circonferenza esterna della mola stessa.

Di seguito, invece, viene mostrata un'altra rappresentazione del ciclo della stazione attraverso l'impiego della rete di Petri.

Macchina

Robot



- | | |
|---------------------------------|----------------------------------|
| Sensori componenti mobili | Posizione iniziale |
| Stato componenti mobili | Presenze pezzo buffer/macchina |
| Commutazione componenti mobili | Chiamate robot prelievo/deposito |
| Loop ciclo passate | Stati robot |
| Fine ciclo | Conta pezzi finiti |
| Macchina esterna in lavorazione | Acquisizione all'avvio |

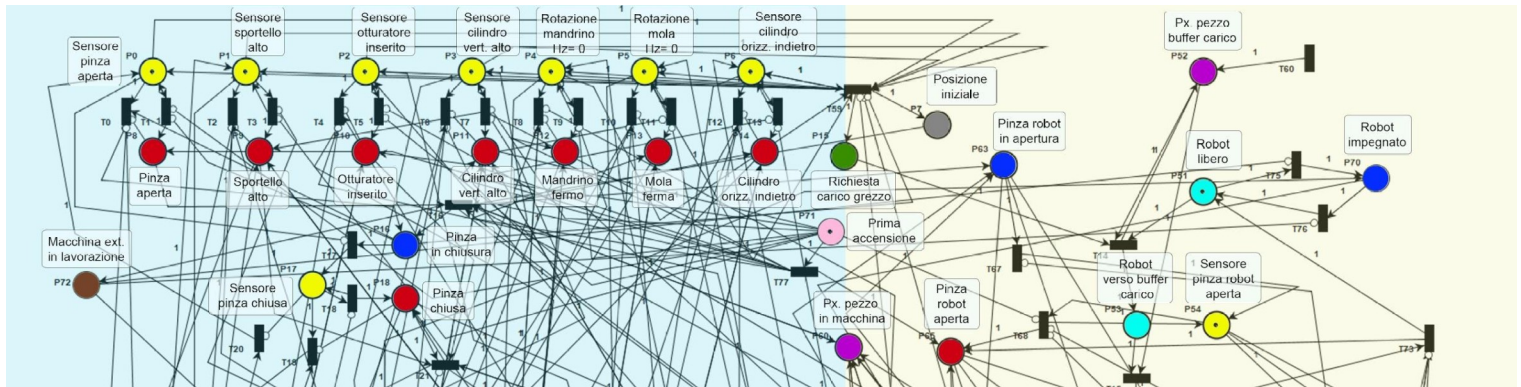
L'insieme dei posti è il seguente:

- P0 Sensore Pinza Mandrino Aperta (giallo)
- P1 Sensore Sportello Aperto (giallo)
- P2 Sensore Otturatore Inserito (giallo)
- P3 Sensore Cilindro Verticale Alto (giallo)
- P4 Frequenza Inverter Mandrino Hz=0 (giallo)
- P5 Frequenza Inverter Mola Hz=0 (giallo)
- P6 Sensore Cilindro Orizzontale Indietro (giallo)
- P7 Posizione Iniziale (grigio)
- P8 Pinza Mandrino Aperta (rosso)
- P9 Sportello Aperto (rosso)
- P10 Otturatore Inserito (rosso)
- P11 Cilindro Verticale Alto (rosso)
- P12 Mandrino Fermo (rosso)
- P13 Mola Ferma (rosso)
- P14 Cilindro Orizzontale Indietro (rosso)
- P15 Richiesta Carico Pezzo Grezzo in Macchina (verde scuro)
- P16 Pinza Mandrino in movimento Avanti (blu)
- P17 Sensore Pinza Mandrino Chiusa (giallo)
- P18 Pinza Mandrino Chiusa (rosso)
- P19 Sportello in movimento Avanti (blu)
- P20 Sensore Sportello Chiuso (giallo)
- P21 Sportello Chiuso (rosso)
- P22 Otturatore in movimento Indietro (blu)
- P23 Sensore Otturatore Disinserito (giallo)
- P24 Otturatore Disinserito (rosso)
- P25 Cilindro Verticale in movimento Avanti (blu)
- P26 Sensore Cilindro Verticale Basso (giallo)
- P27 Cilindro Verticale Basso (rosso)

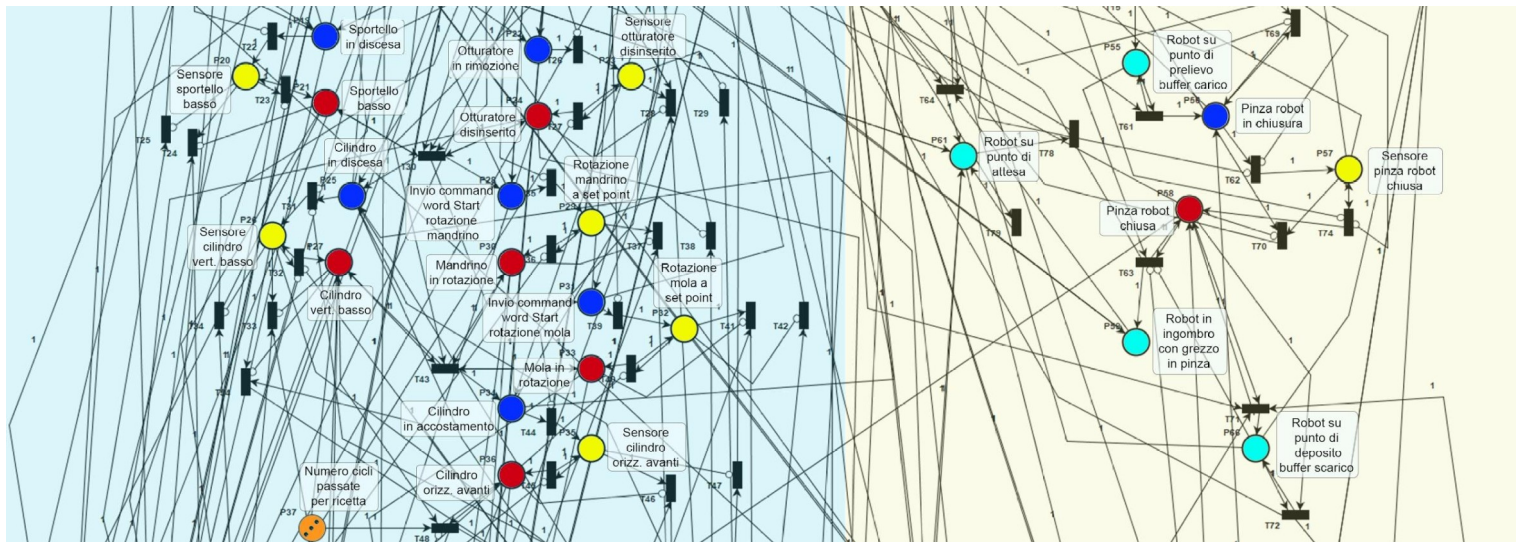
- P28 Invio command word Start Rotazione Mandrino (blu)
- P29 Mandrino a Set Point (giallo)
- P30 Mandrino in rotazione (rosso)
- P31 Invio command word Start Rotazione Mola (blu)
- P32 Mola a s Set Point (giallo)
- P33 Mola in rotazione (rosso)
- P34 Cilindro Orizzontale in movimento avanti (blu)
- P35 Sensore Cilindro Orizzontale avanti (giallo)
- P36 Cilindro Orizzontale avanti (rosso)
- P37 Numero Cicli Passate per ricetta (arancio)
- P38 Start loop Passate (arancio)
- P39 Cilindro verticale in movimento intermedio (blu)
- P40 Sensore cilindro verticale Intermedio (giallo)
- P41 cilindro verticale Intermedio (rosso)
- P42 Feedback numero Passate (arancio)
- P43 Fine ciclo (verde)
- P44 Cilindro Orizzontale in movimento indietro (blu)
- P45 Invio Command word Stop rotazione Mandrino (blu)
- P46 Invio Command word Stop rotazione Mola (blu)
- P47 Cilindro verticale in movimento Indietro (blu)
- P48 Otturatore in movimento Avanti (blu)
- P50 Pinza mandrino in movimento Indietro (blu)
- P51 Robot Libero (azzurro)
- P52 Presenza Pezzo su Buffer Carico (fucsia)
- P53 Robot in movimento verso Buffer di carico (azzurro)
- P54 Sensore Pinza Robot Aperta (giallo)
- P65 Pinza Robot Aperta (rosso)
- P55 Robot su punto Prelievo da Buffer (azzurro)
- P56 Pinza robot in movimento chiusura (blu)

- P57 Sensore Pinza Robot Chiusa (giallo)
- P58 Pinza Robot Chiusa (rosso)
- P59 Robot in Macchina con Grezzo in pinza (azzurro)
- P60 Presenza Pezzo in macchina (fucsia)
- P63 Pinza Robot in movimento Apertura (blu)
- P61 Robot in posizione di Attesa-Fuori Ingombro (azzurro)
- P62 Richiesta Prelievo Lavorato da Macchina (verde scuro)
- P68 Robot in Macchina su punto di prelievo (azzurro)
- P69 Robot verso Scarico con Lavorato in Pinza (azzurro)
- P66 Robot in Deposito su Buffer Scarico (azzurro)
- P67 Pezzi Lavorati +1 (bianco)
- P70 Robot Impegnato (azzurro)
- P71 Prima Accensione (rosa)
- P72 Macchina esterna in lavorazione (marrone)

La Rete rappresenta il ciclo automatico di una delle due stazioni interfacciato con il carico e lo scarico da parte del Robot, la sequenza delle operazioni è la seguente: prelievo di un grezzo dal buffer di carico, deposito in macchina del pezzo da lavorare, scambio del pezzo tra il robot e la pinza della stazione, attesa fuori ingombro e inizio ciclo macchina. Quest'ultimo avviene con la discesa dello sportello, rimozione otturatore, discesa cilindro verticale con start rotazione mandrino e mola, accostamento cilindro orizzontale, loop passate e riposizionamento. A questo punto il Robot provvede a liberare la stazione dal pezzo ormai lavorato e al deposito altrove.

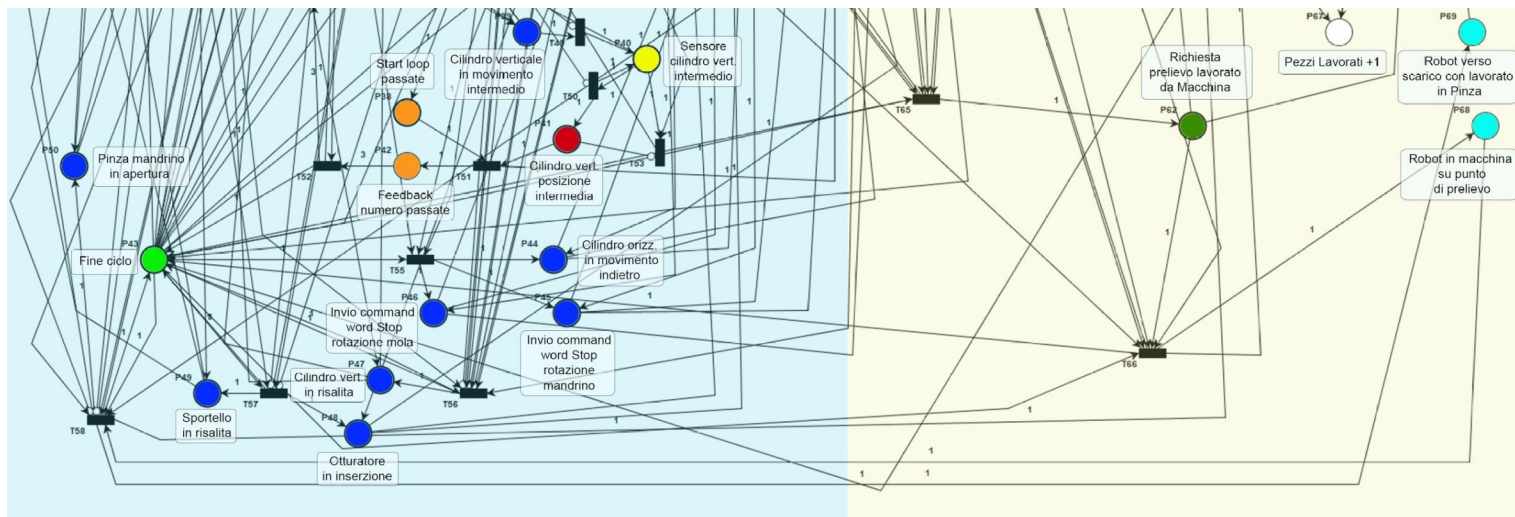


La rete simula ogni passaggio macchina a partire da una situazione di Posizione Iniziale, individuata dalla marcatura iniziale avente il Robot a disposizione con la pinza aperta. Ogni movimento prevede la commutazione di una valvola, la perdita del segnale del sensore della posizione precedente, l'arrivo del sensore della nuova posizione che notifica il raggiungimento del nuovo stato. Nella rete raffigurata nell'immagine sottostante sono visibili transazioni orizzontali e verticali, quelle orizzontali rappresentano gli step del ciclo macchina, quindi sono passaggi logici del programma e danno luogo a quelle verticali che, invece, si possono assimilare ai comandi delle uscite di un PLC, quindi alla commutazione delle valvole o la gestione degli inverter, con l'unica eccezione data da quelle verticali comprese tra un posto giallo e uno rosso, ovvero quelli ad indicare il sensore di un componente e lo stato da esso identificato. Queste transizioni hanno lo scopo notificare il raggiungimento di un nuovo stato, alcuni esempi sono T1, T3, T18, T27, e così via. Le transizioni T0, T2, T19 o, ancora, T28 hanno lo scopo di verificare che lo stato sia stato abbandonato al momento di una commutazione, infatti simulano lo spegnimento dei sensori dello stato precedente dovuto allo spostamento del componente.



La transazione sorgente posta in alto a destra nella figura della rete simula l'arrivo di un nuovo grezzo sul buffer di carico, quando ciò avviene, se il Robot non è impegnato in altro e la macchina è in posizione iniziale e quindi richiede di essere caricata, si dà il via alle missioni robot che una volta eseguito il caricamento si posizionerà in un punto di attesa diverso da quello iniziale finché la lavorazione non è terminata.

Alla rete rappresentata sono stati imposti limiti di capacità sui posti pari a 1 per una migliore resa grafica, già provata dai numerosi archi. In fase di analisi, invece, erano previsti posti limitatori con singolo token allo stesso scopo: per limitare ad uno lo stato possibile del componente in ogni momento. In altre parole un cilindro, ad esempio, che presenta il sensore di posizione indietro acceso avrà il posto stato di cilindro indietro con un token; senza la limitazione ad uno dello stato il posto ad esso associato, fintanto che il sensore rimane acceso, avrebbe la transazione in ingresso sempre abilitata, quindi ci sarebbe un accumulo di token sul posto stato che non avrebbe senso.



La Rete implementata, quindi, per come costruita risulta essere limitata e non presentare Deadlock⁸, per cui è facilmente asseribile che sia ben costruita e rappresenti in maniera fedele il ciclo macchina desiderato.

Alla stregua di quanto visto, per mezzo del secondo flowchart e della rete di Petri, il ciclo automatico implementato in ambiente Simatic segue fedelmente quegli step verificando ad ogni segmento le condizioni logiche del passo successivo, rispettando il capitolato aziendale strutturando un ciclo a gradini. In ognuno dei quali si movimentava un'utenza per volta verificando al passo successivo l'avvenuto cambiamento di stato dell'uscita in serie alla condizione logiche per il nuovo gradino da effettuare.

Lo strumento appena utilizzato, la Rete di Petri, è un fondamentale strumento di analisi, permette di evidenziare come alcune movimentazioni, ad esempio, possano essere compiute in contemporanea e dare una certa priorità ad alcuni piuttosto che ad altri. Ciò è indispensabile quando non è noto a priori il funzionamento che deve avere una stazione o una macchina,

⁸ Deadlock: termine proprio delle Reti di Petri, in cui in un certo istante si possa presentare una situazione bloccante, dovuta da due o più azioni da compiere che non sono permesse poiché l'una non è attuabile a causa del non verificarsi delle altre. Nell'ambito delle Reti di Petri essa si verifica quando si raggiunge uno stato per cui non sono più attive transizioni.

oppure se le specifiche di progetto non sono chiare o lasciano molto spazio al progettista.

Qui le specifiche erano definite dall'azienda e non era lasciato molto spazio al programmatore, ma la rete ha messo in risalto la possibilità di optare per una sequenza di operazioni che contempli la contemporaneità di alcuni componenti, come la rimozione dell'otturatore e la discesa dello sportello, che si discosta dalla scelta aziendale di un movimento per ogni step di programma, ma garantisce una maggiore flessibilità al programma e un lieve risparmio di tempo ciclo.

Sarebbe inoltre possibile ampliare la rete precedente inserendo a destra della rappresentazione del robot l'altra stazione, strutturata in maniera simile a quella già mostrata, per simulare fedelmente l'intera macchina con l'automazione esterna, ciò potrebbe mostrare punti critici sul funzionamento del robot che attualmente non sono saltati all'occhio. Input questo che si lascia all'azienda come punto di partenza per ulteriori miglie e riguardanti il tempo ciclo dell'intera stazione.

Soluzioni Proposte

Il ciclo automatico è stato costruito per mezzo del SET della merker di passo, verificate le condizioni, ed il RESET della stessa al SET del passo seguente. In ambiente Simatic oltre al ciclo analizzato con gli ausili precedenti è stato implementato un ulteriore ciclo a passi forzati in caso di cambio mola, che permette un ciclo simulato, senza le rotazioni, per effettuare un'acquisizione della misura del raggio dell'attrezzo così da poter comandare la rotazione della spazzola al ciclo successivo alla corretta frequenza.

L'impiego di un trasduttore lineare, per misurare il raggio, porta in seno l'errore dello strumento, ma la rilevazione della misura è soggetta soprattutto a disturbi legati alle vibrazioni in fase di lavorazione. Questo problema ha una maggiore risonanza considerando che i PLC in generale non nascono per fare grandi manipolazioni di dati e le loro capacità in tal senso variano molto a seconda della CPU in uso. In questo caso in particolare si è costretti ad utilizzare la CPU 315 che per l'appunto ha dei limiti di gestione dati, rischiando quindi che la CPU soffra effettuando questo tipo di operazioni.

In una prima fase si era pensato di acquisire un numero di letture del trasduttore in fase di accostamento e di storicizzarle su una DB di appoggio per poi effettuare una media delle stesse, il risultato sarebbe poi stato utilizzato come misura effettiva per ottenere il raggio della mola ad ogni ciclo. Ciò avrebbe quindi portato all'implementazione di un filtro a media mobile, questa opzione però è stata scartata appunto per le motivazioni sopra citate.

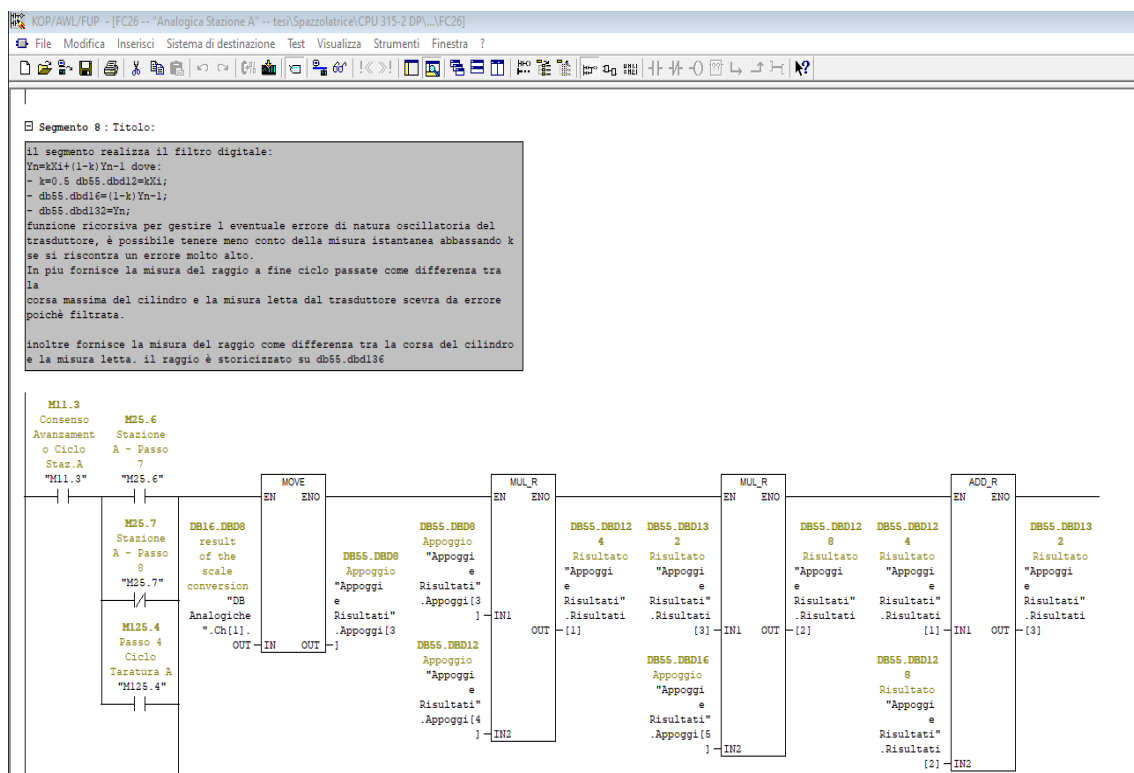
Un'altra possibilità presa in considerazione è stata l'impiego di un Filtro di Kalman,; anche questa possibilità , però, per i limiti hardware è stata messa da parte.

A seguito di test sul trasduttore si è optato per scegliere una soluzione meno onerosa in termini computazionali per il PLC, ovvero è stato implementata questa tipologia di “filtro digitale”:

$$Y_i = k * X_i + (1-k) * Y_{(i-1)}.$$

dove k è una costante compresa tra 0 e 1, essa esprime la fiducia che si

vuole dare al trasduttore in termini di affidabilità, più si riscontra che la misura sia esente da errore più si impone k vicino ad 1, imponendo k unitario si presume che il trasduttore non abbia errore. $Y(i-1)$ sta ad indicare la misura precedentemente effettuata e filtrata. Al primo ciclo, ossia quello di acquisizione precedentemente introdotto, si andrà a scrivere su una DB il valore appreso imponendo che sia proprio $Y(i-1)$, così da avere un punto di partenza per il filtro.

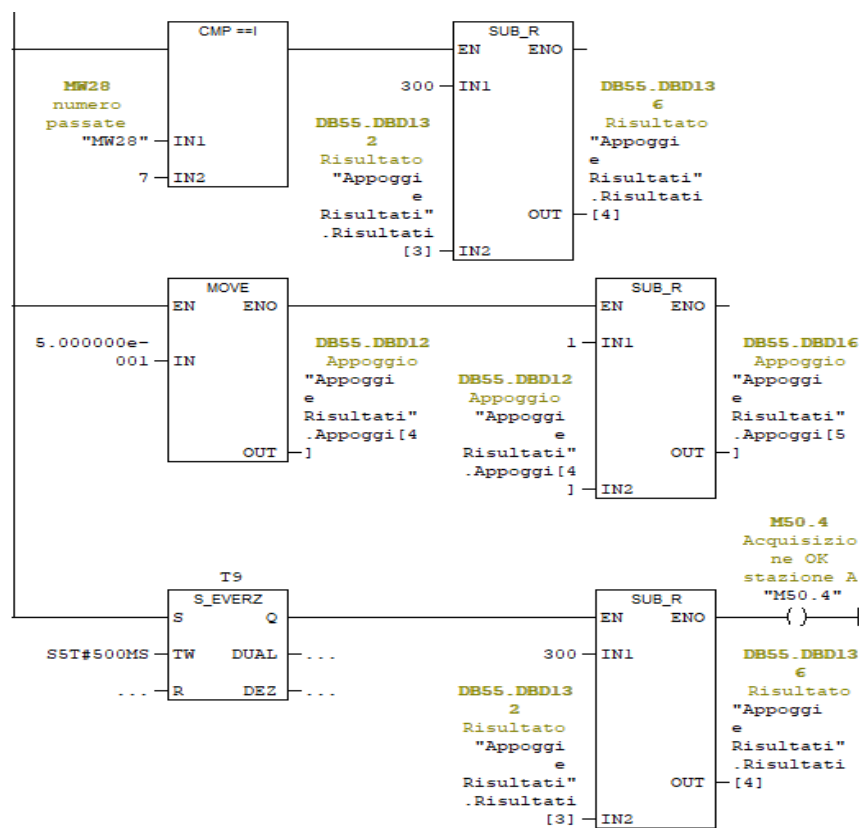


Il valore del raggio della mola deriva, quindi, dal valore in uscita dal filtro digitale, infatti, come visibile dalla precedente immagine il filtro lavora nell'arco temporale scandito da M25.6 e M25.7, quando non si sta facendo il ciclo di taratura.

In questo lasso temporale nel ciclo avviene il conteggio delle passate sul pezzo, visto che la corsa massima dello stelo del cilindro di accostamento è

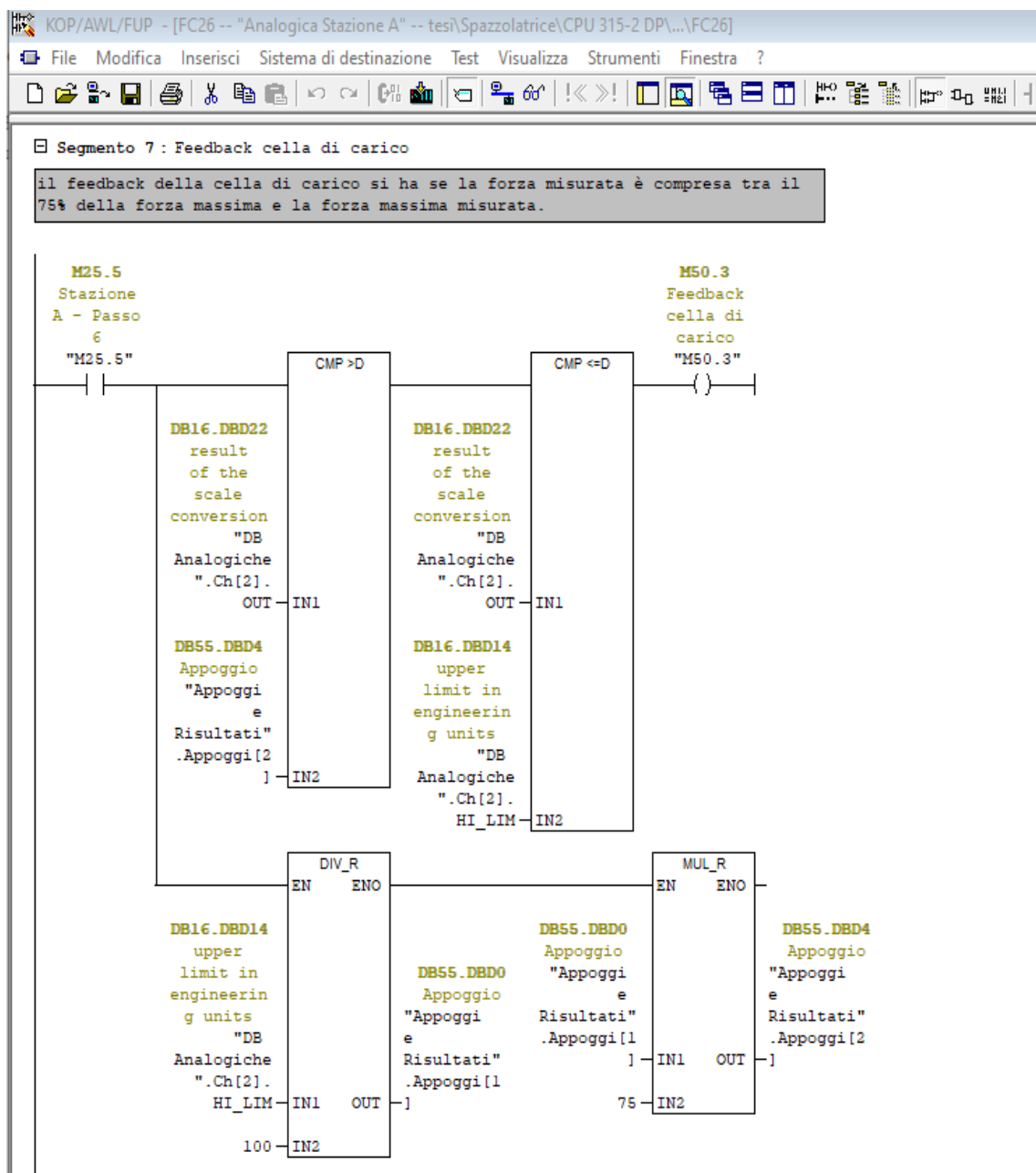
di 300 mm, andando a sottrarre a questo valore l'uscita del filtro otteniamo il raggio della mola.

Alla stessa maniera durante il ciclo di taratura, al passo 4, ovvero quando M125.4 è nello stato 1 il filtro lavorerà nell'arco di tempo dettato da T9 (figura in basso) e allo stesso modo per sottrazione si scriverà su Y(i-1) il valore acquisito, come precedentemente detto.



Risulta corretto dire che questa scelta si è fatta senza poter effettivamente provare la macchina con il nuovo software, per cui per il momento questa sembra essere la scelta migliore in rapporto all'hardware. Ciò non toglie però la possibilità di poter ritornare sui propri passi ed optare per un'altra opzione tra quelle per il momento abbandonate.

Tutte queste elaborazioni vengono eseguite solo dopo essere entrati in accostamento sul pezzo, il che è notificato dalla cella di carico. Quest'ultima viene calibrata da pannello attraverso l'impiego di Master, (campioni standard creati appositamente allo scopo) fornendo un valore della forza di spinta della mola sul pezzo. Empiricamente si è imposto che la cella di carico notifichi l'accostamento quando il valore istantaneo è compreso tra il 75% del valore massimo, ottenuto in fase di calibrazione, ed il valore massimo; come visibile dall'immagine sottostante.



L'ulteriore richiesta è quella di imporre alla mola una velocità tangenziale costante al variare del proprio raggio. Impostata a 25 Hz la frequenza della mola quando il raggio è massimo, ossia appena la mola è stata sostituita, è possibile calcolare il valore della velocità tangenziale che si vuole preservare al variare del raggio come:

$$V_{tang} = r * \text{velocità_angolare} \text{ ovvero}$$

$$\text{vel_angolare} = V_{tang}/r$$

Dalla formula precedente si nota che volendo mantenere V_{tang} costante la velocità angolare da imporre si calcola dividendo il valore richiesto dalla velocità tangenziale per la misura del raggio trovato di volta in volta.

Per ottenere la frequenza da inviare all'inverter bisognerà tener conto che la velocità angolare è espressa in rad/sec mentre la frequenza è espressa in Hz, ovvero:

$$\text{Freq_inv} = \text{vel_angolare} * 6,28.$$

Queste sono dunque le soluzioni a cui si è arrivati per far fronte alle problematiche legate al ripristino del controllo dell'usura della mola, ciò a cui si voleva arrivare per soddisfare le richieste aziendali. All'atto pratico risulta quindi che, rispetto alla lavorazione attuale della macchina, mettendo in pratica questa modifica software la lavorazione avrà un controllo della mola garantito dalla movimentazione del cilindro di accostamento tramite la misurazione del trasduttore con un onere computazionale adeguato all'hardware disponibile. L'avanzamento del cilindro, inoltre, non sarà più regolamentato da un regolatore di pressione,

bensi da una valvola proporzionale di cui è possibile regolare via software la portata e la pressione in camera, anche in relazione al momento dello sfioro sul pezzo, monitorato dalla cella di carico, che garantisce che una volta toccato lo statore con la mola questa non risenta di fluttuazioni di pressione non controllate. Per tutto il tempo in cui la stazione risulta in accostamento, inoltre, entra in funzione il filtro digitale che alla fine di ogni ciclo della lavorazione di un pezzo fornisce in uscita una misurazione filtrata della corsa del cilindro, questo valore poi viene sottratto all'intera lunghezza dello stelo del cilindro ed è quindi possibile monitorare nel tempo il raggio della mola che inevitabilmente, per usura, diminuirà di volta in volta. In questo modo sarà possibile comandare ad ogni ciclo una frequenza di rotazione mola diversa in maniera inversamente proporzionale all'usura della mola stessa tramite il raggio.

Conclusioni

L'obiettivo principale di questo lavoro di tesi era quello di sviluppare un nuovo software PLC per la gestione di una macchina spazzolatrice per statori presente presso lo stabilimento Denso che ha sede in Abruzzo. Le richieste erano quelle di migliorare la qualità della produzione attraverso una migliore gestione della mola, per abbassare il numero degli scarti ed allungare il tempo di vita della mola. In particolare, la richiesta ha riguardato il controllo della velocità tangenziale che si vuole mantenere costante al variare della misura del raggio della mola che, per usura, subisce variazioni. Inoltre è stato richiesto di seguire standard aziendali per la stesura del programma riguardanti la scelta obbligata del linguaggio di programmazione: il KOP, e la divisione del programma in vari sottoprogrammi con uno specifico nome.

Si tiene a precisare che il lavoro eseguito è un Retrofit, per cui non sono state apportate modifiche ai componenti di cui la macchina è formata, si è lavorato in ottica di riuso per dare una nuova vita all'automazione già esistente.

Dopo aver analizzato l'attuale codice in funzione sull'impianto, per raggiungere l'obiettivo ci si è avvalsi di strumenti quali flowchart e Reti di Petri per l'analisi delle sequenze lavorative e la strutturazione del programma. Si è giustificato l'impiego del PLC ripercorrendo un pò di

storia dell'Automazione Industriale disquisendo sulle caratteristiche che rendono fondamentale l'uso di questi dispositivi in quest'ambito. Dal punto di vista della programmazione ci si è focalizzati sull'Ambiente Simatic Manager di Siemens, poiché era la richiesta aziendale, mostrando i possibili linguaggi ammessi e optando per il KOP per rientrare nei canoni imposti dall'azienda.

Con l'ausilio degli strumenti precedentemente menzionati si è potuto programmare il nuovo software, si sarebbe voluto approfondire la trattazione con una analisi dei benefici auspicati in fase di progettazione, ma per questioni di produzione l'impianto non è ancora disponibile per le modifiche. Non si è potuto quindi osservare come i requisiti a cui si è fatto fede in fase di progetto possano beneficiare sul processo produttivo in termini di tempo ciclo e affidabilità di lavorazione, né fare una stima del risparmio economico garantito all'azienda, in termini di miglior utilizzo delle mole e riduzione di pezzi mal processati.

Ulteriori possibili sviluppi rispetto a quanto già presentato potrebbero riguardare l'approfondimento dell'analisi tramite le Reti di Petri temporizzate del ciclo macchina per cercare di velocizzare la lavorazione e ricercare una migliore soluzione per ottenere il valore del raggio rispetto a quella a cui si è ricorso.

In ogni caso si ritengono soddisfatti i requisiti di progetto e le richieste per un miglior controllo sul processo rispetto alla lavorazione attuale.

Bibliografia

- [1] Siemens, Simatic, Programmazione con STEP 7, manuale, Edizione 03/2006
- [2] Siemens, Simatic, Sistema di automazione S7-300, Caratteristiche delle unità modulari, Manuale del prodotto, 06/2017
- [3] Siemens, Simatic, Sistema di automazione S7-300, Dati delle Unità, Manuale di riferimento, 02/2004

Sitografia

(1) <http://noisiamola5i.altervista.org/blog/wpcontent/uploads/2011/10/Storia-dei-PLC.doc>

(2) <http://xoomer.virgilio.it/tsc/appunti/plc.doc>