Università Politecnica delle Marche

Facoltà di ingegneria

Corso di Laurea in

Ingegneria Informatica e dell'Automazione

Studio e sviluppo di algoritmi di controllo tramite PLC

Study and development of control algorithms via PLC

Tesi di laurea di: Masoud Morcos Safwat Sobhi

Relatore: Prof. Gianluca Ippoliti

Correlatore : Prof. Giuseppe Orlando

Anno accademico: 21/22

Sommario

Il presente elaborato si propone come obiettivo lo studio di come è strutturato un PLC sia Hardware che software analizzando i suoi elementi funzionali e le fasi principali per programmarlo al fine di sviluppare una serie di algoritmi di controllo mediante dei linguaggi di programmazione del PLC.

L'elaborato si propone anche come si tradurre da un linguaggio SFC al Ladder vediamo anche come si programma un PLC utilizzando anche un simulatore.

Indice

Capitolo 1

Introduzione

La prima parte della tesi riporta un'introduzione al PLC con particolare riferimento alla architettura costruttiva.

La seconda parte della tesi parla delle caratteristiche del Hardware.

La terza parte parla degli elementi funzionali del PLC e le fasi principali per iniziare a programmare.

La quarta parte introduce i linguaggi di programmazione usati, in particolare il linguaggio Ladder e vediamo come si traduce da SFC a Ladder.

La quinta parte si basa su creare dei programmi per dei modelli reali facendo la simulazione sul simulatore PLC.

Capitolo 2

PLC

Cos'è il PlC

Un PLC è un dispositivo o sistema digitale elettronico che utilizza una memoria programmabile per memorizzare informazioni o istruzioni, atte a realizzare specifiche funzioni, finalizzate al controllo di sistemi combinatori e sequenziali per la gestione di macchine e processi, quali: operazioni logico-aritmetiche, temporizzazioni, conteggi, comparazioni, codifiche e decodifiche.

Il PLC nasce come sistema per la produzione, in sostituzione del sistema a relais

E’ caratterizzato da:

- Dimensione contenuta
- Energia consumata
- Contenuta facile
- Rrogrammabilità
- Versatilità
- Robustezza

Il PLC, come i circuiti a relay, è un dispositivo che elabora "istantaneamente" una serie di comandi da imporre ad un sistema automatizzato in risposta alla situazione che si presenta "istantaneamente" in ingresso al PLC.

I comandi sul sistema automatizzato sono imposti da grandezze elettriche di tensione o di corrente dei seguenti tipi:

ON/OFF

digitale (più di due livelli di uscita a disposizione)

analogico (uscita variabile in modo continuo).

La situazione istantanea del processo da controllare è letta attraverso il livello di segnale (tensione o corrente) proveniente da uno o più sensori distribuiti nel processo (di tipo ON/OFF, digitale o analogico).

Il PLC è un controllore con architettura general-purpose dedicato al controllo logico sequenziale.

Progettato per l‟uso in un ambiente industriale e quindi con caratteristiche di:

- Affidabilità
- Espandibilità
- Semplicità di programmazione
- Possibilità di facile migrazione tra dispositivi di produttori diversi.

Definizione

Il PLC è un‟apparecchiatura composta da componenti elettronici, fornita di memoria programmabile e non, contenenti sia dati che programmi, in grado di leggere ed eseguire le istruzioni dei programmi stessi, interagendo con un sistema da controllare, tramite dispositivi di input e output del tipo digitale o analogico (da Norma CEI 65-23).

Definizione (IEC 1131.3)

Il PLC è 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, di sequenziamento, di temporizzazione, di conteggio e calcolo aritmetico, e per controllare, mediante ingressi ed uscite sia digitali che analogici, vari tipi di macchine e processi.

Norme di riferimento

CEI 65-23 (EN 61131-1-2-3) : si occupa delle informazioni generali, definizioni delle apparecchiature e relative prove.

CEI 65-40 (EN 61131-1-2-3) : prende in esame i linguaggi di programmazione.

CEI 44-5 (EN 60204-1) : si occupa delle apparecchiature elettroniche e dell‟equipaggiamento programmabile fornendo i criteri per l‟uso e l‟installazione.

CEI 110-13, 110-25 : si occupa della costruzione ed installazione dei PLC con riferimento alla compatibilità elettromagnetica (EMC).

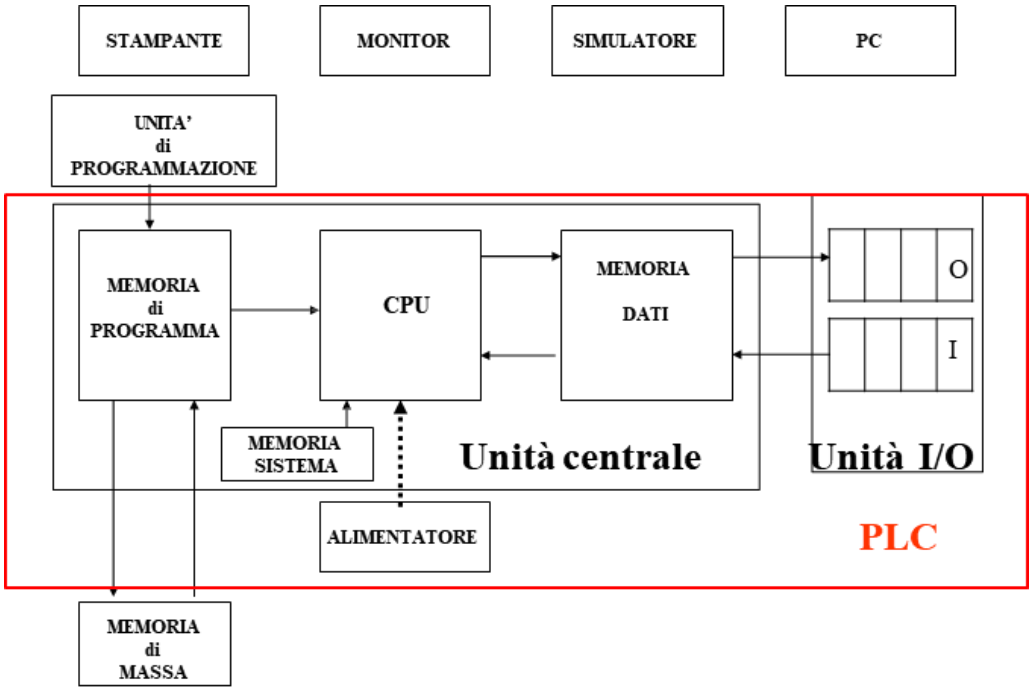
Capitolo 3

Hardware

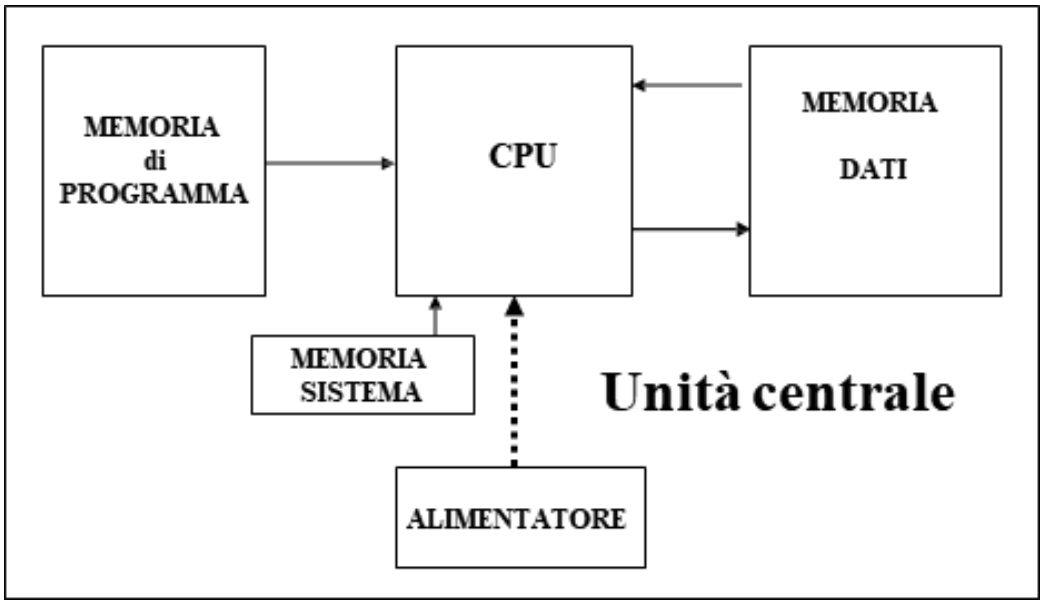
La parte Hardware e' composta sostanzialmente da due parti principali

Unita' centrale dove ci sono la CPU che elabora i dati presneti nelle vare memorie (memoria dati, memoria di programma e memoria sistema) leggendo e scrivendo le informazioni.

Unita' Ingresso Uscita che rappresenta la interfaccia tra l'elaborazione del programma attuata dalla CPU del PLC degli ingressi e gli attuatori.



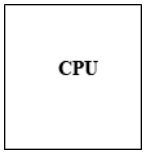
Unita' centrale



I PLC possono operare secondo uno dei seguenti tipi di scansione:

- Sincrona I / O
- Sincrona I e asincrona O
- Asincrona I / O

CPU

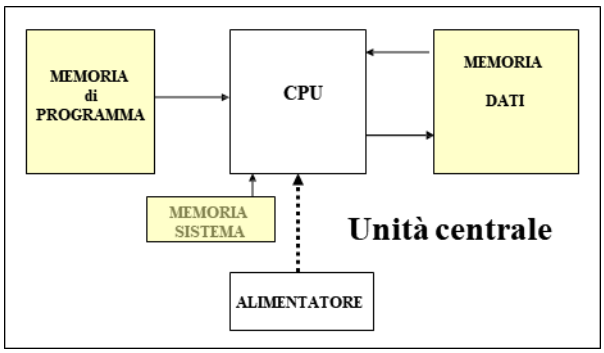


La CPU (Central Processing Unit o Unità Centrale) è la parte più importante del PLC di cui costituisce "l'intelligenza". Essa è l'unità di governo del sistema ed il suo elemento centrale è un componente integrato denominato Microprocessore. Il microprocessore racchiude in sè tutte le funzioni di calcolo e controllo del processore centrale di un normale calcolatore. La sua caratteristica più importante è la programmabilità che ha consentito il grande passo in avanti dalla logica cablata alla logica programmabile. Attualmente i microprocessori utilizzati come CPU dei controllori programmabili sono molto vari in quanto non esiste una qualsiasi forma di standardizzazione. Ogni costruttore impiega il microprocessore che ritiene più adatto alle prestazioni che vuole fornire al suo sistema.

La CPU legge e scrive le informazioni in dispositivi chiamati memorie.

La CPU lavora secondo un ordine sequenziale detto SCANSIONE.

Memorie



Le memoria si posssono distinguere in due modi in base all'impiego o in base alle caratteristiche.

In base all'impiego si distinguono le memorie di:

SISTEMA

PROGRAMMA

DATI

In base alle caratteristiche si distinguono in:

ROM

RAM

EPROM

EEPROM

La memoria di sistema è una memoria ROM realizzata dal costruttore. Essa governa il *sistema operativo* del PLC.

La memoria di programma può essere sia RAM che EPROM:

In fase di messa a punto del programma è bene operare su memorie RAM;

In faseoperativa del PLC il programma viene scritto su EPROM

Il PLC ha bisogno di memoria sia per il proprio sistema operativo sia per la memorizzazione del programma utente sia per l'elaborazione dei dati intermedi durante l'esecuzione del programma. Di solito il costruttore utilizza, per la memorizzazione del sistema operativo, una memoria di tipo ROM (Read Only Memory), che ha appunto le caratteristiche di essere non volatile e di non poter essere modificata visto che è una memoria di sola lettura. Il programma utente , al contrario, deve poter essere modificato in quanto la sua stesura è demandata all'utente che lo adatterà alle sue esigenze iniziali e, qualora fosse necessario, lo modificherà in seguito a nuove esigenze. Lo stesso si può dire riguardo la memoria necessaria per la memorizzazione dei risultati intermedi in quanto sulla stessa il PLC effettuerà continuamente operazioni di lettura e scrittura. Quindi, sia per il programma utente che per la memorizzazione dei risultati intermedi, il costruttore fornisce il PLC di una memoria di tipo RAM (Random Access Memory) che può essere letta e riscritta. La dimensione della RAM utente è uno dei parametri che caratterizza maggiormente un PLC in quanto da essa dipende la lunghezza del programma che può essere gestito dal PLC. Per PLC di piccola taglia, attualmente, si hanno memorie utente che hanno dimensione da tre a quattro Kbyte, a cui corrisponde la capacità di memorizzare programmi di circa mille istruzioni. Ovviamente in PLC di taglia superiore la dimensione della RAM

varia proporzionalmente con la complessità del set di istruzioni del linguaggio di programmazione proprio del PLC e con le dimensioni, prevedibilmente superiori, dei programmi necessari per gestire automatismi complessi.

La memoria di sistema :

Come già detto, la memoria di sistema serve a conservare tutte quelle particolari istruzioni che servono per la gestione ed il controllo del funzionamento della CPU e che pertanto costituiscono un vero e proprio SISTEMA OPERATIVO del PLC. Dato che il suo contenuto è di primaria importanza per il controllore, vengono utilizzate delle memorie di tipo ROM, per evitare la sua involontaria cancellazione. Nulla vieta comunque che sia PROM o EPROM, purchè non accessibile all'utente.

La memoria di programma :

È la memoria destinata a contenere le istruzioni che costituiscono il programma eseguibile dal PLC. Per svolgere tale funzione essa deve essere accessibile all'utente (a cui è demandata la stesura del programma stesso) e viene quindi realizzata con memorie di tipo RAM.

La memoria dati :

È anche detta MEMORIA DI LAVORO e prevede due sezioni distinte: i FLAG ed i REGISTRI ; − I FLAG (o MERKER) Si tratta di una certa quantità di memoria in formato WORD o DOUBLE WORD che può essere indirizzata anche in formato BYTE od a singoli BIT e che può essere utilizzata dall'utente per memorizzare risultati intermedi durante l'elaborazione del programma. Tali risultati possono essere successivamente utilizzati in altre parti del programma. I singoli BIT di queste memorie possono essere "SETTATI" (cioè posti al valore logico 1) o "RESETTATI" (cioè posti al valore logico 0).

Alimentatore

L'unità centrale di un PLC è equipaggiata pertanto con un alimentatore in cui sono raggruppati tutti i dispositivi necessari per fornire tale alimentazione quali :

− TRASFORMATORI ;

− RADDRIZZATORI (Convertitori C.A./C.C.)

− STABILIZZATORI.

Molulo I/O

I moduli o schede d'uscita e d'ingresso rappresentano sostanzialmente la interfaccia tra l'elaborazione del programma attuata dalla CPU del PLC degli ingressi e gli attuatori che costituiscono il sistema di comando verso l'impianto controllato.

Unita' d'ingressi

Le informazioni provenienti dal processo controllato possono essere semplicemente definite come "segnali in ingresso". Questi segnali sono ovviamente di tipo elettrico e dovranno essere trattati in modo che siano riconoscibili dalla CPU. Il compito delle schede d'ingresso (come poi, in senso inverso, faranno anche le schede d'uscita) è quello di consentire il dialogo tra il PLC ed il gruppo di potenza, o , per essere più precisi, per permettere al PLC di acquisire i comandi e lo stato degli attuatori del suddetto gruppo. Tali segnali possono essere sia segnali di tipo binario, caratterizzati dalla possibilità di assumere due soli valori (" 1 logico" e " 0 logico"), a prescindere dalla natura iniziale del segnale, che potrebbe essere anche di tipo analogico o comunque un segnale instabile nel tempo, ma lo si vuole trattare come segnale "digitale binario", caratterizzato quindi da due soli stati possibili : assenza di tensione o presenza di tensione. I segnali possono essere anche di tipo analogico, quindi variabili nel tempo dentro un prefissato intervallo di valori, in questo caso devono preventivamente essere trattati da appositi convertitori A/D prima di essere elaborati dalla CPU.

Schede d'ingresso digitali

Nel caso di segnali da trattare come segnali di tipo ON/OFF le schede d'ingresso devono essere in grado di "capire" quando il segnale in ingresso è da considerare ON e quando è da considerare OFF, quando si tratta di un disturbo ed inoltre di isolare galvanicamente la CPU dall'esterno in modo che eventuali sbalzi di tensione o sovraccarichi od addirittura corto circuiti non danneggino la stessa. Uno dei compiti svolti dalle schede d'ingresso è quello di adattare il livello e le caratteristiche del segnale. Infatti, mentre la tensione di funzionamento interna del PLC è una tensione bassa (di solito 5 V), i segnali possono presentarsi con livelli di tensione diversi (24, 48, 110, 220 V). La prima operazione svolta dalle schede di ingresso consiste nella messa in forma o squadratura del segnale. Per determinare con certezza se il segnale è ON oppure OFF, di solito le schede d'ingresso sono costruite in modo che il segnale esterno viene riconosciuto tale entro un intervallo di valori prefissato. Ad es. si consideri un segnale a 24 V

Schede d'ingresso analogiche

Esistono in commercio dei sensori analogici (termocoppie, resolver, ecc.) che forniscono, in relazione al livello di un liquido in un serbatoio, alla temperatura di un forno, alla pressione in una tubazione etc., un segnale variabile con continuità entro due limiti. I valori di questi segnali sono standardizzati; valori tipici sono : ± 50 mV, ± 1 V, ± 5 V, ± 10 V, 0..10 V, 0..20 mA, ± 20 mA, +4..20 mA. Per rendere possibile l'elaborazione da parte del PLC del valore del segnale in ingresso, qualunque esso sia, bisogna convertire il segnale analogico variabile in un segnale digitale comprensibile alla CPU.

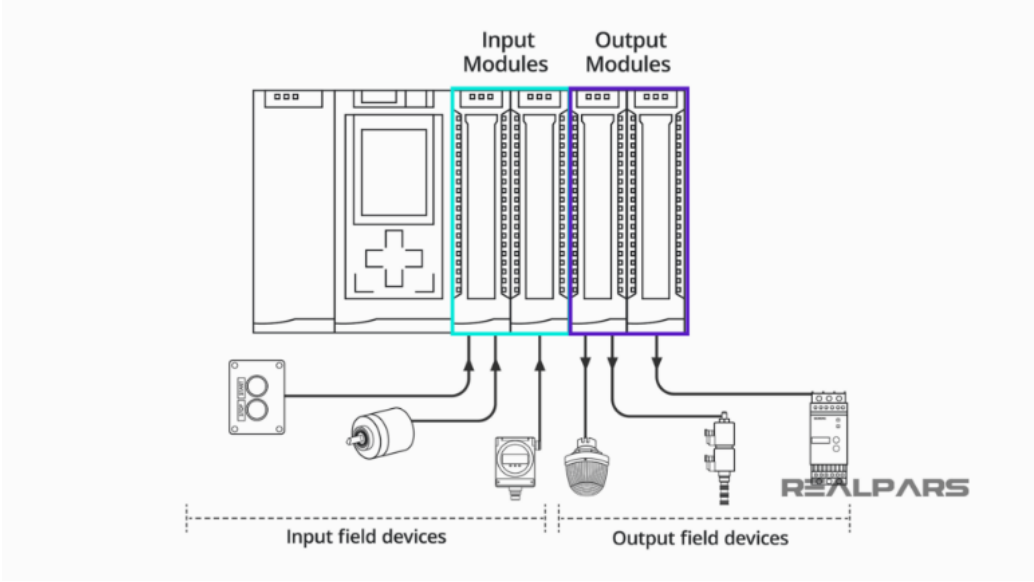
Schede d'uscita digitali

I moduli o schede d'uscita rappresentano sostanzialmente la interfaccia tra l'elaborazione del programma attuata dalla CPU del PLC e gli attuatori che costituiscono il sistema di comando verso l'impianto controllato. Per quanto concerne il numero di uscite presenti su ogni scheda di uscita valgono le stesse considerazioni fatte per le schede di ingresso. Si definisce "tempo di emissione dell'uscita" l'intervallo di tempo che intercorre tra l'istante cui appare un'immagine d'uscita nella memoria del PLC e quello in cui viene raggiunta la soglia di tensione a cui corrisponde l'effettiva attivazione dell'uscita. Per le uscite a relè questo tempo dipende principalmente dal tempo di salita dei relè, che varia da due a più decine di millisecondi, mentre per le uscite statiche (a transistor) tale

tempo è decisamente più breve. Le tensioni più frequentemente utilizzate sono 24 o 48 V, si possono comunque avere tensioni diverse. La corrente che ogni singola uscita è in grado di erogare va da 100 mA a 2 A.

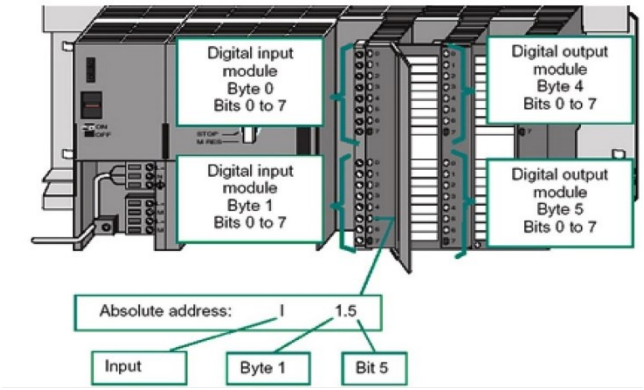
Schede d'uscita analigiche

Le schede di uscita analogiche svolgono, in senso inverso, le stesse funzioni delle schede di ingresso analogiche. Operano cioè una conversione digitale-analogica (D/A) dei valori, elaborati dal PLC così come definito da programma, che devono essere trasmessi all'attuatore collegato alla scheda, il quale è costruito per ricevere un segnale analogico in tensione od in corrente. Vale tutto quanto già detto a proposito delle schede d'ingresso analogiche.



Indirizzamento dell'I/O

Nella configurazione del PLC le variabili di ingress e uscita sono idintificate tramite un riferimento posizionamenale



Capitolo 4

Elementi funzionali del PLC

Ogni PLC dispone di un "*magazzino di funzioni*".

In generale, gli elementi funzionali si distinguono in:

- Dispositivi logici combinatori;
- Dispositivi logici sequenziali;
- Operatori matematici;
- Dispositivi di comunicazione PLC/PLC;
- Dispositivi di comunicazione PLC/PC

I dispositivi logico combinatori e logico sequenziali sono:

- Ingressi esterni (per il PLC sono contatti NA);
- Uscite esterne (bobine di relè);
- Uscite di controllo interne (simulate nel PLC):
- Ritentive;
- Non ritentive;
- Uscite di controllo speciali:
- Disabilitazione uscite;
- Generazione di impulsi periodici;
- Reset del PLC;
- Blocchi funzione (temporizzatori, contatori…).

Capitolo 5

Programmazione: fasi principali

La prima fase per la programmazione di un PLC che deve gestire un qualunque impianto di automazione industriale od applicazioni diverse, è quella dell'ANALISI FUNZIONALE del problema da risolvere. In questa prima fase non interessa quello che sarà il contenuto del programma, il quale deve essere visto come una scatola nera di cui interessa sapere ciò che vi entra, le elaborazioni che deve effettuare e ciò che ne dovrà uscire. In pratica bisogna definire :

− Qual'è il punto di partenza, cioè quanti, quali e di che tipo sono i segnali in ingresso, con quali caratteristiche, con quale durata e con quale significato ;

− L'esatta definizione e descrizione del funzionamento dell'impianto anche nei minimi particolari, perchè in base a ciò si dovrà definire le elaborazioni che il PLC (ovvero il programma che deve essere scritto) dovrà svolgere.

− Qual'e' il punto di arrivo vale a dire i segnali che dovranno essere forniti al processo : come , quando, in che forma, con quale durata.

− A queste fasi segue quella della definizione della " Lista di Occupazione degli I/O " ; questa lista è importante per il programmatore perchè gli permette di sapere in quale determinato ingresso è collegato il tal sensore ed in quale uscita è collegato il talaltro attuatore. Il modo di come si descrive il funzionamento dell'impianto può essere diverso a seconda di come ognuno è abituato a lavorare, cioè si può ricorrere ad es. ad uno o più diagrammi di flusso, se il processo è costituito da più fasi di lavorazione, e quindi fare una descrizione di massima sulla relazione che esiste tra ogni fase e poi delle descrizioni dettagliate di ogni fase. Oppure si può fare una pura descrizione discorsiva sequenziale del funzionamento dell'impianto precisando dettagliatamente tutte le prescrizioni che devono essere rispettate e le varie modalità di comportamento richieste in situazioni particolari. In altri termini, bisogna dire cosa si deve fare, non come lo si vuole o lo si deve fare. Subito dopo la lista di occupazione degli I/O si può eventualmente fare anche una lista di assegnazione delle variabili interne (memorie a disposizione dell'utente) che si intende utilizzare per scopi particolari o comunque definibili già prima di iniziare la stesura del programma, come per es. quali memorie si utilizzeranno per memorizzare dei valori di conteggio, quali per memorizzare situazioni particolari del processo ecc. Solo dopo tutto questo lavoro di preparazione è possibile iniziare la stesura del programma che verrà fatta tenendo conto soprattutto di quanto è scritto nella descrizione del funzionamento dell'impianto in relazione alla successione temporale delle varie fasi. Dopo avere scritto il programma si può analizzare le possibilità di ottimizzarlo e minimizzarlo per renderlo il più efficiente possibile. E' importante inserire dei commenti e delle osservazioni che rendano il programma più chiaro e leggibile.

Definizione del sistema di campo (sistema controllato):

Tipo di operazioni

Modalità delle operazioni

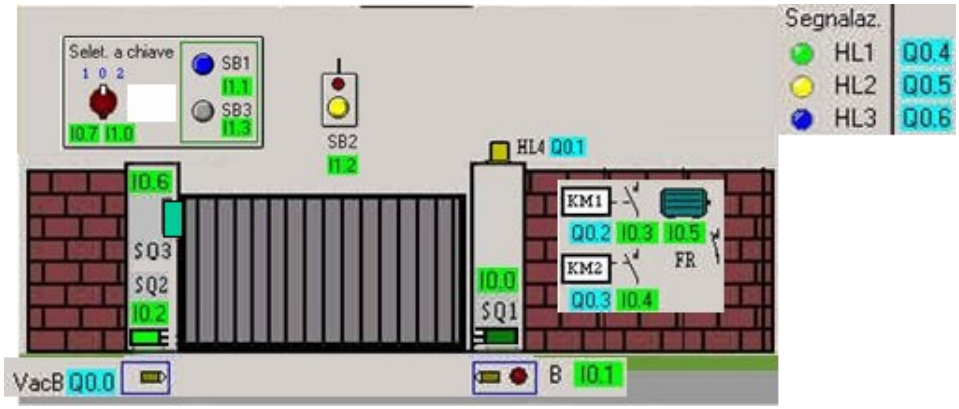
Numero e tipologia degli I/O

Assegnazione I/O / Layout sistema

Progetto n° 0 Automazione di un cilindro pneumatico

TABELLA DI ASSEGNAZIONE I/O

Simbolo	Indirizzo	Commento
INGRESSI		
USCITE	ESTERNE	
USCITE	INTERNE	(MERKER)
BLOCCHI	FUNZIONE	



Imputazione e verifica del programma con il linguaggio scelto

Capitolo 6

Linguaggi di programmazione

Programmare un PLC significa molto semplicemente trasferire in esso una sequenza di istruzioni (programma) in un linguaggio di programmazione opportunamente codificato, e di solito proprio del PLC che si sta utilizzando, tramite delle periferiche dedicate a questo scopo (unità di programmazione).

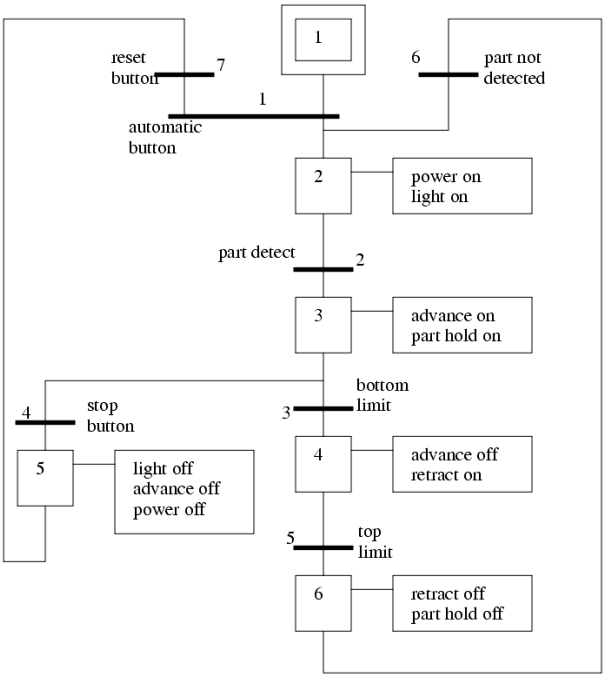
Il compito del PLC, come si è già visto, sarà quello di determinare lo stato delle uscite, ad esso collegate, in funzione dello stato degli ingressi, secondo le relazioni stabilite dal programma che il PLC esegue. IL PLC può essere programmato a svolgere una certa sequenza di operazioni, più o meno complessa, sulla base dell'insieme delle funzioni elementari da esso eseguibili (operazioni logiche, conteggi, comparazioni, temporizzazioni, etc), in altre parole sulla base del SET di istruzioni di cui è dotato. Tramite questo set di istruzioni, appartenenti ad un determinato linguaggio di programmazione, l'utente imposta la soluzione del problema sotto forma di programma, cioè di una lista di istruzioni appartenenti al set base del linguaggio. È del tutto ovvio, a questo punto, capire che disponendo di linguaggi di programmazione con un set base di istruzioni diverso, la soluzione dello stesso problema dipenderà dalle possibilità offerte dai diversi linguaggi.

Una volta che si sono acquisite le tecniche fondamentali di programmazione, non è comunque pensabile di passare da un linguaggio di programmazione di un PLC ad un'altro in maniera immediata ed automatica. Sarà sempre necessario avere una buona conoscenza del linguaggio di programmazione e quindi delle possibilità che questo offre, per poter sfruttare appieno le potenzialità del PLC stesso. Molti PLC ancora oggi posseggono un solo specifico linguaggio di programmazione. Ciò è del tutto comprensibile ed accettabile se si pensa ad un PLC appartenente alla cosiddetta fascia bassa, di prestazioni limitate, in grado di gestire un numero limitato di punti di I/O. Con il crescere della potenza del PLC, l'unicità del linguaggio di programmazione diventa un limite non più accettabile. Visto che il PLC deve operare in un contesto di automazione industriale, la sua capacità di espressione funzionale deve essere omogenea con le esigenze di automazione industriale. Basti pensare che i più elementari problemi da risolvere nell'ambito dell'automazione industriale sono del tipo : " se il tal contatto è chiuso e contemporaneamente quell'altro contatto è aperto, allora attiva la tal uscita".

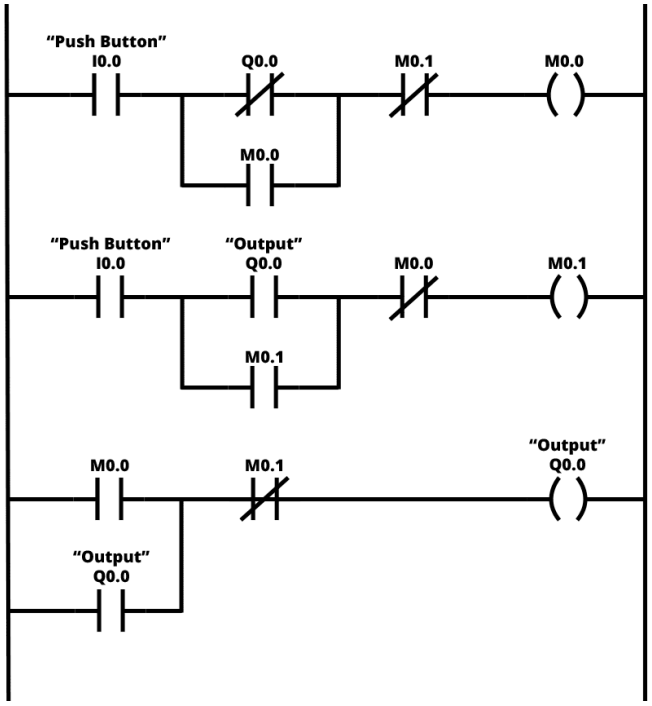
I linguaggi attualmente in uso si possono considerare appartenenti a due categorie ben precise, cioè quella dei linguaggi grafici e quella dei linguaggi letterali. La differenza fra questi due tipi consiste essenzialmente nella modalità di rappresentazione visiva delle combinazioni logiche che costituiscono le varie sequenze in cui è suddiviso un programma : in quelli grafici si fa uso di simboli grafici mentre in quelli letterali si fa uso di codici letterali mnemonici cui è attribuita una determinata funzione . È da precisare che vi sono due scuole di pensiero, quella americana e quella tedesca. Per la scuola americana, i linguaggi per PLC sono :

- Ladder diagram o linguaggio a contatti o linguaggio a relè od, ancora, schema a contatti;
- Boolean Mnemonics, o linguaggio booleano;
- Functional Block, o linguaggio a blocchi funzionali o diagramma funzionale o schema funzionale;
- HLL (High Level Language) o linguaggio ad alto livello

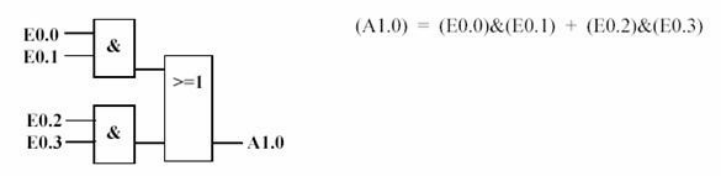
Esempio di linguaggio Grafcet



Esempio di linguaggio a contatti (LADDER)



Esempio di linguaggio booleano (Instruction List)



Capitolo 7

Linguaggio a contatti

Appartiene alla categoria dei linguaggi grafici perchè si presenta in maniera simile ad uno schema elettrico funzionale, con delle semplici modifiche.

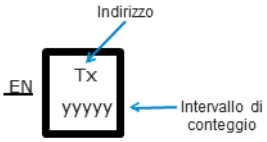
In questo tipo di linguaggio, per rappresentare lo stato dell'operando, viene valutato il livello logico della funzione e non il livello fisico del sensore collegato all'operando che è interessato (ad es. un ingresso in cui c'è presenza di tensione o assenza di tensione). Tra l'altro è da ricordare che l'operando di un'istruzione può essere una memoria o un'uscita.

CONTATTO NORMALMENTE APERTO di un relè

CONTATTO NORMALMENTE CHIUSO di un relè

BOBINA DI ECCITAZIONE di un relè

TEMPORIZZATORE



CONTATORE

Anche la forma che assume il programma deriva dalla logica a relè:

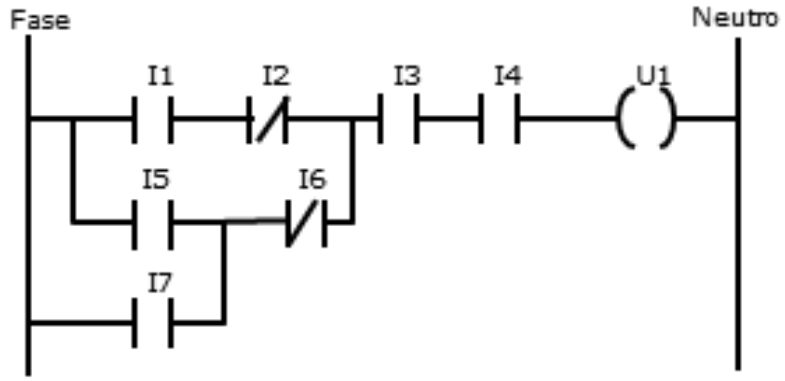
- Due linee verticalilaterali (montanti della scala) rappresentanti l'alimentazione.
- Le linee orizzontali (pioli della scala=rung) che alimentano una bobina se una certa combinazione di contatti abilita il flusso di energia.

I contatti possono essere associati:

- Agli ingressi digitali provenienti dal processo (o meglio al loro stato rappresentato in particolari bit della memoria)
- A condizioni interne al dispositivo (rappresentate da bit di memoria)

La bobina può essere associata a un bit della memoria e, col suo tramite, comandare un'uscita digitale o segnalare una condizione interna.

In un programma in linguaggio a contatti, il flusso di energia può andare sempre e solo da sinistra verso destra, senza possibilità di inversione:



La bobina contrassegnata con U1 potrebbe essere alimentata attraverso le sequenze:

– I1-I2-I3-I4

– I5-I6-I3-I4

– I7-I6-I3-I4

Ma non attraverso la sequenza:

– I7-I5-I1-I2-I3-I4 (che potrebbe rappresentare una continuità elettrica)

Con l'aumento delle potenzialità dei PLC, il loro insieme di istruzioni è stato quindi ampliato con una serie di funzioni che possono essere utilizzate all'interno di un programma in linguaggio a contatti, rispettandone la costruzione logica.

Una possibile divisione in categorie delle istruzioni potrebbe essere la seguente:

- Istruzioni di base (contatti e uscite di tipo relè).
- Istruzioni di temporizzazione e conteggio.
- Istruzioni per il controllo del programma.
- Istruzioni per la manipolazione di dati.
- Istruzioni per la realizzazioni di funzioni speciali.

Capitolo 8

Traduzione da SFC a Ladder

Chiariamo innanzitutto che questo linguaggio (o metodo) utilizza "Passi" (o step) per identificare la parte che esegue le azioni (i rettangoli nel grafico) e le transizioni per identificare il passaggio da un passo al successivo (tratti orizzontali).

Uno dei modi più semplici per ricreare in Ladder questo linguaggio è quello di assegnare un bit (variabile BOOL) ad ogni Passo e condizionare le azioni a questi Bit.

Le azioni non devono necessariamente essere inserite nella stessa parte di programma, ma scritte in altri blocchi di programma, a patto che le variabili utilizzate per rappresentare i passi siano di tipo Globale (pubbliche).

1) Inizializzazione impostando con bobine Set tutte le fasi iniziali dei SFC.

ad esempio, è il merker M0.1. In definitiva l'inizializzazione si riduce a un ramo con il contatto di prima scansione e la bobina Set della Fase iniziale (che è una memoria interna, insomma un merker, come tutte le fasi del tuo SFC).

2) Azioni: parallelo di tutti i contatti delle Fasi a cui un'azione è associata, quindi si chiude il ramo con la bobina relativa a quell'azione.

3) Transizioni: questa sezione realizza l'avanzamento della sequenza. Si tratta di una fila di rami, tutti con la medesima struttura, che è la seguente:

- contatto della fase attiva (partendo ovviamente dalla fase iniziale);

- logica della transizione (che può essere anche un solo contatto, es.: Start, se il pulsante Start è la tua condizione di transizione...)

- il ramo si chiude con il parallelo di due bobine: Set della fase successiva e Reset di quella attiva (in quest'ordine!!)

E così via per tutte le altre fasi... fino all'ultima che, solitamente reimposta la fase iniziale.

Ad ogni fase si associa un bit di memoria (marker di fase)

Ad ogni transizione si associa un bit di memoria (marker della transizione)

Quattro sezioni:

Sezione di inizializzazione

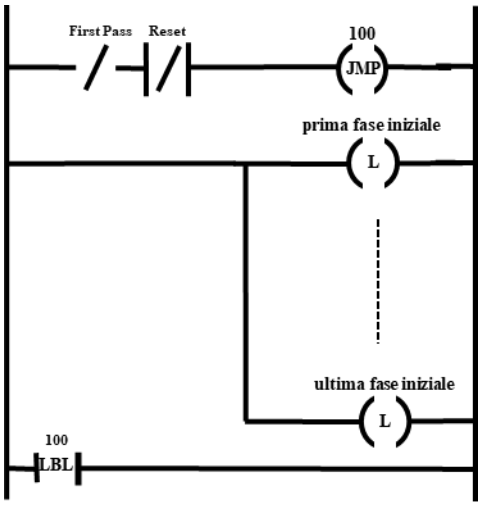
Sezione di esecuzione delle azioni

Sezione di valutazione delle transizioni

Sezione di aggiornamento della condizione

Inizializzazione

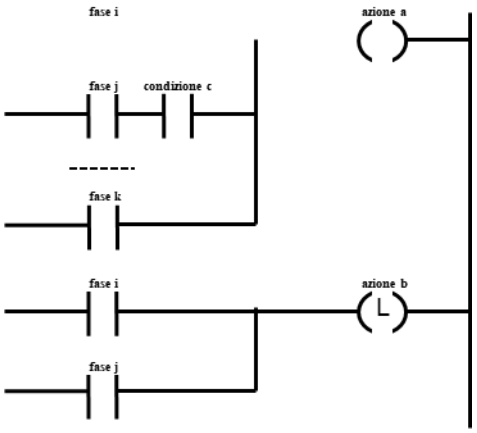
Eseguita una sola volta inizializza ad 1 i marker delle fasi iniziali



Esecuzione delle azioni

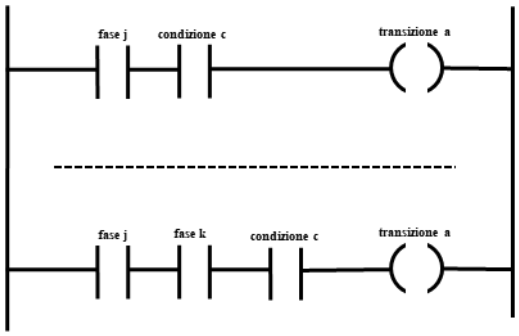
Per ogni azione continua si prevede un rung di abilitazione su cui si trovano in OR tutte le fasi che implicano quella azione eventualmente in AND con delle condizioni

Per le azioni memorizzate si userà LATCH/UNLATCH bobina



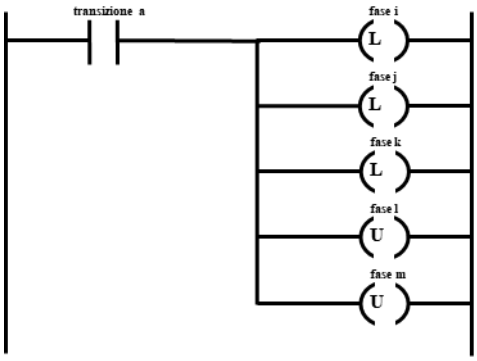
Valutazione delle transizioni

Il marker di ogni transizione è attivato se la transizione è superabile si ha un rung per ogni transizione



Aggiornamento condizione

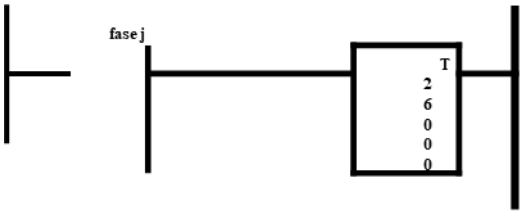
In corrispondenza delle transizioni attivate spegne i marker delle fasi a monte ed accende quelli delle fasi a valle



Traduzione delle variabili temporali

Le variabili temporali devono essere realizzate con dei temporizzatori senza ritenzione attivati dall'accensione dei marker di fase

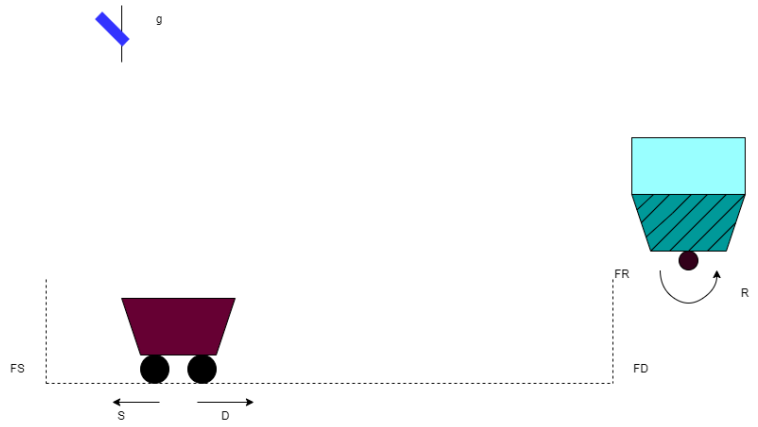
Questi temporizzatori possono essere messi in una sezione a parte



Capitolo 9

Esempi di controllo

Carrello



Descrizione del proplema :

Il carrello trasportatore riportato nella slide precedente il quale ,alla richiesta di un operatore, si sposta alla estrama destra di un binaria, viene caricato mediante il ribaltamento di un serbatoio, e si riporta a sisnistra per consentire lo scarico del matriale. Il Sistema comprende tre segnali da sensori finecora, due per il carrello, FS e FD, e uno per il serbatoio, FR; un segnale da una leva di attivazione ciglo g; due comandi di attuazione per il moto del carrello, S e D, e uno per il ribaltamento per serbatoio,R.

Il programma in SFC

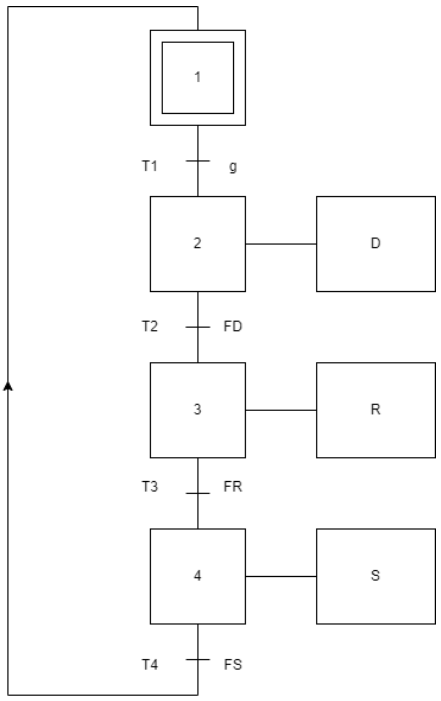
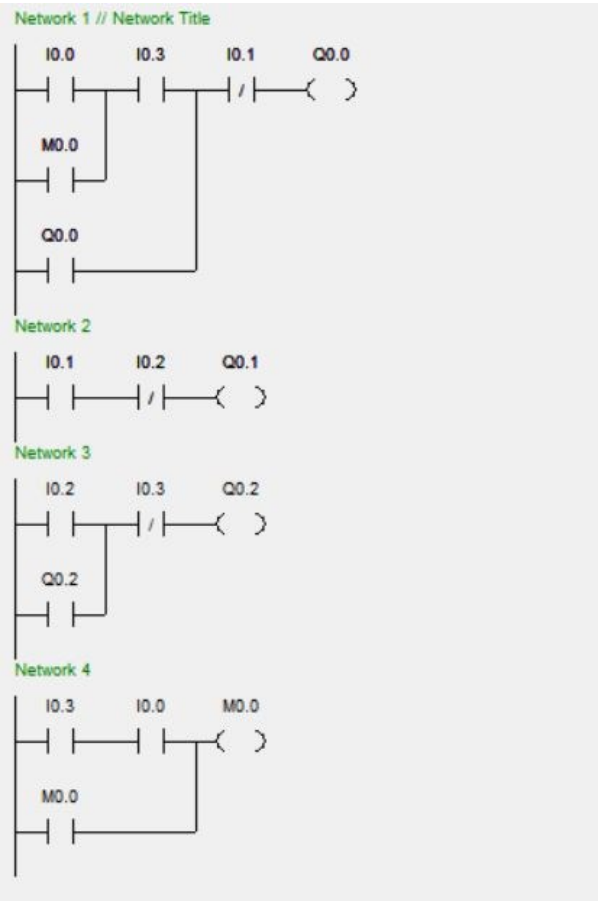


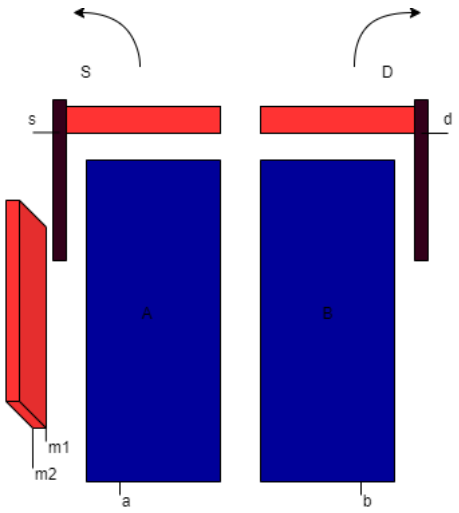
Tabella delle variabili

	Simbolo	Indirizzo	Commento
1	g	I0.0	Pulsante (NO) Start ciclo automatico
2	FD	I0.1	Sensore (NO) indica la posizione del carrello a destra
3	FR	I0.2	Sensore (NO) indica la posizione del serbatoio
4	FS	I0.3	Sensore (NO) indica posizione del carrello a destra
5	#	M0.0	Marker per farsi che il ciclo rimane continuo
6	D	Q0.0	Bobina eccitazione elettrovalvola per il moto del carrello verso destra
7	R	Q0.1	Bobina eccitazione elettrovalvola per il ribaltamento del serbatoio
8	S	Q0.2	Bobina eccitazione elettrovalvola per il moto del carrello verso sinistra

Il programma in ladder



Parcheggio



Descrizione del proplema :

Si consideri il Sistema per l'entrata a pagamento in un parcheggio schemattizzato in fin . La barriera di ingress e' composta da due barre: la barra di sinistra si puo' apprire da sola per far passare un motoveicolo; le due barre si possono aprire insieme per far passare una automobile. Se gli attuatori delle due barre non sono alimentati le barre si richiudono con un meccanismo a molla.. Sulla sinistra vi e' una gettoneria con due feritoie per il passaggio di monete di 100 e 200 lire, sul pavimento due placche con celle di carico per rilevare la presenza di un motoveicolo (segnale dalla sola placca A) o di una automobile (segnali dale placche A e B).

Il Sistema di controllo ha a disposizione I seguenti siegnali d'ingress di tipo digitale :

a	presenza di veicolo su placca A
b	presenza di veicolo su placca A
m1	passaggio moneta da 100 lire
m2	passaggio moneta da 200 lire
s	barra di sinistra chiusa
d	barra di destra chiusa

I segnali di comando, di tipo digitale, sono:

S	apri la barra di sinistra
D	apri la barra di destra

Per far aprire la sola barra di sinistra un moto veicolo deve essere sulla sola placca A e deve essere inserita (almeno) una moneta da 100 lire; la barriera si richude quando il veicolo non e' piu' sulla placca A.

Per far aprire entrambe le barre un'auto si deve portare sulle placche A e B e devono essere inserite due monete da 100 lire oppure da 200 lire (almeno); la barriere si richiude quando non vi e' la presenza dell'auto sulla placche. Si assume che un viecolo si appoggia prima sulla placca A deve impegnare la placca B entro un secono per essere considerato un auto, altrimenti e' considerato un moto veicolo. Le barre devono essere entrambe chiese prima di poter trattare un nuovo veicolo.

Il programma in SFC

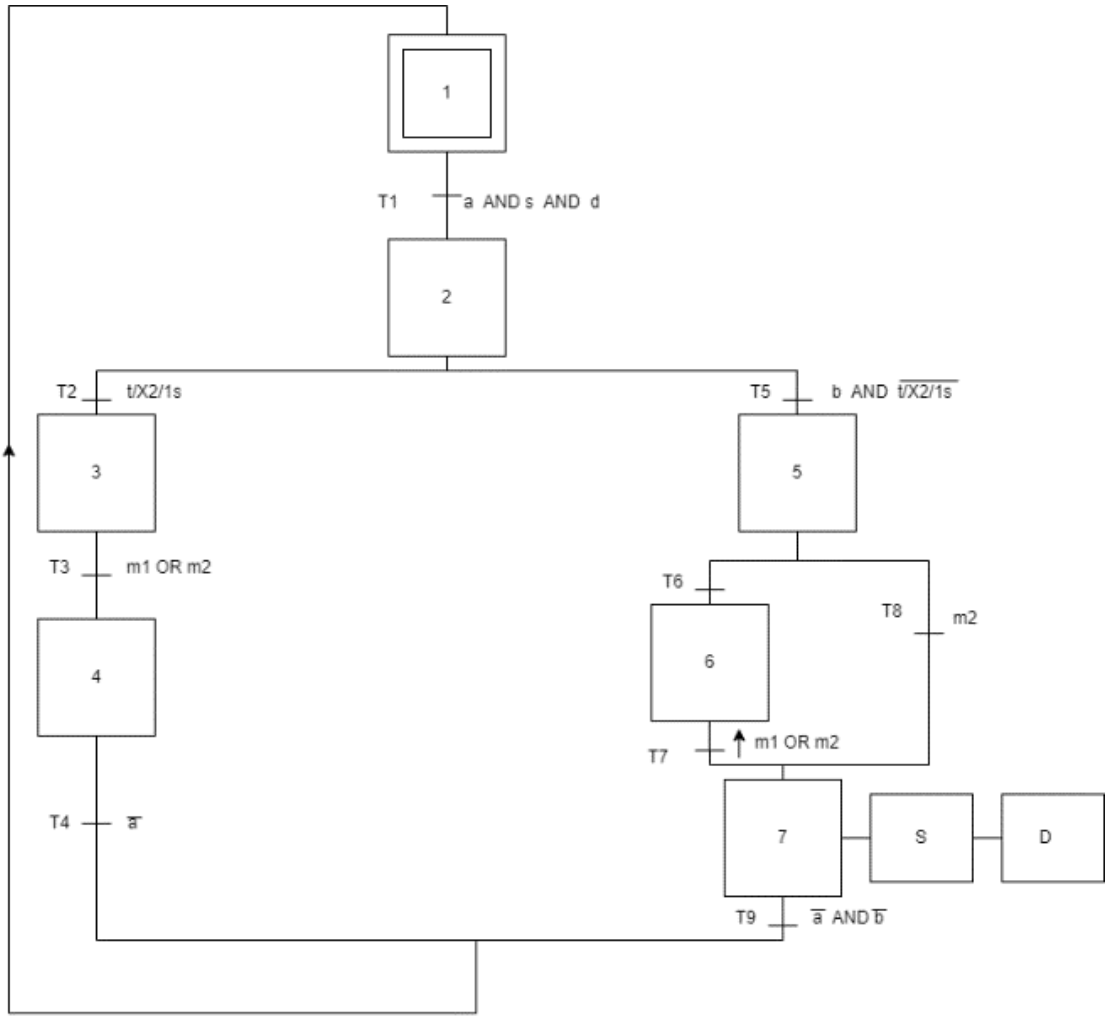
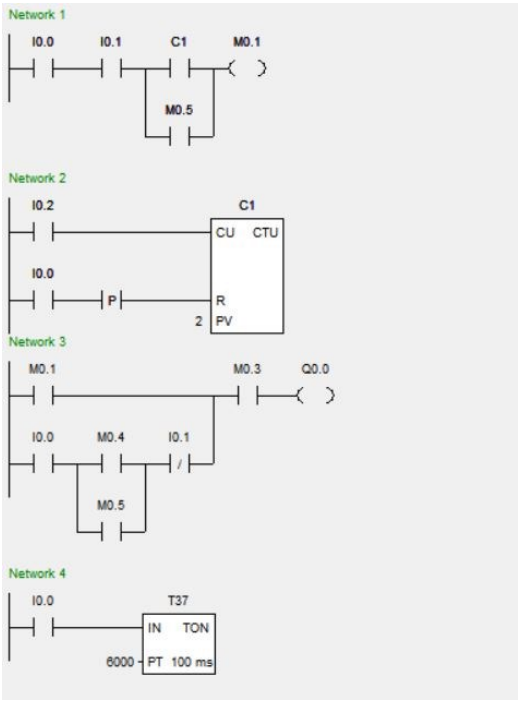
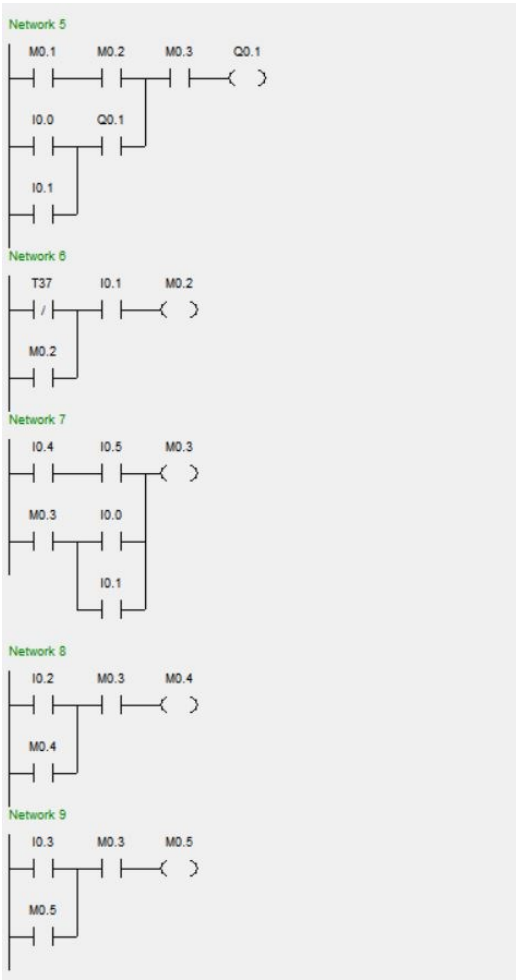


Tabella delle variabili

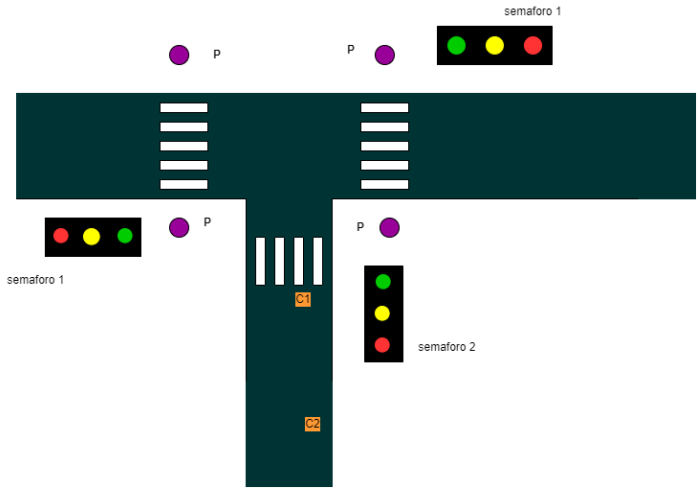
	Simbolo	Indirizzo	Commento
1	a	I0.0	Sensore (NO) indica la presenza di veicolo su placca A
2	b	I0.1	Sensore (NO) indica la presenza di veicolo su placca B
3	m1	I0.2	Sensore (NO) indica il passaggio moneta cento lire
4	m2	I0.3	Sensore (NO) indica il passaggio moneta 200 lire
5	s	I0.4	Sensore (NO) indica la barra di sinistra è chiusa
6	d	I0.5	Sensore (NO) indica la barra di destra è chiusa
7	#	M0.1	Marker per far alzare le due barre
8	#	M0.2	Marker per dire che questo un'auto o meno
9	#	M0.3	Marker si attiva quando si chiudono le due barre
10	#	M0.4	Marker che funziona quando cento lire e due barre chiuse
11	#	M0.5	Marker che funziona quando cento lire e due barre chiuse
12	S	Q0.0	Bobina eccitazione elettrovalvola per aprire la di sinistra
13	D	Q0.1	Bobina eccitazione elettrovalvola per aprire la di destra

Il programma in ladder





Semaforo



Descrizione del proplema :

Si abbia l'incorocio di strade regolato da Sistema simaforico rappresentato in fig

I segnali di ingress disponibili, di tipo digitale, sono:

P	pulsante richiesta pedonale
C1	rilevatore presenza continua auto
C2	rilevatore presenza continua auto

I comandi disponibili, di tipo digitale, sono:

V1	luce verde del semaforo 1
G1	luce gialla del semaforo 1
R1	luce rossa del semaforo 1
V2	luce verde del semaforo 2x
G2	luce gialla del semaforo 2
R2	luce rossa del semaforo 2

Il semaforo 1 e' posto su una strada di grande traffic e deve essere normalmente verde. Il ciclo semaforico, descritto piu' avanti si deve attivare :

1)solo se il verde del semaforo 1 e' stato presente per almeno 300 secondo

2)immediatamente, se entrambi I relevatori C1 e C2 segnalano la presenza continua di auto (coda al semaforo 2)

3)dopo 30 secondi, se vi e' la richesta da parte di un pedone

4)dopo 240 secondi, se il relevatore C1 segnala la presenza continua di un auto

L’ordine di presentazione delle quattro specifiche e’ anche quello di priorita’ che deve essere rispettato nel loro soddisfacimento.

Il ciclo semaforico e’ quello usuale compost, per ogni semaforo, da una successione verde-giallo-rosso-verde. Il giallo deve durare cinque secondi mentere il verde del semaphore 2 sessanta secondi

Il programma in SFC

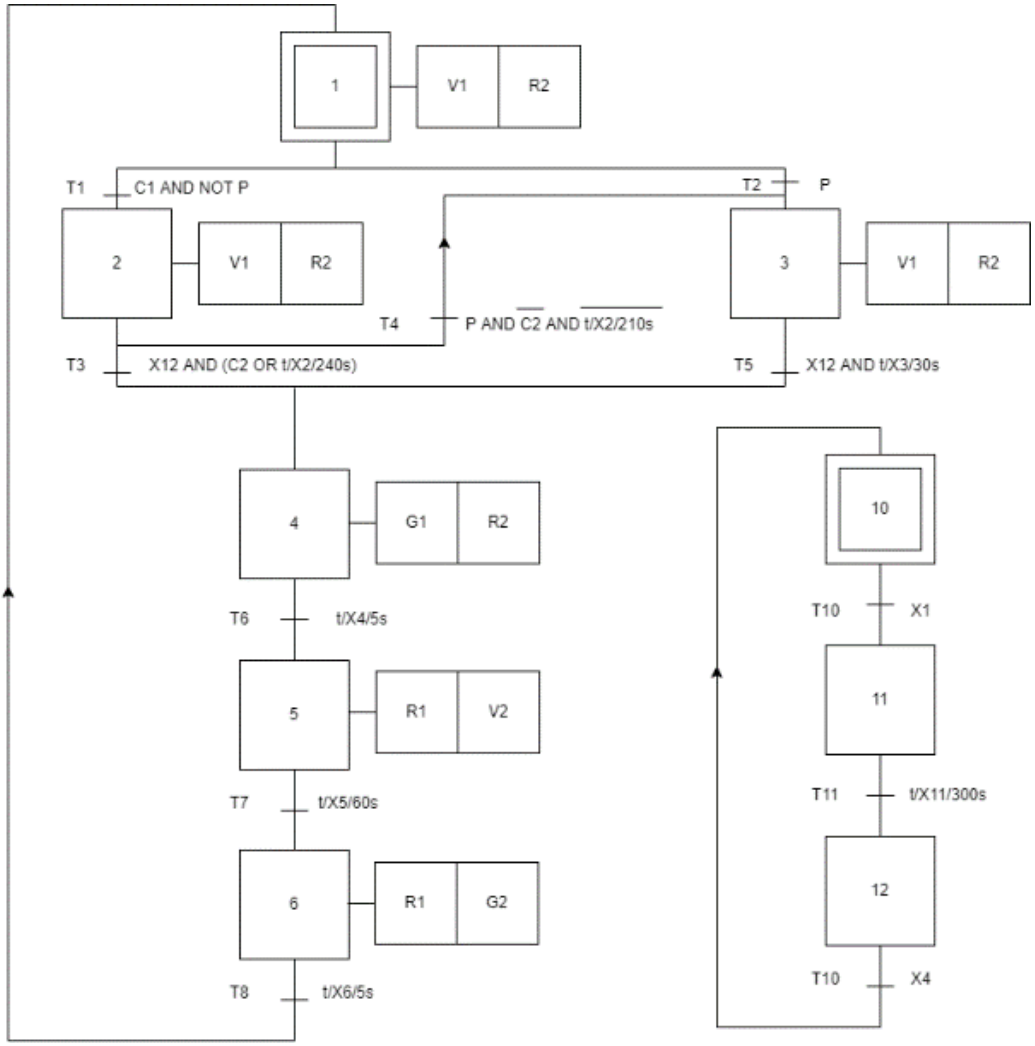
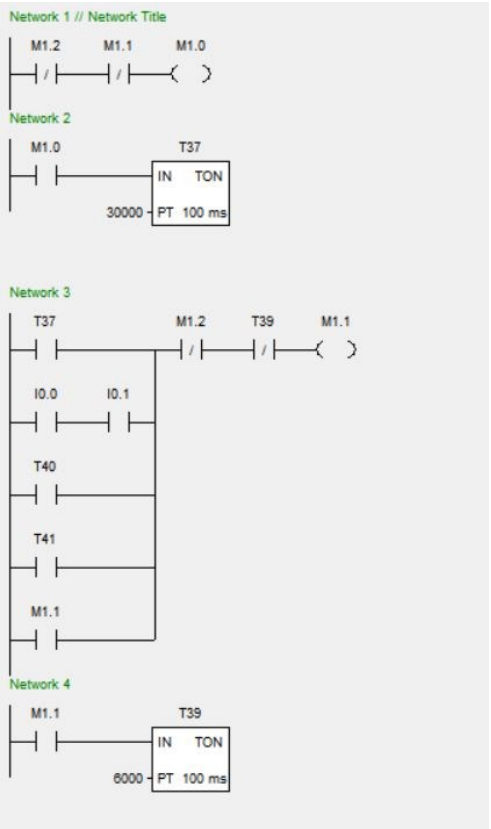


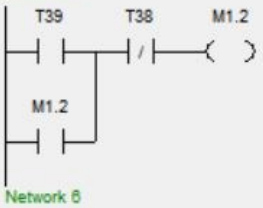
Tabella delle variabili

	Simbolo	Indirizzo	Commento
1	C1	I0.0	Sensore (NO) indica l'esistenza di auto in C1
2	C2	I0.1	Sensore (NO) indica l'esistenza di auto in C2
3	P	I0.2	Pulsante (NO) richiesta pedonale
4	#	M0.0	Marker per il pulsante di pedone
5	#	M1.0	Marker per verde 1 rosso 2
6	#	M1.1	Marker per verde 2 rosso 1
7	#	M1.2	Marker per giallo per 1 e 2
8	V1	Q0.0	Bobina per la luce verde del semaforo 1
9	V2	Q0.1	Bobina per la luce verde del semaforo 2
10	G1	Q0.2	Bobina per la luce gialla del semaforo 1
11	G2	Q0.3	Bobina per la luce gialla del semaforo 2
12	R1	Q0.4	Bobina per la luce rossa del semaforo 1
13	R2	Q0.5	Bobina per la luce rossa del semaforo 2

Il programma in ladder

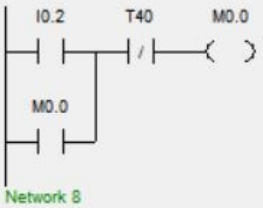


Network 5

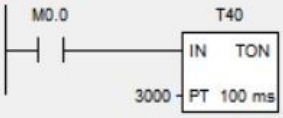


Network 6

Network 7

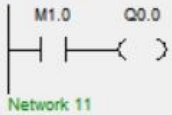


Network 8

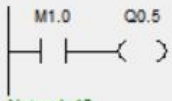


Network 9

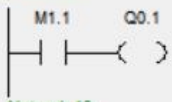
Network 10



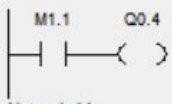
Network 11



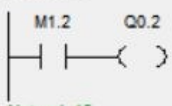
Network 12



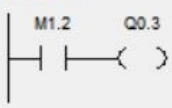
Network 13



Network 14



Network 15



Capitolo 10

Test e simulazione

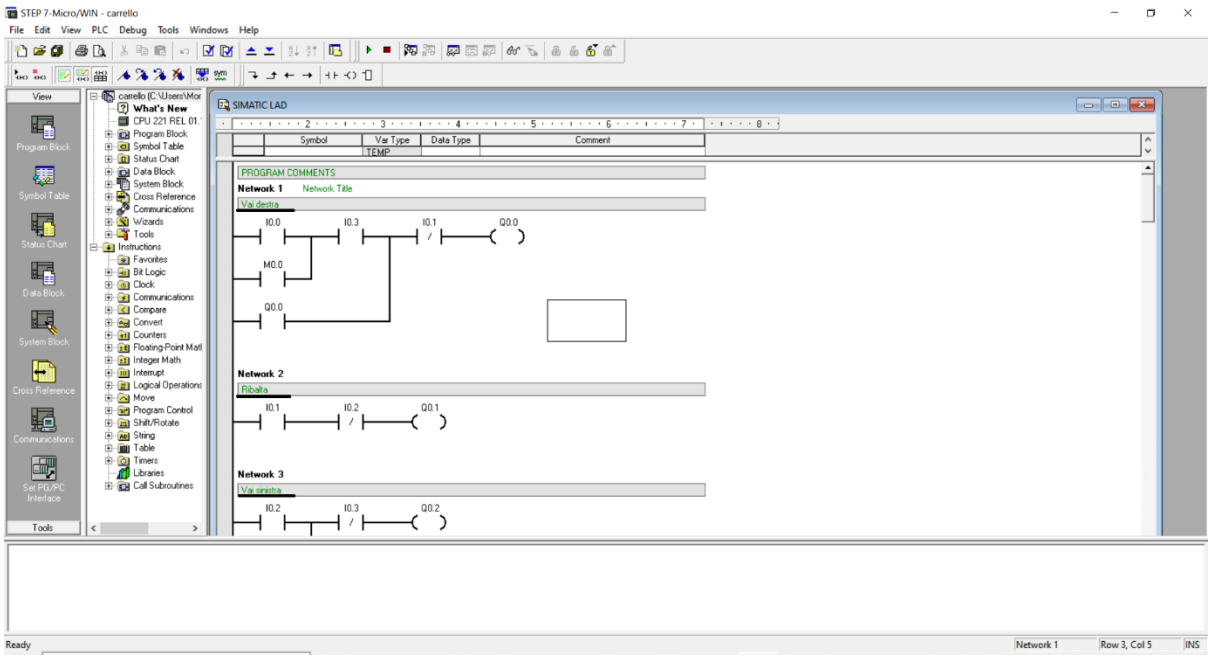
Dopo che il programma è stato sviluppato, bisogna passare alla fase di correzione e messa a punto dello stesso. Ciò può essere fatto trasferendo il programma nella memoria del PLC ed eseguendolo simulando l'azionamento dei sensori tramite un opportuno simulatore. La simulazione permette di analizzare il comportamento del PLC di fronte alle varie situazioni che si possono presentare nella realtà e verificare quindi se il programma è rispondente con quanto si vuole fare. Dopodichè si può passare alla fase successiva che è quella della documentazione dei programmi.

Per effettuare questo test ho utilizzato due programmi il primo per scrivere il codice in ladder (V4.0 STEP 7 MicroWIN SP9) si puo' inistallare dalla cartella ladder e l'icona

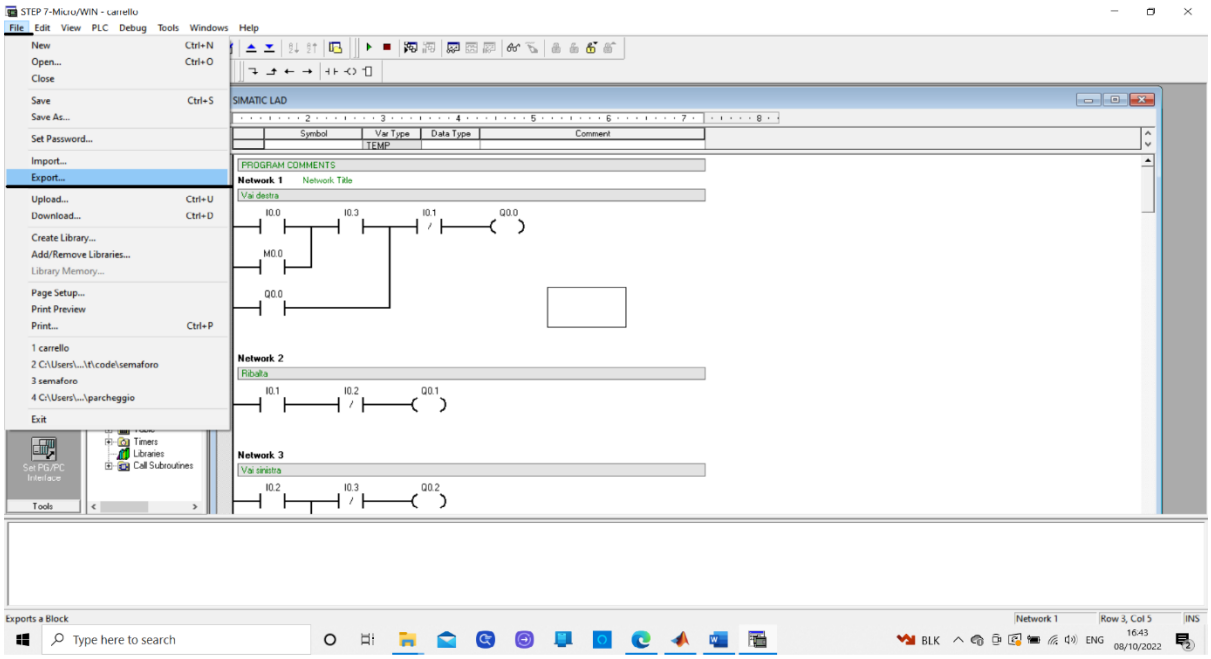
Il secondo programma che funziona come un simulatore del PLC (S7-200)si puo' inistallare dalla cartella PlC simulatore e ha l'icona

Procedimenti per il test

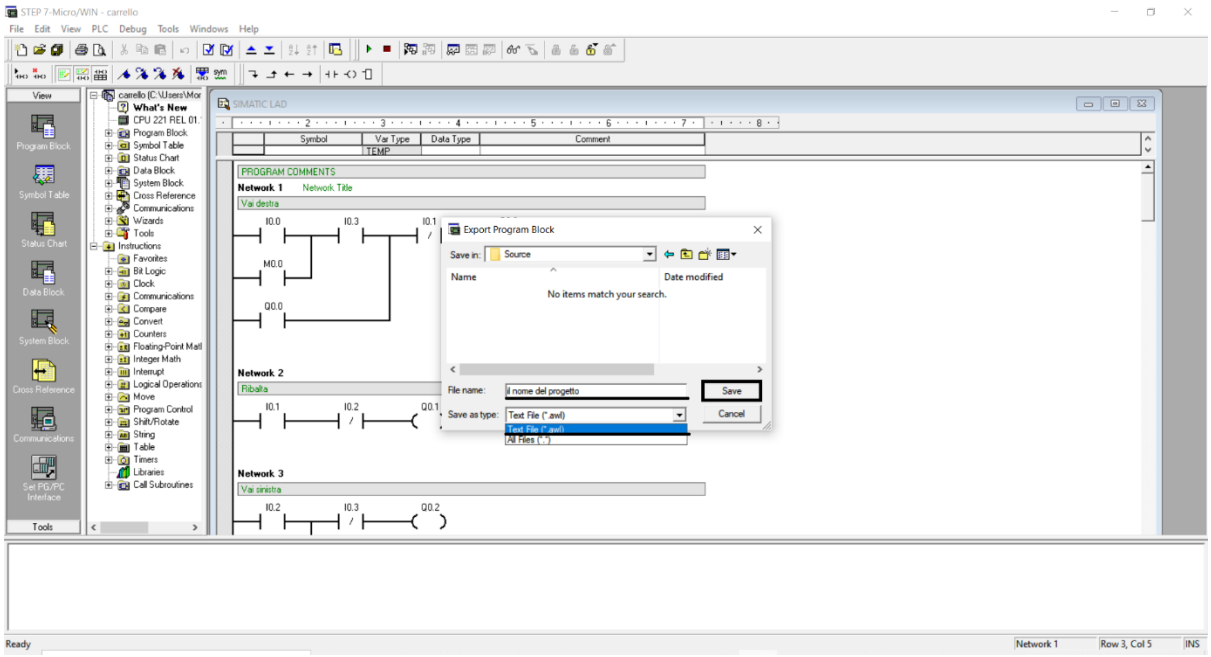
1) Scrivi il codice ladder che sara' fatto come questo esmpio (essensiale metterre i commenti).



2) Apri il menu File e scegli la voce Export.

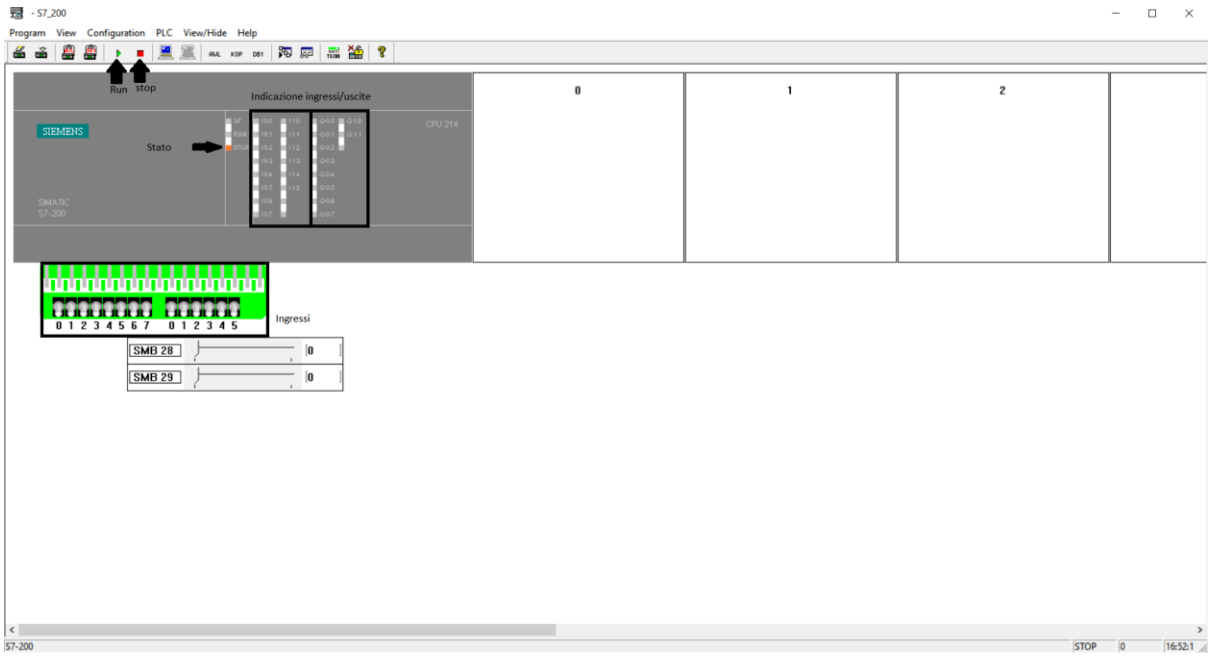


3) Scrivi il nome del file e salvalo in formato “awl” nella nuova finestra.

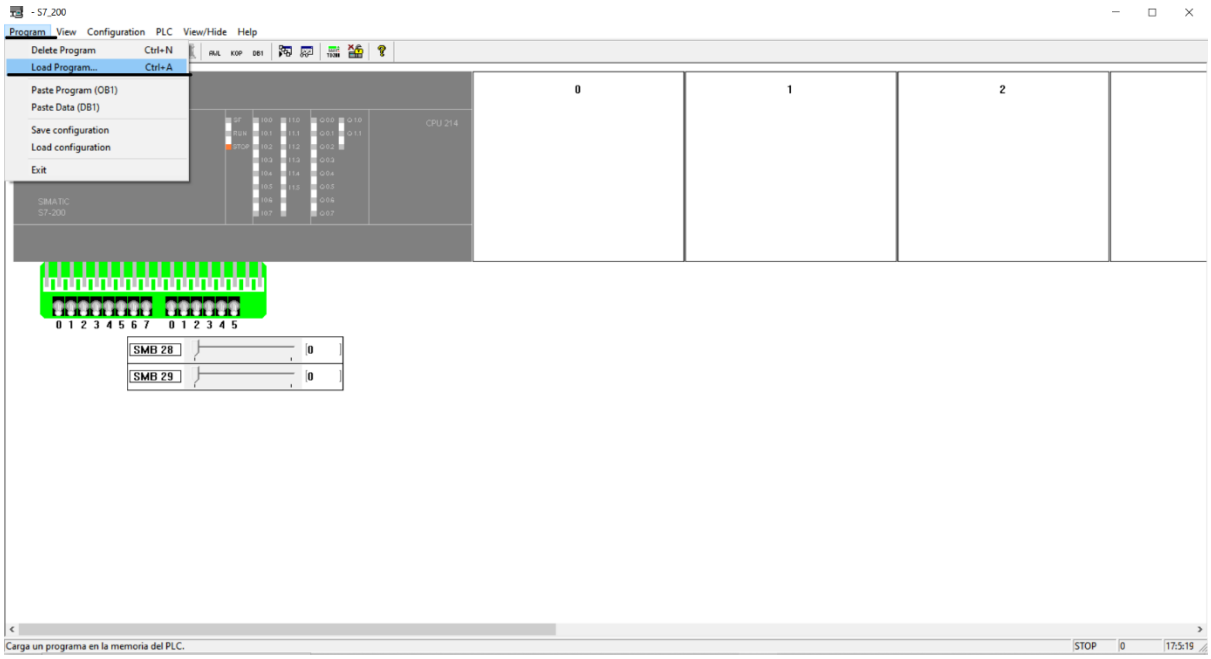


4) Apri il programma del simulatore.

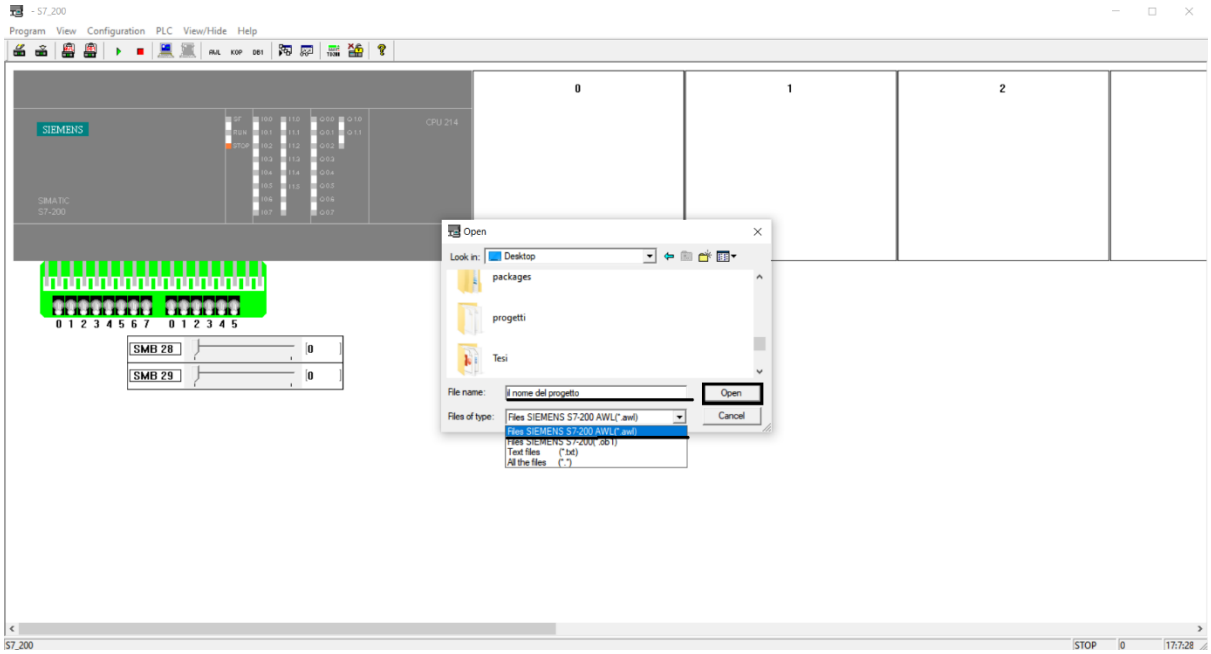
NB:Per caricare il programma lo stoto del PLC deve essere Stop.



5) Apri il menu Program scegli la voce LoadProgram.



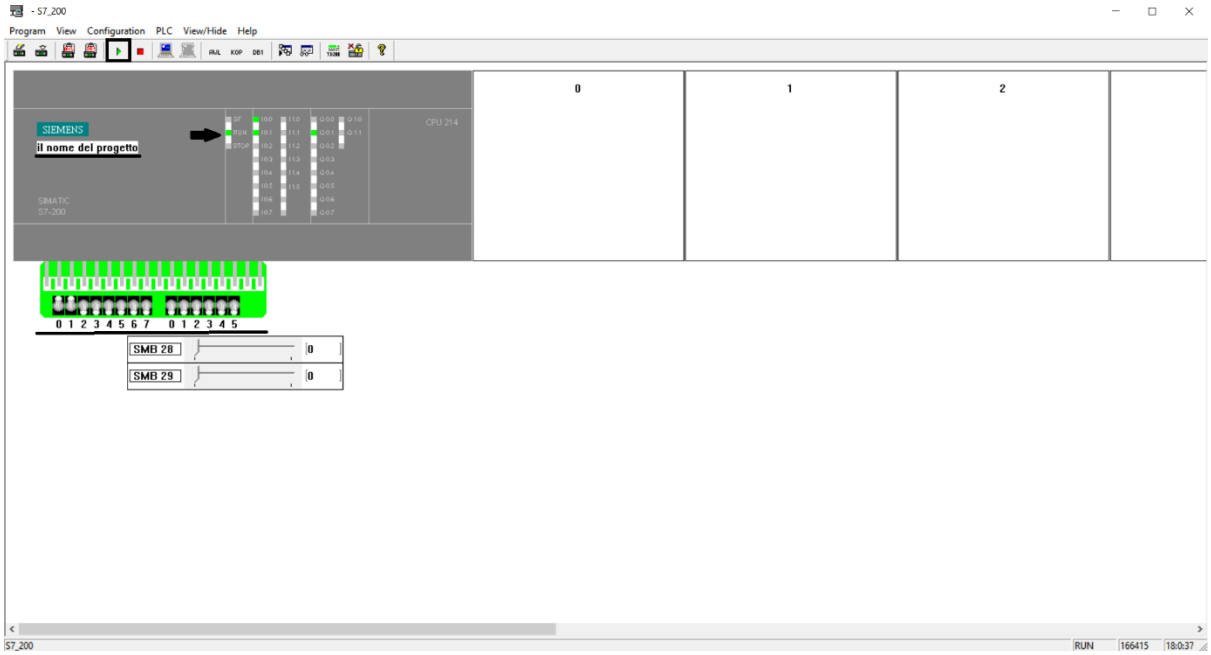
6) Metti il nome del file in formato sempre “awl” nella nuova finestra.



7) Ultimo passo Porta il PLC in modalita’ Run per testare il processo

Noti che il nome del programma e’ stato caricato

In questo esempio e gli ingressi I0.0 e I0.1 sono attivi questo va ad attivare la bobina Q0.0.



Capitolo 11

Conclusione

Nel presente lavoro di tesi è stato affrontato il controllo di sistemi automatici come Carrello, Parcheggio e semaforo implementando degli algoritmi di controllo scritti in Ladder. facendo anche delle simulazioni su un PLC virtuale, per arrivare a tale scopo sono partito dalla definizione del PLC per sapere come è fatto sia dalla parte dell'architettura conoscendo i componenti Hardware che dalla parte software conoscendo gli elementi funzionali, le funzioni integrate e come vengono programmati attraverso i vari linguaggi di programmazione. Una volta che si sono acquisite le tecniche fondamentali di programmazione per iniziare a codificare il problema, prima bisogna definire il campo dello specifico sistema automatico (Carrello o Parcheggio oppure Semaforo) e assegnare I/O e il Layout del sistema che mi era sato molto utile e poi come ultima cosa è fare il programma in Ladder e poi testarlo utilizzando il simulatore cosi per verificare il codice e modificare le parti non funzionanti, provare tutte le casistiche possibili e renderlo più efficiente.

Bibliografia

Dispensa sui PLC per gli allievi versione PDF (malignani.ud.it)

DISPENSA PLC - I SEGRETI DEGLI IMPIANTI ELETTRICI (fabrizio-urlando.cloud)

PLC (unict.it)

Microsoft PowerPoint - Introduction to PLCs.ppt (uned.es)

Dispense Informatica Industriale (unict.it)

Parameterization (uniroma3.it)

Le slide per Prof. Gianluca Ippoliti

IL libero PLC e automazione industriale di Chiacchio, Pasquale