

CORSO DI Elettrotecnica e Automazione
LABORATORIO DI TDP

PLC

CONTROLLORE MODICON TSX MICRO

SOFTWARE PL7

PLC

Programmable Logic Controller

Controllore a Logica Programmabile

Generalità

Il PLC è un sistema elettronico che utilizza una memoria programmabile per elaborare informazioni e istruzioni che servono ad attuare funzioni specifiche di controllo e automazione quali controlli sequenziali, logica combinatoria, temporizzazioni, conteggi, calcoli matematici.

Il PLC, come un computer, utilizza un microprocessore(CPU) per coordinare tutte le operazioni interne. Il computer è progettato per risolvere problemi non specifici e quindi si dice che è di “uso generale”, è capace di elaborazioni molto potenti(può spostare più di 500000 KByte/sec; utilizza programmi di dimensioni > 10000 Kbyte; effettua operazioni binarie con 32 bit) e nei processi di automazione è utilizzato come supervisore in quanto è capace di controllare decine di PLC. Il PLC è un computer dedicato, con particolari caratteristiche come l'elevata immunità ai disturbi elettrici, la capacità di svolgere sequenze di funzionamento ripetitive 24 ore su 24, la capacità di lavorare in condizioni ambientali difficili(0÷55°C), la facilità di programmazione , e quindi adatto a risolvere problemi di controllo e automazione dell'impianto in ambienti industriali(per contro ha una bassa capacità di elaborazione: capacità di spostamento dati < 10Kbyte/sec; utilizza programmi di dimensioni < 10 Kbyte; effettua operazioni binarie su un 1 bit singolo).

Il PLC trova impiego anche in ambienti civili (domotica) per il controllo locale e/o a distanza dell'illuminazione, degli accessi, del riscaldamento, ecc.

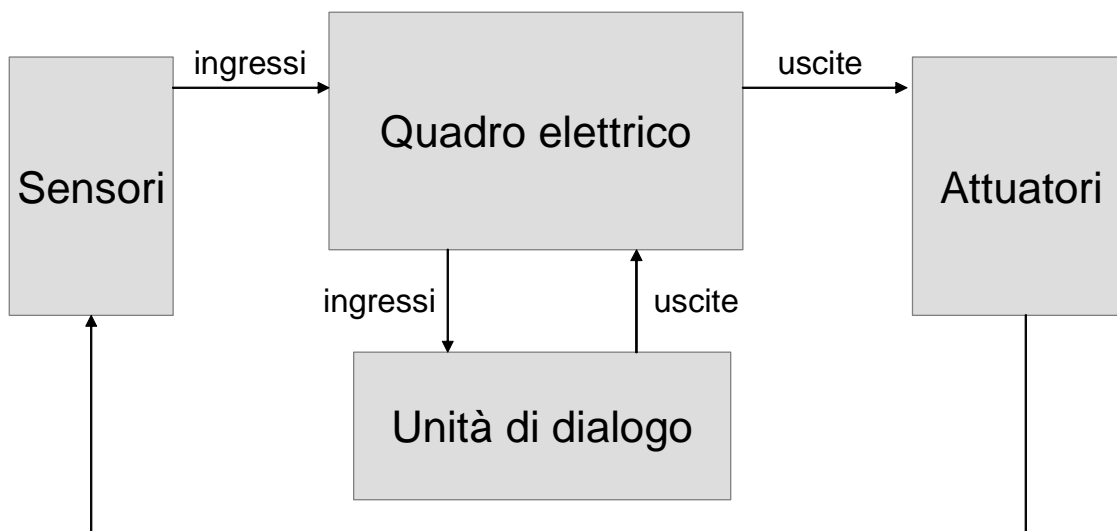
Caratteristiche del PLC:

- è **versatile** perché impiega programmi che si prestano a risolvere problemi su diversi impianti; inoltre vi è la possibilità di scelta tra *diversi linguaggi* per la compilazione del programma;
- ha un elevato grado di **flessibilità** in quanto la logica funzionale, contenuta nel programma memorizzato sul controllore, può essere facilmente modificata a seconda delle esigenze senza dover modificare alcun cablaggio: una modifica della logica di funzionamento con un PLC avviene in modo semplice e veloce modificando il programma, mentre con un quadro di comando a logica cablata richiede una difficile e costosa modifica del cablaggio (inoltre in un quadro di comando a logica cablata è difficile la localizzazione dei guasti, in caso di guasto l'impianto può fermarsi per molto tempo, vi è ingombro e assorbimento elevato per i grossi impianti, la durata delle apparecchiature limitata nel tempo poiché vi sono organi in movimento);
- è completamente statico, di minimo ingombro e molto **affidabile** perché è progettato per poter lavorare in ambienti difficili come quelli industriali e ha una elevata sicurezza di funzionamento grazie all'uso di sofisticate procedure diagnostiche;

- il *cablaggio si riduce* al solo collegamento al PLC dei sensori in ingresso e degli attuatori in uscita, eliminando nel caso dell'applicazione elettromeccanica, relè ausiliari, contatti, temporizzatori, ecc., la cui funzione è sostituita dal programma in memoria;
- mette a disposizione, contrariamente ai sistemi di controllo elettromeccanici, un numero di relè, temporizzatori, contatori, ecc., molto alto e per ognuno di essi moltissimi contatti, il cui numero è limitato solo dalla quantità di memoria disponibile sul controllore;
- è *interfacciabile* con stampanti, terminali video, p. c. e consente un controllo di tipo on line dell'impianto che permette di apportare modifiche al programma della macchina o dell'impianto in qualsiasi momento e stampare dati, messaggi e tutta la documentazione relativa.

Struttura di un impianto automatizzato

Qualsiasi impianto automatizzato è strutturabile nel seguente modo:



Un automatismo può essere realizzato con due tecniche: *logica cablata*, *logica programmabile*.

La *logica cablata*, qualunque sia la tecnologia utilizzata(elettromeccanica, elettronica, pneumatica, elettropneumatica)consiste nel collegare, fisicamente, gli elementi che costituiscono l'automatismo(teleruttori, relè ausiliari, temporizzatori, schede elettroniche, valvole, elettrovalvole, sensori).

Nella logica cablata elettromeccanica il quadro elettrico è costituito da :

- elementi di controllo(relè ausiliari, temporizzatori, contatori, ecc.),
- azionamenti(contattori),
- morsettiere,
- cablaggio, ecc;

l'unità di dialogo comprende:

- elementi di controllo quali i pulsanti, inoltre possono esserci uscite come le lampade di segnalazione(quadro elettrico e pulsanti possono essere cablati nello stesso armadio o in due posti diversi.),
- oppure può essere un quadro di comando sinottico.

La *logica programmabile* consente di realizzare gli automatismi sostituendo i relè (o le schede elettroniche o le elettrovalvole) e i relativi collegamenti con un circuito elettronico formato da celle di memoria, e con un software.

(le celle di memoria sono bit che possono assumere due stati: 1 o 0; un bit = 1 equivale ad un relè eccitato, un bit = 0 equivale ad un relè diseccitato).

Nella logica programmata il quadro elettrico comprende l'hardware:

- alimentatore,
- PLC,
- azionamenti(contattori),
- morsettiere, ecc.;

l'unità di dialogo può comprendere:

- l'unità di programmazione (tastiera alfanumerica con visualizzatore oppure p. c. e relativo software),
- i pulsanti ed eventualmente anche le uscite per le segnalazioni(anche qui uscite e pulsanti possono essere cablati nello stesso armadio del quadro elettrico o in due posti diversi),
- oppure un quadro sinottico.

Un PLC può sostituire un quadro di comando a logica cablata:

La *logica di funzionamento* in un quadro cablato viene stabilita dai collegamenti dei fili,dal cablaggio,appunto,dei vari componenti elettrici.

Nel PLC la *logica di funzionamento* viene stabilita dal programma, che è una sequenza di istruzioni che il suo microprocessore deve compiere.

Tra le istruzioni che un microprocessore può elaborare vi sono anche le operazioni della logica binaria: ad esempio l'operazione logica AND che equivale a un collegamento in serie di contatti, l'operazione logica OR che equivale a un collegamento in parallelo di contatti. Ciò significa che un PLC può essere usato per elaborare segnali elettrici per il comando di una macchina alla stessa stregua di un circuito di comando elettromeccanico.

Allora qualsiasi schema funzionale della logica cablata elettromeccanica, elettronica, elettropneumatica, può essere tradotto in una sequenza di istruzioni, cioè in un programma per un PLC che svolga la stessa funzione.

Classificazione dei PLC

Con riferimento alla quantità di ingressi e uscite che gestisce, un PLC si può classificare:

- di *gamma(o taglia) bassa* quando può controllare un numero massimo di *64 ingressi/uscite*,
- di *gamma media* quando può controllare fino a *512 ingressi/uscite*,
- di *gamma alta* quando può controllare fino a *1024 ingressi/uscite*.

Se occorre gestire un numero di ingressi/uscite maggiore si ricorre a più PLC collegati fra di loro.

In base al criterio costruttivo i PLC sono:

- *monoblocco*
- *modulari*.

Il tipo monoblocco, o compatto, si presenta con alimentatore, processore e un modulo d'ingresso/uscita assemblati in un unico blocco.

Esso ha una configurazione degli ingressi e delle uscite rigida, cioè non modificabile, se non in qualche caso in cui all'unità base si può aggiungere una unità di espansione anch'essa rigida.

In genere i PLC compatti sono di gamma bassa.

Il tipo modulare ha una struttura costituita da un rack(rastrelliera, cestello) in cui sono inseriti *un modulo base*, contenente alimentatore e processore, e *più moduli d'ingresso/uscita* che trovano alloggio negli slot (fessura) del rack stesso. I moduli sono collegati fra di loro attraverso il bus del rack. Per evitare errori di posizionamento dei moduli, ogni modulo ha un apposito accoppiatore codificato maschio-femmina che non permette un inserimento errato. Il PLC modulare può avere diverse configurazioni a seconda del tipo di alimentazione (continua o alternata), del processore usato, e delle schede I/O impiegate.

Dal punto di vista dell'impiego i PLC si classificano in:

- *sequenziali*
- *multifunzione*

I PLC sequenziali possono essere compatti o modulari, di taglia piccola o media, e oltre a essere impiegati negli automatismi che funzionano secondo una logica sequenziale (e quindi sostituire le funzioni dei quadri cablati) possono: avere contatori veloci; elaborare numeri binari; fare calcoli matematici.

I PLC multifunzione sono in genere modulari e di gamma alta e offrono ulteriori prestazioni: misura ad es. di temperatura, di velocità, ecc. ; regolazione ad es. della velocità di un motore elettrico; controllo assi (regolazioni di velocità e di posizione); dialogo tra PLC e PLC o tra PLC e periferiche.

Criteri di scelta

Quando la logica da realizzare non è complessa, non si prevedono modifiche o ampliamenti dell'impianto conviene usare un quadro cablato; in caso contrario, dati gli elevati costi di cablaggio e manutenzione, conviene usare il PLC. Comunque, per scegliere il controllore che meglio si adatta alle proprie esigenze è opportuno seguire alcuni criteri di massima: *criteri funzionali, criteri tecnologici, criteri operativi, criteri economici.*

Criteri funzionali

permettono di scegliere il PLC in base alla qualità e alla quantità delle funzioni che può svolgere, prendendo in considerazione:

- numero e tipo di segnali da elaborare (numero I/O, segnali analogici e/o digitali)
- moduli speciali (per encoder, controllo assi, lettura codici a barre)
- quantità di memoria occorrente
- tempo di ciclo del PLC

Criteri tecnologici

permettono di adattare il controllore all'ambiente di lavoro considerando:

- tipo di sensori e attuatori
- tipo di morsettiere e cablaggi (preferire controllori capaci di gestire morsettiere

- intelligenti e con morsettiere staccabili per non scollegare il cablaggio)
- non volatili in caso di interruzioni di alimentazione)
- caratteristiche ambientali (controllori che devono lavorare con vibrazioni, alte temperature, agenti chimici)

Criteri operativi

consentono di scegliere il controllore che meglio si adatta alle esigenze presenti e future per i seguenti motivi

- possibilità di dialogo con altri controllori o calcolatori(PLC predisposto all'interfacciamento con altri PLC,calcolatori, reti di comunicazione)
- assistenza tecnica (disponibilità di ricambi, documentazione tecnica, corsi di formazione)
- vita operativa (obsolescenza veloce, 4-6 anni,quindi preferire modelli nuovi)
- linguaggi di programmazione(scelta del PLC che, potendo disporre di più linguaggi, consenta una facile programmazione a persone con preparazione tecnica diversa)

Criteri economici

valutano

- costo delle apparecchiature (risparmio iniziale può significare maggiori spese future)
- costo delle interruzioni del funzionamento dell'impianto (considerare disponibilità parti di ricambio, preparazione del personale addetto alla programmazione e alla manutenzione)

Normativa

Le norme tecniche più importanti che riguardano la progettazione l'installazione e la programmazione dei PLC sono:

- **IEC 204-1 → CENELEC EN 60204-1 → CEI 44-5 : “Sicurezza del macchinario. Equipaggiamento elettrico delle macchine. Parte I: regole generali”.**

Questa norma è importante per la corretta installazione del PLC in quanto essi vengono impiegati principalmente a bordo macchina.

- **IEC 1131-1 → CENELEC EN 61131-1 → CEI 65-23: “Controllori programmabili. Parte I: regole generali”.**

Ha la funzione di regolamentare la standardizzazione del PLC.

- **IEC1131-3 → CENELEC EN 61131-3 → CEI 65-40: “PLC programmabili. Parte 3: Linguaggi di programmazione”**

Specifica le regole sintattiche e semantiche dei linguaggi di programmazione usati per programmare i PLC.

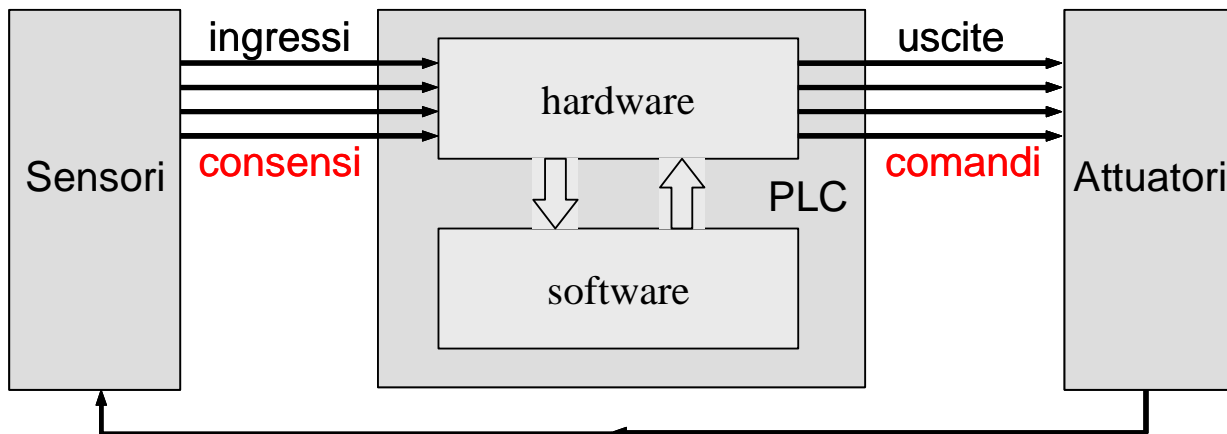
N.B.: Quando una norma IEC viene interamente approvata dal CENELEC, al numero IEC si somma 60.000 e viene aggiunto il suffisso EN; ciò per avere la corrispondenza tra codice europeo e internazionale .

Principio di funzionamento di un PLC

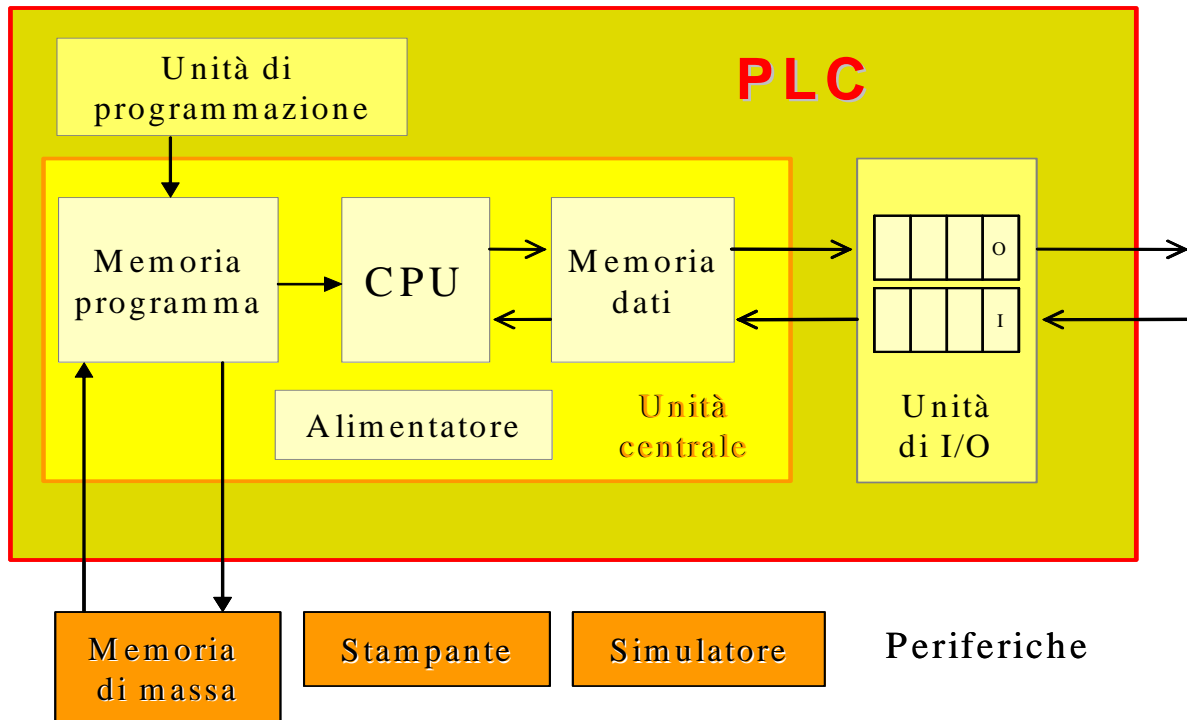
Il PLC a differenza del PC non deve interfacciarsi con l'uomo ma con circuiti elettrici. Il suo microprocessore non si aspetta quindi di ricevere un segnale da un mouse o da una tastiera ma dai sensori, dispositivi elettrici quali un pulsante, un contatto di allarme, un segnale di livello, ecc. Una volta elaborati i segnali di ingresso tramite il programma, il PLC produrrà in uscita un risultato che non deve essere visualizzato su un monitor, ma che deve comandare gli attuatori, motori- cilindri pneumatici- lampade di segnalazione, ecc. Vi sono quindi dei circuiti di ingresso che adattano i segnali elettrici provenienti dal macchinario o dall'impianto alle tensioni e correnti interne al PLC, dei circuiti di uscita che compiono la sequenza contraria.

Per il funzionamento del PLC sono necessarie l'**hardware** (parte materiale, fisica del dispositivo) e il **software** ("parte leggera", programma).

Attraverso i sensori giungono in ingresso i consensi, segnali che vengono elaborati dal PLC secondo il programma stabilito, fornendo in uscita gli opportuni comandi per gli attuatori.



Hardware di un PLC

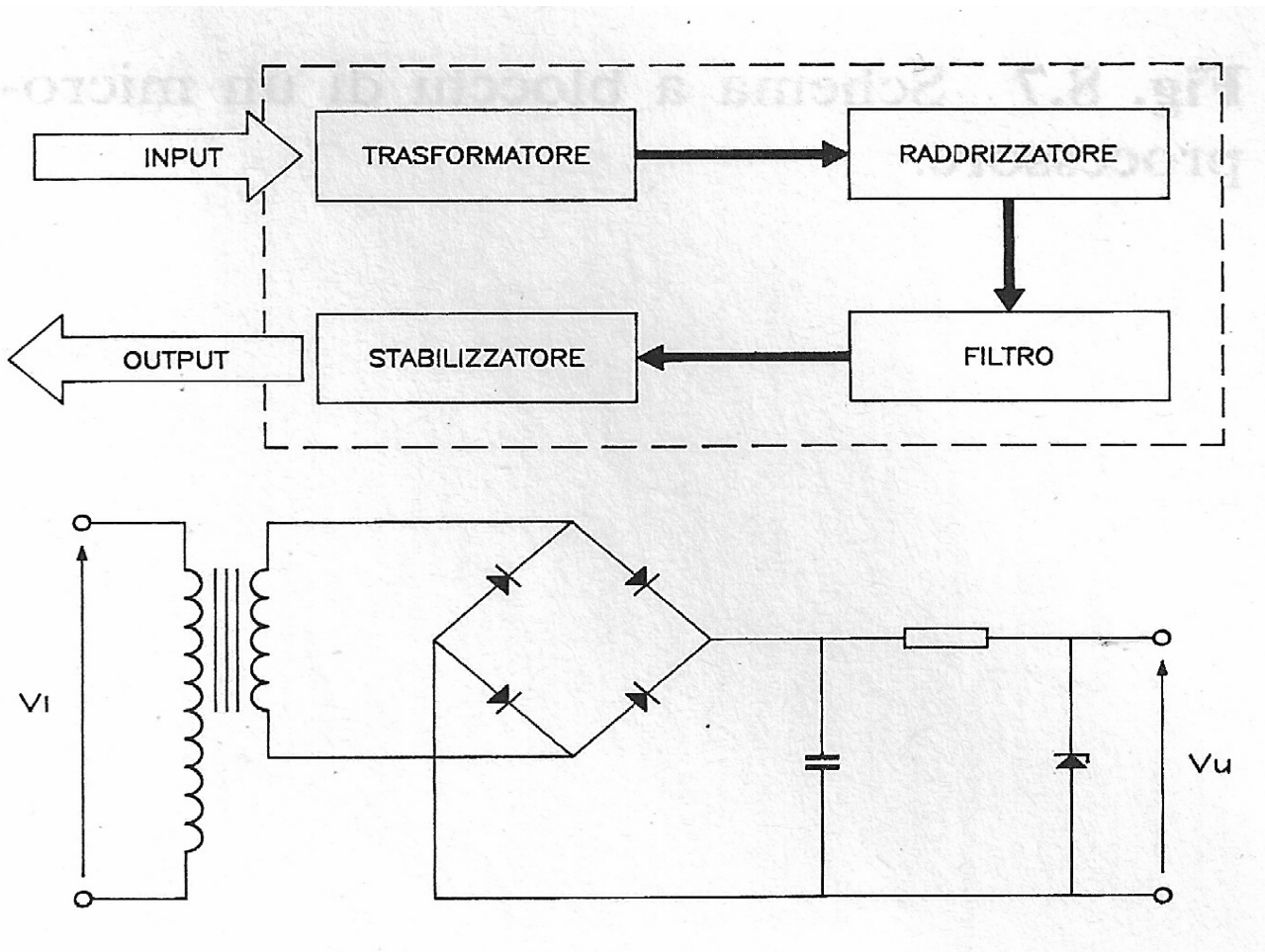


Ogni PLC è costituito dai seguenti componenti:

- alimentatore;
- unità centrale;
- unità di programmazione;
- unità di memoria;
- unità di ingresso e uscita.

Alimentatore

Serve a convertire la tensione di rete 230 V in tensione continua a 5, 12 o 24 V per il funzionamento della CPU.



La tensione alternata di rete viene abbassata tramite un trasformatore e raddrizzata mediante un raddrizzatore a semionda, a doppia semionda o a ponte di Graetz. La tensione raddrizzata viene livellata, cioè viene eliminata la componente alternata residua, con un filtro costituito da un condensatore collegato all'uscita del raddrizzatore, in parallelo al carico oppure con un filtro LC. Infine un diodo Zener consente di stabilizzare la tensione in uscita contro le variazioni della tensione in ingresso o della resistenza del carico.

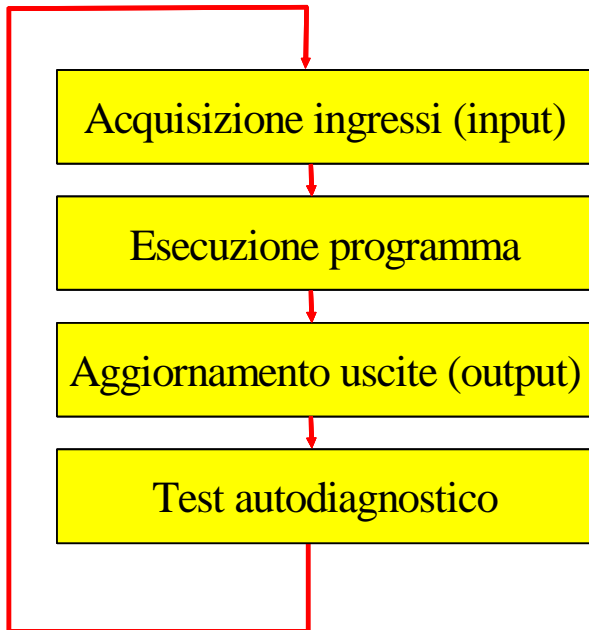
L'alimentatore è provvisto di una batteria tampone che salvaguarda i dati del programma salvati nella memoria RAM in caso di interruzione dell'alimentazione esterna.

Unità centrale

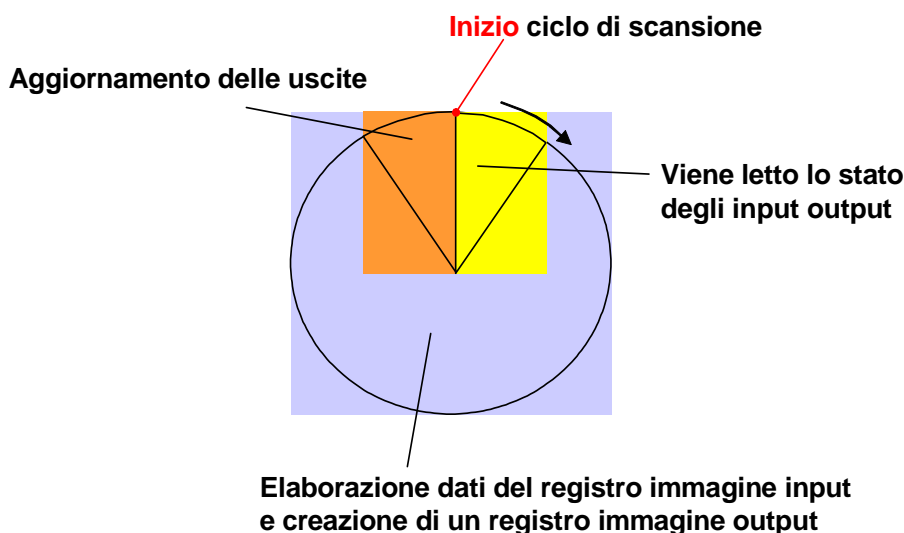
L'unità centrale o CPU (Central Processing Unit) è la parte che organizza e controlla tutte le operazioni eseguite dal PLC.

La CPU impiega un microprocessore per eseguire le istruzioni del programma utente che, in base ai segnali di ingresso, genera i segnali di uscita. In particolare le funzioni della CPU sono:

- Acquisire i segnali di ingresso generandone una immagine negli appositi registri della memoria dati (registri immagine).
- Controllare la sequenza con cui le istruzioni del programma vengono lette, interpretate ed eseguite.
- Interrompere la sequenza normale del programma(interrupt) in caso di salti condizionati e richiami a sottoprogrammi.
- Generare un registro immagine delle uscite aggiornandole al termine di ogni ciclo di scansione.



Tali funzioni vengono svolte dalla CPU secondo un ordine ciclico detto **scansione**.



Se il test di autodiagnosi rivela qualche cattivo funzionamento del PLC l'esecuzione viene fermata e si ha un segnale di allarme.

Dunque la **scansione** è l'esecuzione delle operazioni da parte della CPU secondo un ordine ciclico e in un certo tempo che dipende dalla quantità di memoria e che va da mezzo ms ad alcune decine di ms.

In genere i PLC hanno un tempo di macchina chiamato **watchdog** (cane da guardia) che viene regolato ad un valore definito in configurazione (di 250 ms, valore predefinito, 500 ms massimo) e ha la funzione di fermare il ciclo se la scansione supera tale tempo massimo. Ciò per evitare che si generino dei cicli di programma ripetitivi(loop) dovuti a malfunzionamenti.

Unità di programmazione

E' il dispositivo che permette di inserire nella memoria del PLC il programma che dovrà essere eseguito.

Consente la scrittura, la lettura, le modifiche(editing), l'eventuale correzione(debug) e il controllo del programma.

Può essere predisposta dal costruttore del PLC ed essere costituita da una tastiera alfanumerica e da un dispositivo di visualizzazione (display), oppure può essere un personal computer.

In ogni caso l'unità di programmazione deve essere interfacciata per poter essere collegata al PLC. Nel caso l'u. di p. sia un p c è necessaria l'interfaccia di comunicazione seriale tipo RS232C oppure RS422.

Unità di memoria

La memoria è un componente in grado di immagazzinare e mantenere delle informazioni in modo permanente o temporaneo.

La memoria di un PLC (come per il computer) è costituita da dispositivi a semiconduttore, con tecnologia CMOS a basso consumo, realizzati da una serie di celle di memoria con funzionamento bistabile. Essa può essere *volatile* se non mantiene le informazioni al cessare dell'alimentazione elettrica, *non volatile* se le informazioni rimangono memorizzate.

Ancora, la memoria di un PLC è suddivisa in memoria dati e memoria programma: la memoria dati o memoria di lavoro memorizza i dati relativi allo stato degli ingressi e delle uscite; la memoria programma è la memoria in cui viene memorizzato il programma utente. Le celle di memoria sono raggruppate in parole (word) ciascuna delle quali è formata da un certo numero di bit (binari digit, cioè cifra binaria che può assumere valore 0 oppure 1).

Una word è generalmente formata da 16 bit, cioè 2 byte (1 byte = 8 bit), o da 32 bit, cioè 4 byte(una word può essere formata anche da un numero di bit diverso da 16 o 32).

Con n bit ci sono 2^n combinazioni possibili: $2^1 = 2$; $2^2 = 4$; $2^3 = 8$; $2^4 = 16$; $2^5 = 32$; $2^6 = 64$; $2^7 = 128$; $2^8 = 256$; $2^9 = 512$; $2^{10} = 1024$.

In genere vengono utilizzati multipli del byte o della word perché è necessario un grande numero di bit per indicare la capacità di memoria(cioè il numero di byte che il PLC consente di memorizzare). Si ottiene così il kilobyte e il kiloword (1 kbyte = 1024byte; 1kword = 1024word), il Megabyte (1Mb = 1024 kbyte = 1048576byte), il Gigabyte (1 Gb = 1024 Mb = 1073742824byte). Dove il numero 1024 è uguale a 2^{10} e nella numerazione binaria rappresenta la potenza del 2 più vicina a 1000, 1 kilo per l'appunto, mentre il numero 1048576 è uguale a 2^{20} e rappresenta la potenza di 2 più vicina a 1000000, 1 Mega, e infine il numero 1073 742 824 è uguale a 2^{30} che rappresenta la potenza di 2 più vicina a 1000000000, 1 Giga.

Ci sono vari tipi di memoria: floppy disk, hard disk, memorie statiche(senza parti in movimento).

Nel PLC, come il computer,si usano vari tipi di memorie statiche:

- RAM,
- ROM,

- PROM,
- EPROM,
- EEPROM.

RAM: Le memorie RAM(Random Acces Memory, memoria ad accesso casuale) sono memorie a lettura / scrittura di tipo volatile(nel PLC c'è una batteria tampone per l'alimentazione di emergenza), utilizzate per la memoria dati, dove le informazioni cambiano in continuazione (dati degli I/O), oppure per la memoria programma, quando non è ancora il programma definitivo.

ROM: Le memorie ROM (Read Only Memory, memoria a sola lettura) sono non volatili(a contenuto non modificabile che viene conservato anche senza alimentazione) e vengono utilizzate per la memoria programma in veste definitiva, quando non verrà più modificato. Contengono il software di sistema, o sistema operativo, cioè il programma scritto dal costruttore (firmware) che consente l'uso e la programmazione del PLC.

PROM: Le memorie PROM (Programmable ROM) sono memorie di sola lettura programmabili una sola volta. A differenza delle ROM, che vengono programmate dal costruttore del PLC, le PROM sono programmate dall'utente. La programmazione di queste memorie avviene mediante eliminazione, con apposito dispositivo, di una connessione fusibile che una volta bruciata non può più essere ripristinata.

EPROM: Le EPROM (Erasable Programmable Read only Memory, memoria a sola lettura programmabile e cancellabile) sono memorie non volatili il cui contenuto può essere cancellato in 10-30 minuti con una macchina a raggi uv chiamata EPROM programmer, attraverso una finestra trasparente in quarzo sul dorso, e programmate di nuovo. Per poter cancellare questa memoria bisogna estrarre il componente dalla sua scheda. Una volta programmata la finestra deve essere coperta con un nastro adesivo non trasparente per evitare che la luce naturale o artificiale possa in parte cancellarla.

EEPROM: Le memorie EEPROM o E²PROM (Electrically Erasable PROM) sono simili alle EPROM ma possono essere cancellate applicando una tensione opportuna senza doverle rimuovere dal PLC.

Inoltre esistono memorie quali NOVRAM, RAM non volatile, e FLASH, una via di mezzo tra le EPROM e le EEPROM.

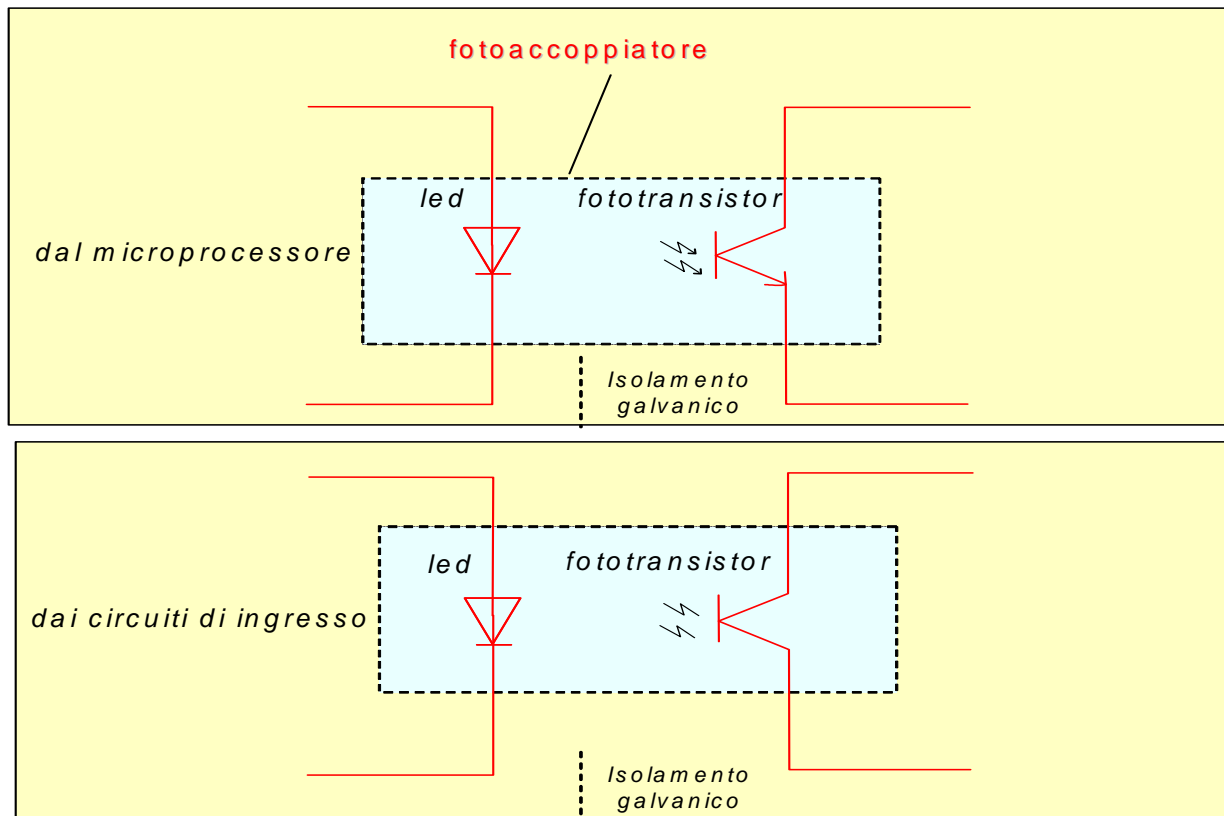
Unità ingresso / uscita (Unità I/O)

Le unità di ingresso e uscita collegano il PLC agli organi di rilevazione e agli attuatori; trasformano i segnali elettrici in ingresso, provenienti da sensori, finecorsa, pulsanti, ecc. e li adattano al controllore e viceversa convertono i segnali in uscita dal controllore e li adattano ai vari azionamenti ed attuatori (contattori, motori, ecc.).

Tali unità si rendono necessarie perché l'Unità Centrale del PLC funziona con pochi volt e pochi mA mentre i circuiti esterni, in modo particolare i circuiti di potenza, funzionano con tensioni e correnti molto più elevate; allora le unità di ingresso e uscita permettono di adattare i valori di tensione e corrente tra PLC e circuiti esterni, consentendo inoltre l'isolamento galvanico tra i circuiti interni e quelli esterni. Infatti in tali unità si inseriscono dei dispositivi fotoaccoppiatori detti optoisolatori che, consentendo la trasmissione dei segnali elettrici tra due circuiti separati elettricamente, proteggono il PLC da sbalzi di tensione o cortocircuiti provenienti dall'esterno oppure da interferenze che il controllore potrebbe interpretare come veri e propri segnali e utilizzarli come tali con conseguenze imprevedibili.

I dispositivi fotoaccoppiatori che realizzano l'isolamento galvanico sono in grado di garantire un isolamento tra due circuiti da 1500 a 4000V.

Un fotoaccoppiatore può essere composto da un diodo emettitore di luce (LED) posto affacciato ad un fototransistor ricevitore, entrambi racchiusi in un unico contenitore; è possibile quindi permettere il passaggio di corrente nel fototransistor solo quando il diodo è alimentato e cioè mentre emette luce: si ottiene così il trasferimento dell'informazione logica tra due circuiti che di fatto sono elettricamente isolati fra di loro (un po' come accade per i trasformatori con gli avvolgimenti separati, dove l'accoppiamento tra il circuito elettrico del primario e il circuito elettrico del secondario avviene tramite il circuito magnetico).



Esempio di isolamento galvanico mediante fotoaccoppiatore

I segnali in ingresso al PLC possono essere digitali ON/OFF oppure analogici.

A seconda del tipo di segnali che devono scambiare con il controllore le unità di ingresso e uscita impiegano schede I/O che possono essere schede ON/OFF e schede analogiche. Schede ON/OFF: in queste schede gli ingressi e le uscite possono assumere solo due stati: ON = 1 e OFF = 0.

Schede analogiche: in queste schede gli ingressi e le uscite possono avere un qualsiasi valore tra un limite inferiore ed uno superiore.

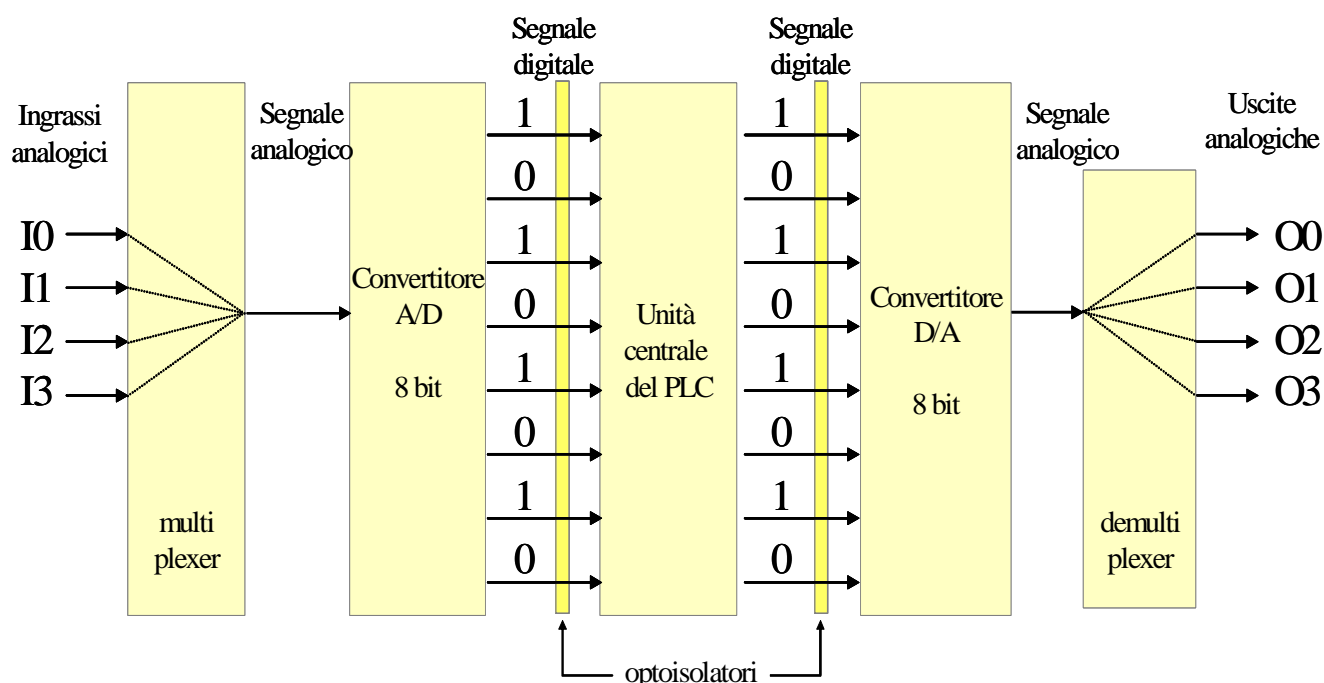
Entrambe possono essere schede di ingresso (che ricevono i segnali provenienti dai trasduttori), di uscita (che mandano i segnali agli attuatori) o miste (che presentano sia ingressi che uscite). Inoltre possono essere schede funzionanti in corrente continua (5-12-24 V), in corrente alternata (110-220 V) oppure schede universali (in cui i segnali possono essere indifferentemente in continua o in alternata). Le schede di ingresso ON/OFF si collegano di solito a finecorsa, pulsanti, contatti di relè, ecc. Contengono i dispositivi

optoisolatori con cui l'unità centrale viene isolata dall'impianto. Su ogni scheda possono esserci 4-8-16-32-64 ingressi. Le schede di uscita ON/OFF si collegano di solito a bobine di relè, contattori, elettrovalvole, lampade di segnalazione, ecc. Le uscite possono essere a relè oppure, per avere maggiore velocità di commutazione, a transistor o triac; contengono gli optoisolatori con cui l'unità centrale viene isolata dall'impianto. Su ogni scheda possono esserci 4-8-12-16-32-64 uscite. Le schede di ingresso analogiche si collegano di solito a trasduttori di temperatura, pressione, velocità, portata, ecc. Gli ingressi analogici variano in modo continuo tra due valori: tra 0 e 10 V oppure tra 0 e 20 mA e vengono trasformati in una grandezza digitale (cioè in una successione di bit) tramite un convertitore A/D (analogico/digitale).

La precisione della trasformazione dipende dal numero di bit : con 4 bit si hanno $2^4 = 16$ valori possibili tra i due valori limite con cui si può rappresentare una grandezza (16 gradini come nel grafico sotto), con 8 bit si hanno $2^8 = 256$ valori (grandezza rappresentabile con un grafico a 256 gradini, che si avvicina un po' di più a quella reale), ecc. Mediante un multiplexer vengono convertiti a turni più ingressi utilizzando un solo convertitore. Le schede di uscita analogiche si collegano di solito a dispositivi di potenza come variatori di velocità, valvole, ecc.

Queste schede fanno l'esatto contrario delle schede di ingresso.

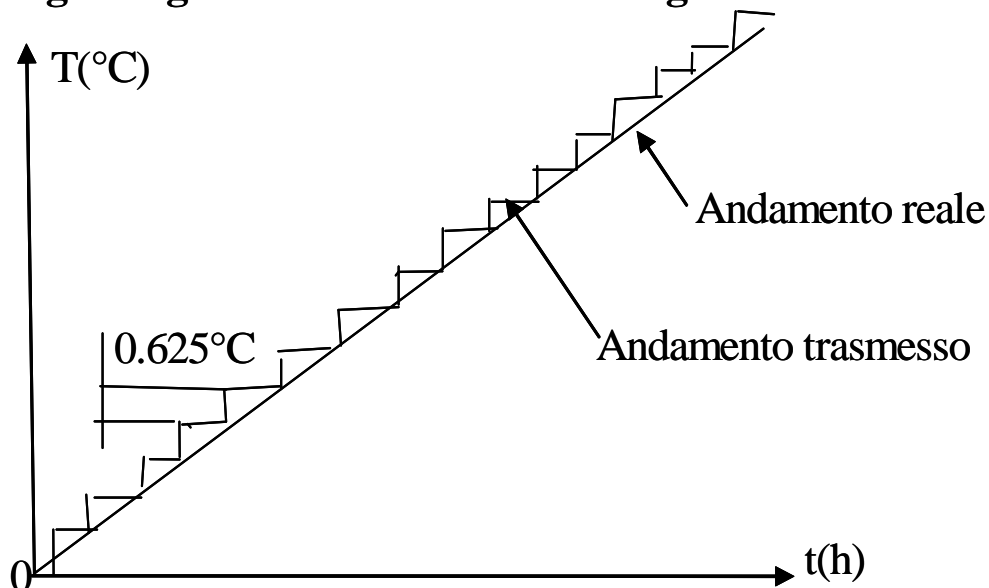
La grandezza digitale viene trasformata in una grandezza analogica tramite un convertitore D/A (digitale/analogico). La precisione della trasformazione dipende dal numero di bit. Mediante un demultiplexer vengono pilotate a turno più uscite analogiche utilizzando un solo convertitore.



La precisione della conversione dipende dal numero di bit.

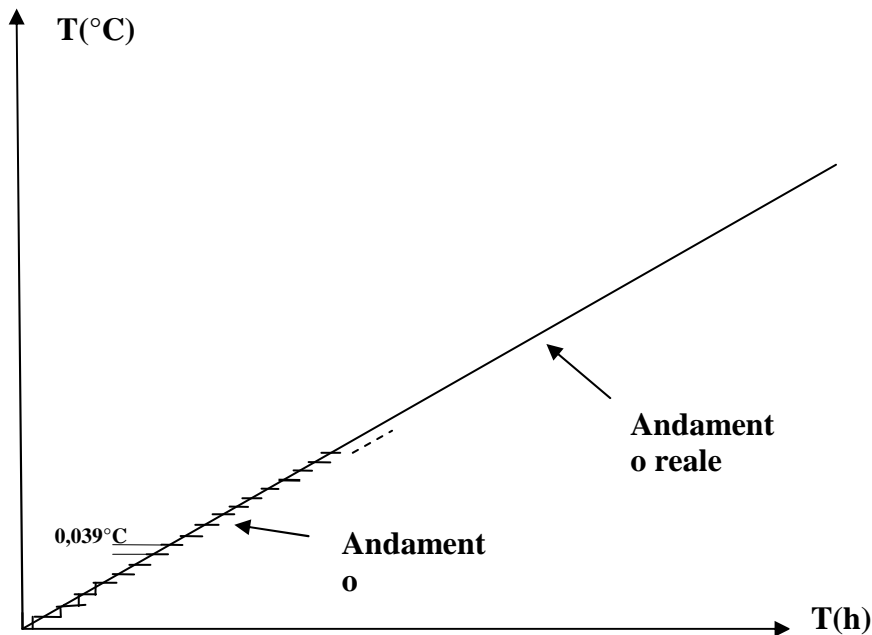
Si supponga di dover tradurre una temperatura che varia linearmente con il tempo tra 0 e 10°C. A prescindere dalla trasformazione in corrente o in tensione, con un convertitore A/D a quattro bit si hanno $2^4 = 16$ combinazioni (sedici possibili segnali digitali e quindi si può ottenere la temperatura con una precisione massima di $(10 - 0)/16 = 0.625^\circ\text{C}$. Questo significa che l'immagine che il PLC può avere della variabile temperatura è quella a gradini della figura e non quella reale.

Immagine digitale di una variabile analogica



Con un convertitore A/D a otto bit si hanno $2^8 = 256$ modi per rappresentare la stessa variazione di temperatura e si può raggiungere una precisione di $(10-0)/256 = 0.039^{\circ}\text{C}$.

Immagine digitale di una variabile analogic



Con 16 bit si ottiene un gradino di $(10-0)/65536 = 0.00015^{\circ}\text{C}$ e quindi una immagine della temperatura che si avvicina molto a quella reale; $2^{16} = 65536$ combinazioni (65536 possibili segnali digitali).

Periferiche

Oltre ai componenti visti, il PLC, per poter dialogare con l'esterno(per la programmazione, i controlli diagnostici, la stampa su carta dei programmi, ecc.), necessita di altre apparecchiature, dette periferiche, come *simulatori*, *memorie di massa*, *stampanti*, *programmatori di EPROM*.

Il simulatore è un dispositivo che, forzando gli stati logici di ingressi e uscite, simula ciò che avviene nel sistema da controllare e rende più veloce il collaudo e la correzione del programma. A questo scopo si impiegano dei commutatori (microswitch) a due posizioni che consentono di commutare l'ingresso desiderato; inoltre si può vedere se un'uscita si eccita impiegando dei led che ne indicano lo stato logico.

Le memorie di massa (floppy disk e hard disk) permettono di conservare i programmi già scritti, anche se non sono sul PLC, evitando così di doverli riscrivere quando servono.

Le stampanti servono per avere su carta i programmi residenti in memoria.

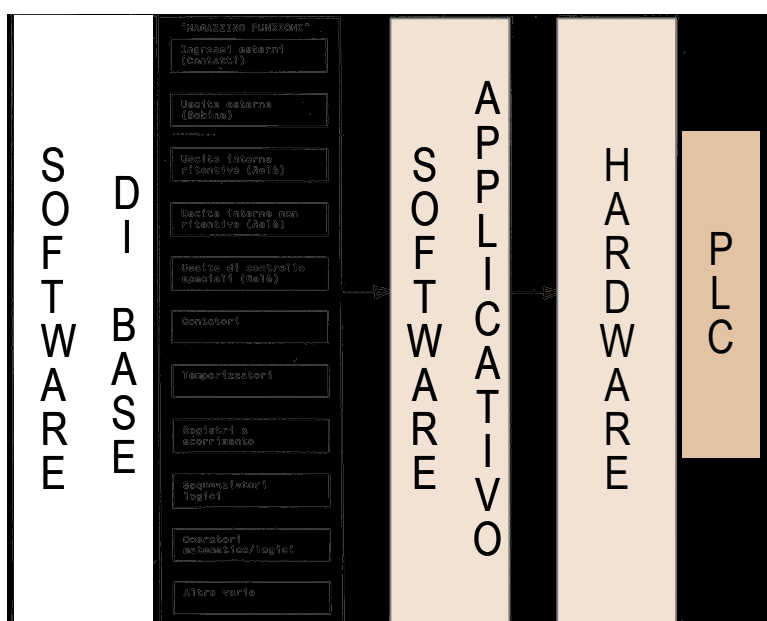
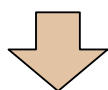
Software di un PLC

I PLC hanno la possibilità di simulare, all'interno dei propri circuiti elettronici, un certo numero di funzioni che riproducono quelle della logica elettromeccanica.

Allora il PLC si può considerare un “magazzino di funzioni” dal quale l'utente preleva istante per istante quelle che gli servono utilizzando come mezzo per prelevarle il software di base.

Il programmatore può impiegare queste funzioni per realizzare il proprio programma applicativo che, associato all'hardware, è in grado di utilizzare le informazioni che gli arrivano dall'esterno per controllare gli attuatori.

programmatore



Linguaggi di programmazione del PLC

Programmare un PLC, per fargli svolgere una certa sequenza di operazioni, vuol dire trasferire nella sua memoria un programma, cioè una sequenza di istruzioni prelevate dall'insieme di istruzioni base.

All'interno del PLC le informazioni devono essere codificate in *linguaggio macchina*, che è un linguaggio binario (fatto di zeri e di uno) che può essere sconosciuto a chi usa i PLC. Per cui sono stati concepiti dei linguaggi di programmazione specifici per applicazioni industriali. Un *linguaggio di programmazione* è un insieme di regole che stabiliscono come scrivere le istruzioni necessarie a comunicare al PLC cosa deve fare.

I linguaggi di programmazione si possono classificare in linguaggi **letterali** e **grafici**.

Sono linguaggi letterali

- il linguaggio a **Lista Istruzioni - IL(Instruction List)**

Detto anche linguaggio **booleano** o **AWL**, un linguaggio macchina, cioè usa direttamente le istruzioni del microprocessore. E' poco pratico e richiede molto tempo per la ricerca di errori nel programma e guasti nell'impianto controllato. E' usato per lo più dagli elettronici e dagli informatici.

- il linguaggio a **Testo Strutturato - ST**

E' un linguaggio ad alto livello come lo sono il **Basic**, il **Pascal**, il **Fortran**, il **C++**, ecc. E' usato per lo più dagli informatici.

Sono linguaggi grafici

- lo **Schema a Blocchi di Funzioni - FBD (Function Block Diagram)**

E' il linguaggio a **simboli logici** usato dagli elettronici per disegnare gli schemi dell'elettronica digitale. Viene impiegato nei sistemi di controllo di grossi impianti come centrali termoelettriche, impianti chimici, ecc.

- lo **Schema Funzionale in Sequenza – SFC(Sequential Function Chart)**

E' un linguaggio sviluppato in Francia con il nome di **Linguaggio Grafcet**.

Rappresenta il funzionamento per passi di un processo automatico in modo simile ad un Flow-Chart, ma dove ogni blocco rappresenta uno stato del processo di lavorazione della macchina. Può essere usato da tutti se conoscono bene ogni passo dell'applicazione.

- lo **Schema a contatti – LD(Ladder Diagram)**

Detto anche linguaggio **Ladder** o **Kop**, lo schema a contatti è così chiamato perché sostituisce la logica a contatti a relè. Infatti deriva dalla tecnologia elettromeccanica e si presenta con una simbologia che riproduce lo schema funzionale: due linee verticali, anziché orizzontali, indicano l'alimentazione, all'interno di queste due linee si scrive il programma.

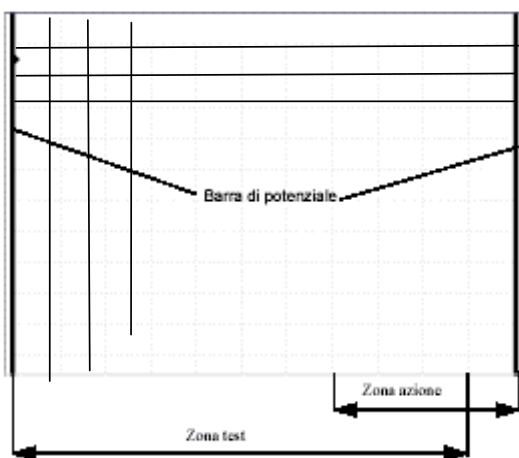
Ogni pagina di programma(o rete oppure *rung*, letteralmente *traversa*, *piolo*) è formata da un certo numero di linee orizzontali (simili ai gradini di una scala, da cui il nome ladder = scala), lungo le quali si inseriscono gli elementi grafici che sostituiscono i segni grafici dello schema funzionale, e da un certo numero di colonne.

A sinistra di tale schema vi è la “*zona test*” in cui in genere si scrivono gli ingressi(variabili-contatti o bit-di ingresso, variabili-contatti o bit- interne, blocchi funzione); a destra la “*zona azioni*” che comprende le uscite (bobine di uscita,bobine interne, blocchi operazione e blocchi comparazione).

La lettura di ogni pagina di programma da parte della CPU segue un percorso dall'alto in basso e da sinistra a destra.

Dunque il Ladder è il linguaggio più naturale da usare per un tecnico che ha una preparazione elettromeccanica(elettrotecnici).

Struttura di una pagina di programma in linguaggio Ladder:



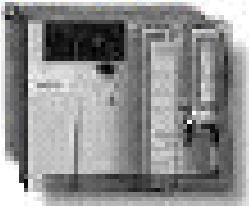
CONTROLLORE MODICON TSX TELEMECANIQUE

L'hardware del TSX MICRO

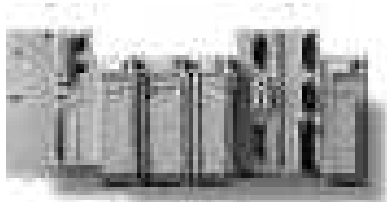
I controllori MODICON TSX TELEMECANIQUE, sono commerciabili in tre modelli:



TSX 07(TSX NANO) di gamma bassa



TSX 37(TSX MICRO) di gamma media



TSX 57(TSX PREMIUM) di gamma alta

Il controllore **TSX MICRO** è disponibile con cinque modelli di processore: **TSX 37.05**; **TSX 37.08**; **TSX 37.10**; **TSX 37.21**; **TSX 37.22**.

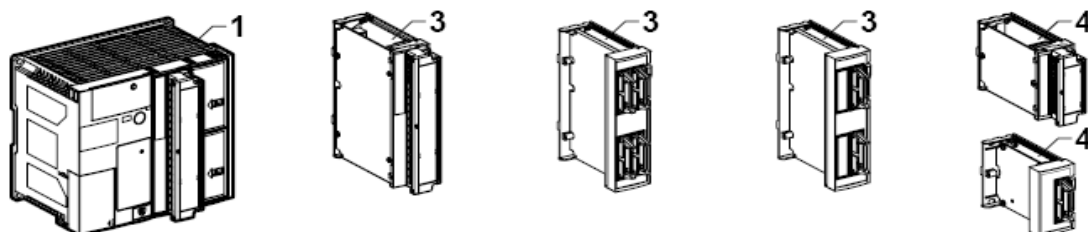
Il **TSX 37.05** può essere di tipo compatto o modulare. E' costituito da un modulo base con 28 I/O digitali (16 I + 12 O) installato nel primo alloggiamento (*slot*) e un alloggiamento libero che può essere completato con un modulo di formato standard o con due moduli di mezzo formato. E' disponibile in una sola configurazione con alimentazione in corrente alternata e non gestisce I/O remoti.

Il **TSX 37.08**, di tipo compatto o modulare, ha di base due moduli con 28 I/O digitali installati nei primi due alloggiamenti e un alloggiamento libero che può essere completato con un modulo di formato standard o due moduli di mezzo formato. E' disponibile in una sola configurazione in corrente alternata e non gestisce I/O remoti.

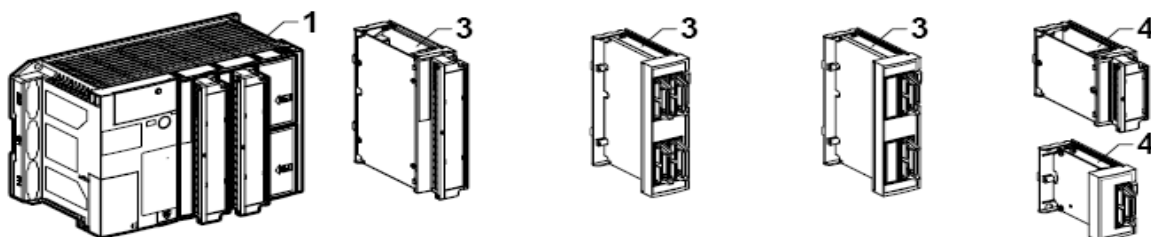
Il **TSX 37.10** può essere di tipo compatto o modulare ed ha un modulo a 28 o 64 I/O di base nel primo alloggiamento e un alloggiamento libero che può essere occupato con moduli di formato standard o di mezzo formato. Ogni base (rack) può essere ampliata tramite un mini-rack di estensione. In versione modulare è disponibile con **sei diverse configurazioni** a seconda del tipo di alimentazione (continua o alternata) e del tipo di modulo I/O che occupa il primo slot del rack di base. Inoltre esso è estendibile, con unità di espansione, fino a 268 I/O. Infatti ogni PLC TSX 37.10, oltre al mini-rack di estensione, può gestire I/O remoti tramite moduli di estensione utilizzando una connessione con PLC Nano oppure un bus AS-i con sensori e attuatori collegati.

I **TSX 37.21/22** sono di tipo modulare con un modulo base estendibile fino a 332 I/O e disponibili entrambi con due configurazioni: corrente continua o alternata senza I/O di base.

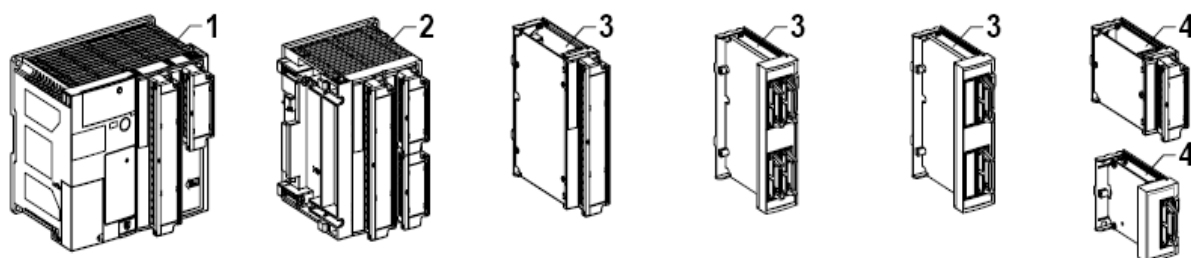
TSX 37 05



TSX 37 08



TSX 37 10



1. base
2. mini-rack di estensione
3. moduli formato standard
4. moduli mezzo formato

configurazioni disponibili

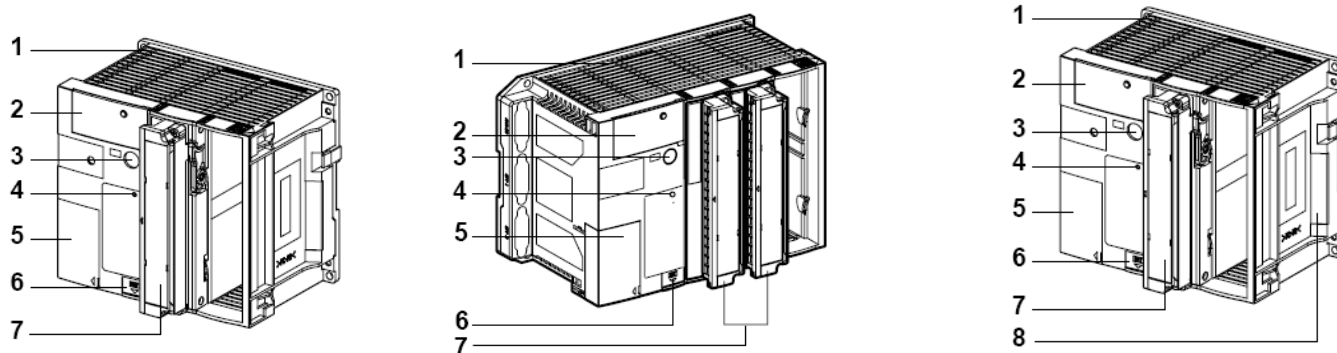
Type of power supply ~ 100...240V = 24V		Integrated I/O modules						PLC references TSX
		TSX module references	Types of input		Types of output		Relay	
			= 24V	~ 115V	Trans. = 24V 0.1A	0.5A		
●		DMZ 28 DR	● (1)				●	3705 028DR1
●		DMZ 28 DR	● (1)				●	3708 056DR1
●		DMZ 28 AR		●			●	3710 028AR1
●		DMZ 28 DR	● (1)				●	3710 028DR1
	●	DMZ 28 DR	● (1)				●	3710 128DR1
	●	DMZ 28 DT	● (2)			●		3710 128DT1
	●	DMZ 28 DTK(3)	● (2)			●		3710 128DTK1
	●	DMZ 64 DTK(3)	● (2)			●		3710 164DTK1

(1) Positive or negative logic inputs
(3) HE10 connector module

(2) Positive logic inputs

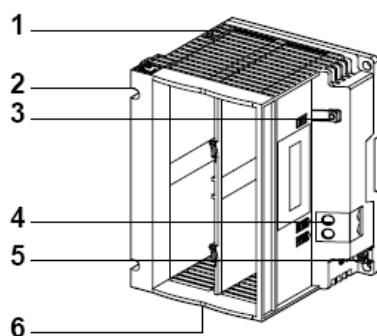
Descrizione fisica del TSX 37 05 08 10

Base



1. Rack a due slot, che integra l'alimentazione, il processore e la relativa memoria.
2. Pannello di visualizzazione centralizzato.
3. Porta terminale TER.
4. Pulsante di RESET.
5. Pannellino d'accesso ai morsetti d'alimentazione.
6. Sportellino d'accesso alla batteria opzionale(TSX PLP 01) e al commutatore di protezione in scrittura del sistema operativo.
7. Modulo a 28 I/O, posizionato di base nel primo slot(37 05), Moduli a 28 I/O, posizionati di base nei primi due slot(37 08), Modulo a 28 o 64 I/O, posizionato di base nel primo slot(37 10).
8. Copri-connettore di raccordo del mini-rack di estensione.

Mini-rack d'estensione



1. Rack di estensione a due slot.
2. Vite di fissaggio dell'estensione alla base.
3. Spia di presenza tensione 24 Vcc.
4. Morsetti di alimentazione protetti da mascherina.
5. Morsetto di massa.
6. Connettore di collegamento al PLC di base.

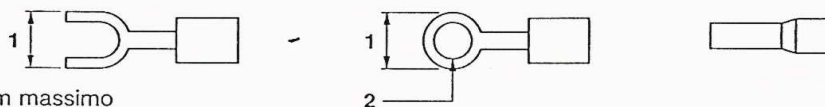
Mezzi di collegamento

1-1 Collegamenti su moduli con morsettiere a vite

Ogni morsetto può ricevere fili nudi o dotati di terminali o capicorda aperti o chiusi.

La capacità di ogni morsetto è:

- minima: 1 filo da 0,28 mm² senza terminale,
- massima: 2 fili da 1 mm² con terminale oppure
1 filo da 1,5 mm² senza terminale oppure
1 capocorda aperto o chiuso per fili da 1 mm².



1 5,5 mm massimo

2 Ø 3,2 mm minimo

Le viti-serrafili sono dotate di un'impronta adatta ai cacciaviti:

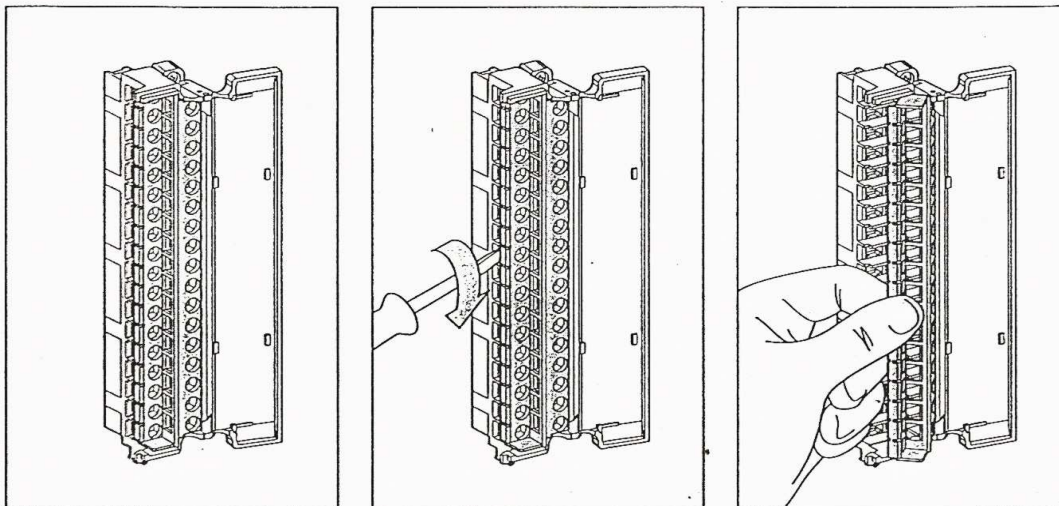
- cruciformi Pozidriv N°1,
- piatti, di diametro Ø 5 mm

Le morsettiere di collegamento a vite sono munite di una mascherina smontabile che garantisce:

- la tenuta delle viti,
- la protezione delle persone.

Nel caso di utilizzo di capicorda chiusi, la mascherina dovrà essere smontata per poter estrarre la vite per il montaggio del capocorda. La mascherina non verrà rimontata per consentire il passaggio dei fili.

Smontaggio della mascherina



1 Apertura sportellino

2 Chiusura mascherina

3 Smontaggio mascherina

1-2 Collegamenti su moduli con connettori HE10

Trefolo precablato da 20 fili, calibro 22 (0,34 mm²),

Consentire il collegamento facile e diretto filo a filo degli ingressi/uscite dei moduli con connettori HE 10 a sensori, preazionatori o morsetti.

Questo trefolo precablato è costituito:

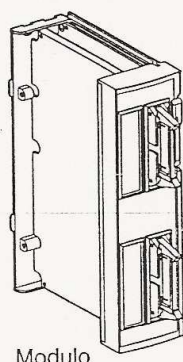
- a un'estremità, da un connettore HE 10 sovrastampato dal quale escono 20 fili di sezione 0,34 mm² protetti da guaina
- all'altra estremità, da fili liberi differenziati tra loro da un codice colore in base alla norma DIN 47100.

Nota: Un trefolo in nylon integrato al cavo permette di scoprire facilmente la guaina.

L'offerta prevede due prodotti con riferimento:

TSX CDP 301: lunghezza 3 metri.

TSX CDP 501: lunghezza 5 metri.



Corrispondenza tra il colore dei fili e il numero del morsetto del connettore HE10

TSX CDP 001		Alto	HE10	
bianco	marrone		1	2
verde	giallo		3	4
grigio	rosa		5	6
blu	rosso		7	8
nero	viola		9	10
grigio-rosa	rosso-blu		11	12
bianco-verde	marrone-verde		13	14
bianco-giallo	giallo-marrone		15	16
bianco-grigio	grigio-marrone		17	18
bianco-rosa	rosa-marrone		19	20
		Basso		

Trefolo di collegamento protetto da guaina calibro 28 (0,08 mm²)

È destinato a consentire il collegamento degli ingressi/uscite dei moduli con connettori HE 10 verso interfacce di collegamento e adattamento a cablaggio rapido TELEFAST 2. Questo cavo è costituito da 2 connettori HE 10 e da un altro cavo piatto trefolato e protetto da guaina con fili di sezione 0,08 mm².

Tenuto conto della sezione ridotta dei fili, si consiglia di utilizzarlo solo su ingressi o uscite a debole corrente (≤ 100 mA per ingresso o uscita).

L'offerta prevede tre prodotti con i seguenti riferimenti:

TSX CDP 102: lunghezza 1 metro

TSX CDP 202: lunghezza 2 metri

TSX CDP 302: lunghezza 3 metri

Cavo di collegamento calibro 22 (0,34 mm²)

Consente il collegamento degli ingressi/uscite dei moduli con connettori HE 10 verso interfacce di collegamento e adattamento a cablaggio rapido TELEFAST 2.

Questo cavo è costituito da 2 connettori HE 10 sovrastampati e da un cavo con fili di sezione 0,34 mm² per il passaggio di correnti più elevate (≤ 500 mA).

L'offerta prevede cinque prodotti con i seguenti riferimenti:

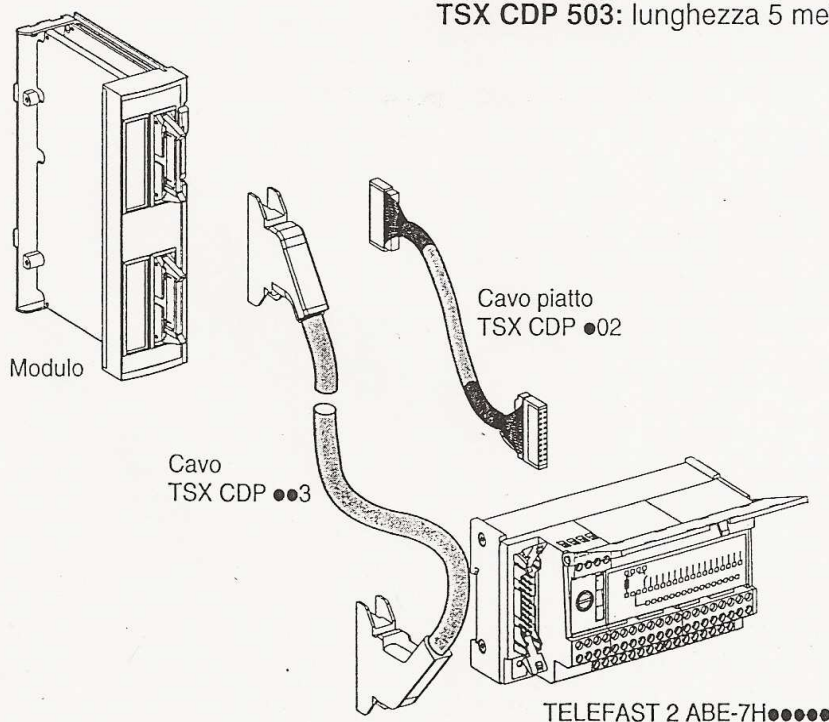
TSX CDP 053: lunghezza 0,50 metro.

TSX CDP 103: lunghezza 1 metro.

TSX CDP 203: lunghezza 2 metri.

TSX CDP 303: lunghezza 3 metri.

TSX CDP 503: lunghezza 5 metri.



Configurazione dei moduli I/O On/Off del TSX 37-10

Il TSX MICRO modulare può avere diverse configurazioni dei moduli I/O per ogni modello di processore. In particolare il TSX 37.10 può avere un numero massimo di 128 I/O utilizzando il solo rack di base, 192 I/O utilizzando rack base + rack estensione, 268 I/O utilizzando rack base + rack estensione + rack a distanza. Ovviamente sotto il numero massimo di I/O, in tutti e tre i casi si possono realizzare più combinazioni.

Per le estensioni a distanza si impiega il controllore TSX 07(TSX NANO). Si può creare così una minirete collegando in parallelo alla scheda di comunicazione STZ 10 quattro TSX 07($4 \times 24 = 96$ I/O).

Analogo discorso vale per i TSX 37.21/22 che ha un rack base con tre slot per i moduli I/O e può arrivare ad un numero di 332 I/O utilizzando rack base + rack estensione + rack a distanza.

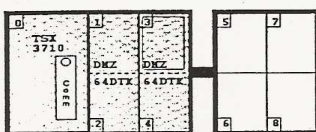
La posizione del processore e dei moduli di I/O di un controllore è identificata da un numero progressivo a partire da sinistra: perciò la CPU occupa la posizione 0, la posizione 1- 2 è occupata da un modulo di formato standard, ecc.

Per ogni controllore è definita la posizione che può occupare ogni tipo di modulo (I/O, conteggio, analogico, comunicazione) nel rack. Inoltre è stabilito anche il numero di moduli, per ogni tipo, consentito per ognuno dei controllori.

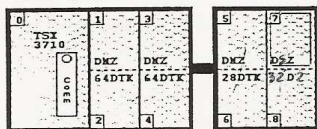
TSX Micro

Configurazione I/U On/Off per TSX 37.10

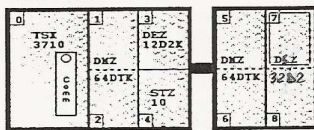
128 I/U



192 I/U

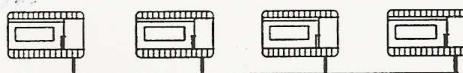


268 I/U



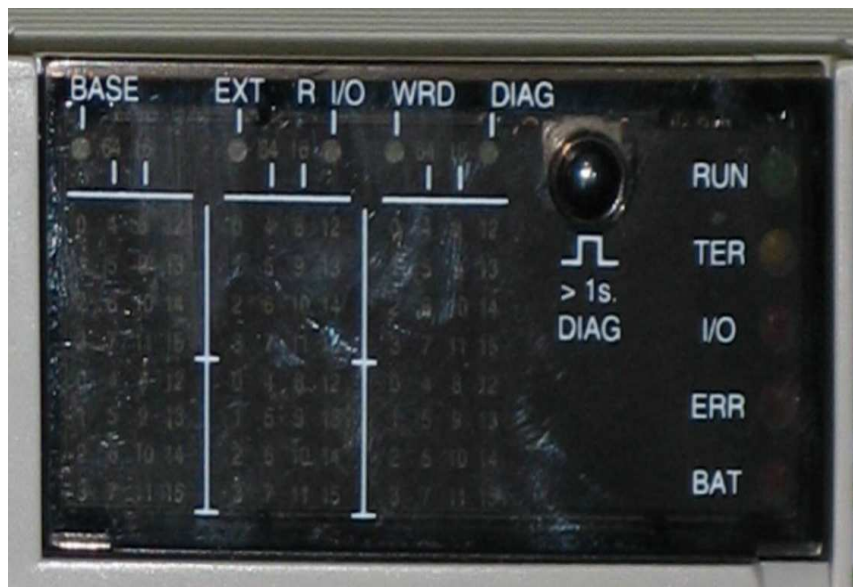
I/U On/Off:

N. max I/U (rack + distanza)	268
N. di moduli a 32 vie	4
N. di moduli a 64 vie (H.D.)	2
N. I/U nel rack di base	128
N. I/U base + estensioni	192
N. I/U a distanza	96
N. di moduli per colleg. I/U a distanza	1
N. di moduli colleg. bus AS-i	1
N. totale di alloggiamenti per moduli	7



Pannello di visualizzazione

- Indica lo stato del PLC e degli I/O
- Permette di accedere alla diagnostica dei canali e dei moduli



Visualizzazione dello stato del PLC

Display block

Display of PLC status

LED	State	PLC
RUN (Green)	On	PLC running
	Flashing	PLC in STOP mode
	Off	No valid application in the PLC or PLC faulty
TER (Yellow)	On	Exchange of data via the terminal port
	Off	No exchange via the terminal port
I/O (Red)	On	I/O supply fault, tripping of a channel, module missing or switched off or not conforming to the configuration
	Off	Operation OK
ERR (Red)	On	CPU fault
	Flashing	No valid application in the PLC or "blocking fault" on the application program
	Off	Operation OK
BAT (1) (Red)	On	Battery faulty or missing
	Off	Battery OK

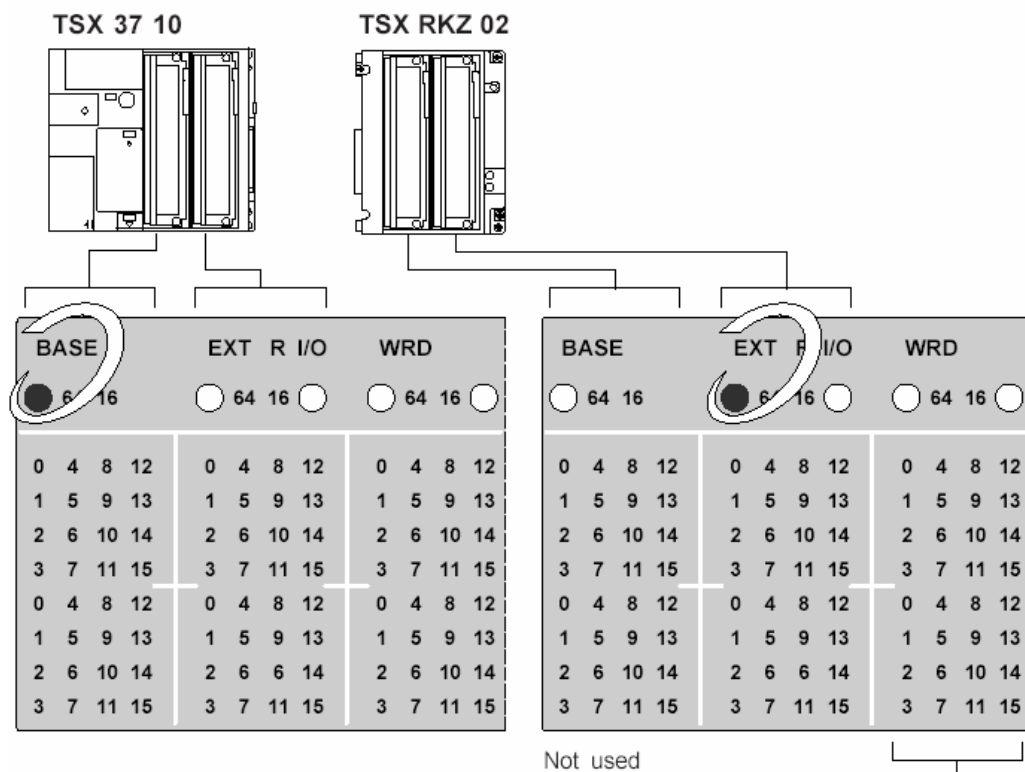
(1) The battery must be changed every two years. Update the label located in the cover giving access to the power terminals.

Visualizzazione dello stato degli I/O

Il pannello di visualizzazione mostra simultaneamente lo stato degli I/O dei 2 moduli:

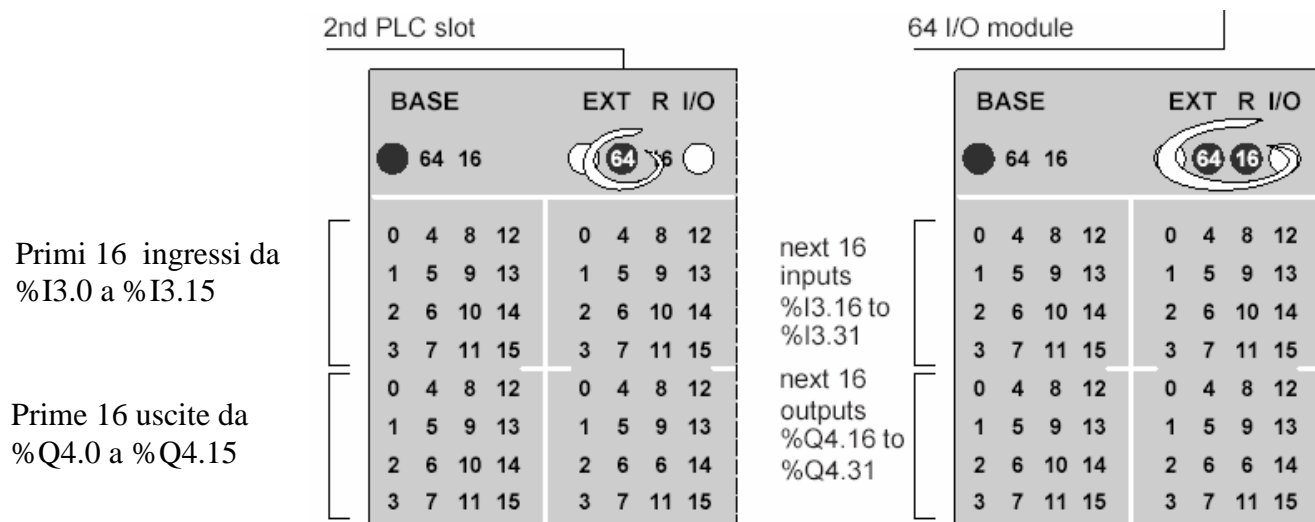
- 2 moduli della base(spia BASE accesa),
- o 2 moduli del mini rack di estensione (spia EXT accesa). Una leggera pressione del pulsante di selezione permette di selezionare il rack visualizzato (BASE o EXT).

- TSX 37 10 PLC with mini extension rack TSX RKZ 02



Visualizzazione dei moduli a 64 canali

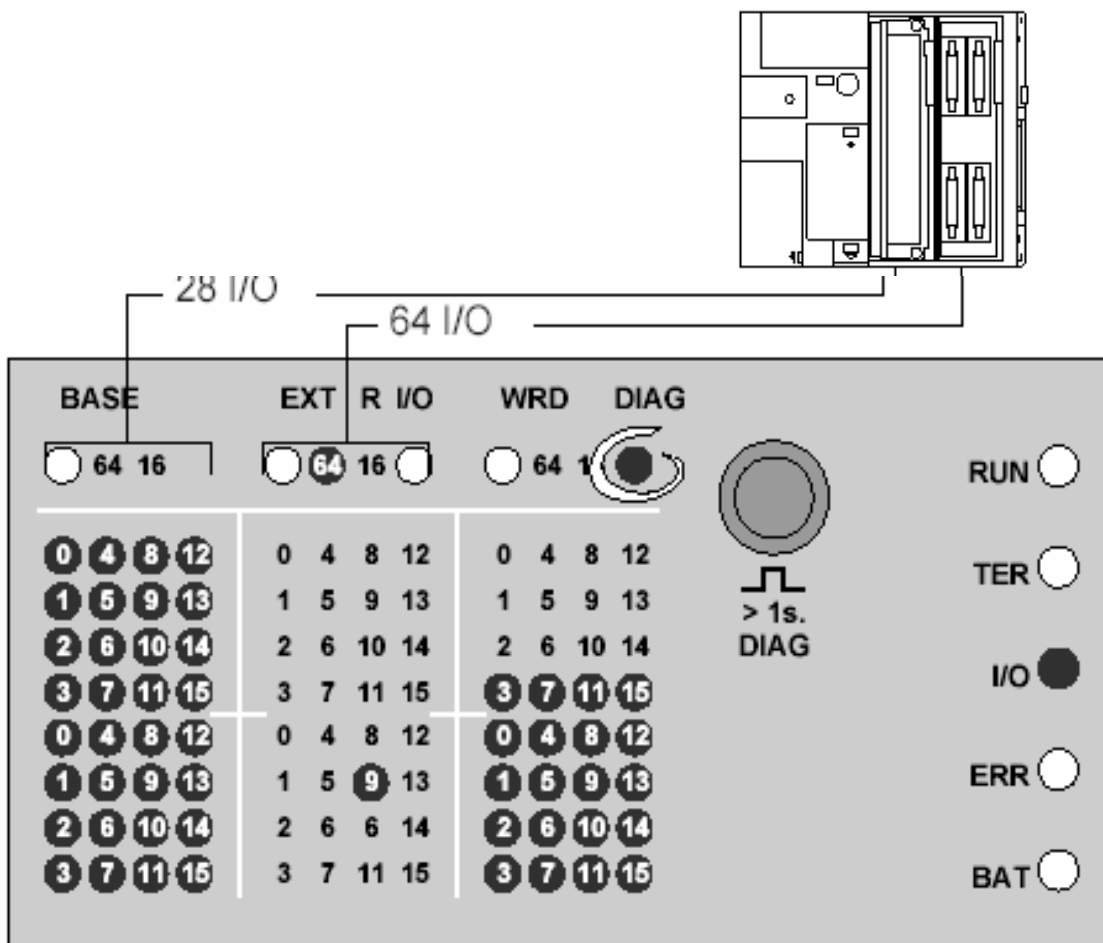
Quando in un alloggiamento è presente un modulo a 64 canali la spia 64 corrispondente è accesa. Una leggera pressione del pulsante permette di visualizzare sia i primi 16 ingressi e le prime 16 uscite (solo la spia 64 è accesa), sia i 16 ingressi e le 16 uscite successive (le spie 64 e 16 sono accese).



Visualizzazione dei guasti(modalità DIAG)

La modalità di diagnostica è accessibile con una lunga pressione (> 1s) del pulsante che accende la spia DIAG.

- Ingresso o uscita guasta(alimentazione guasta, scollegamento di una uscita, ecc.): la spia corrispondente lampeggia velocemente.
- Modulo guasto(modulo non presente, non compatibile con la configurazione, fuori servizio, ecc.): tutte le spie corrispondenti lampeggiano lentamente(16 spie per il formato a modulo singolo,32 spie per un modulo a 28I/O o a 64I/O).



Nota: eseguire un avvio a freddo del PLC per cancellare i guasti di I/O.

IL SOFTWARE PL7

Generalità

I controllori **MODICON TSX TELEMECANIQUE** utilizzano il software di programmazione **PL7**, con filosofia Windows, il cui linguaggio è conforme alle norme IEC 1131.

In particolare il software **PL7-07** gestisce il TSX Nano, il software **PL7 Micro** gestisce il TSXMicro, i software **PL7 Junior** e **PL7 Pro** gestiscono invece i PLC TSX Micro e TSX Premium.

Il software PL7 consente due tipi di struttura del programma:

la struttura **monotask** che prevede il task principale (MAST), struttura di default;

la struttura **multitask** che prevede il task principale, il task rapido (FAST) e le elaborazioni su evento (Evt).

I task(principale e rapido) di un programma PL7 sono composti da sezioni e da sottoprogrammi. Ogni sezione può essere programmata in un linguaggio appropriato all'elaborazione da attuare. I sottoprogrammi possono essere chiamati da qualsiasi sezione del task a cui appartengono o da altri sottoprogrammi dello stesso task.

Questa suddivisione consente di creare un programma strutturato e di creare, o inserire facilmente, dei moduli di programma.

I task principale e veloce (quando questo è programmato) sono attivi di default. Il task di evento è attivato quando si manifesta l'evento che gli è stato associato.

È possibile scegliere l'esecuzione del task master (in configurazione) come ciclica o come periodica. Il task rapido(FAST), che è prioritario rispetto al task master (MAST), è periodico per lasciare il tempo necessario all'esecuzione del task con priorità meno alta.

Inoltre, i trattamenti ad esso associati devono essere brevi per non penalizzare il task master. Come per il task master, il programma associato si compone di sezioni e di sottoprogrammi. I trattamenti di eventi consentono di abbreviare i tempi di risposta del software in seguito ad eventi di comando provenienti da alcuni moduli di funzioni specifiche. Questi trattamenti vengono eseguiti con priorità su tutti gli altri task. Si adattano ai trattamenti che richiedono tempi di reazione molto brevi rispetto all'evento verificatosi. Quando si verifica un evento all'inizio del ciclo del task veloce, quest'ultimo interrompe l'esecuzione dei task a più bassa priorità in modo da poter proseguire con l'elaborazione master; il task interrotto verrà ripreso al termine dell'elaborazione del task master.

Il software PL7 propone **4 linguaggi** di programmazione: **Lista Istruzioni**, **Testo Strutturato**(utilizzabile esclusivamente con il software PL7 Junior e PL7 Pro), **Schema a contatti**, **Schema Funzionale in Sequenza**.

Questi linguaggi possono essere utilizzati contemporaneamente nella stessa applicazione: una sezione di programma può essere scritta in Ladder, un'altra in testo strutturato, ecc.

Creazione di una applicazione

Il modo di funzionare del PL7 MICRO segue le stesse modalità di Windows e prevede, quindi, l'uso di mouse o tastiera, menu a tendina, barre degli strumenti.

Dopo avere avviato **PL7 Micro**, sullo schermo appare la finestra principale che permette di gestire una applicazione(permette di creare, aprire, trasferire una applicazione, collegarsi con il TSX Micro).

Sul bordo superiore della finestra si trova la **barra dei titoli** che porta il nome dell'applicazione attiva.

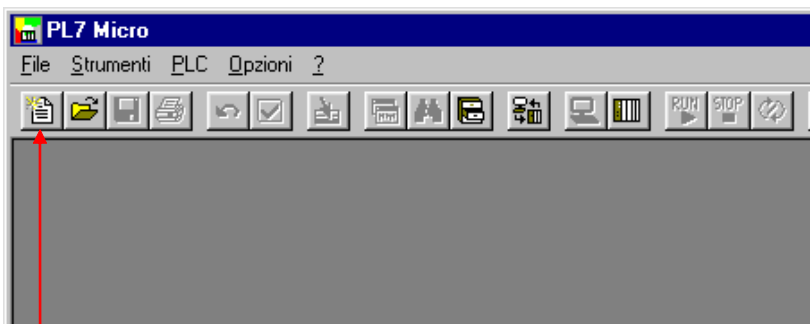
Sotto la barra dei titoli vi è la **barra dei menu** composta dalle voci

Φιλε Στρυμεντι ΠΑΧ Οπζιονι Γυιδα

ad ogni menu corrisponde una finestra coi relativi comandi.

Al di sotto della barra dei menu si trova la **barra degli strumenti** costituita da una serie di icone o pulsanti di comando che permettono di attivare routine prestrutturate che eseguono rapidamente determinate procedure.

La creazione di una applicazione permette di selezionare tipo di PLC, tipo di processore, tipo di scheda di memoria(processore e scheda di memoria possono essere modificati in seguito(vedi configurazione):

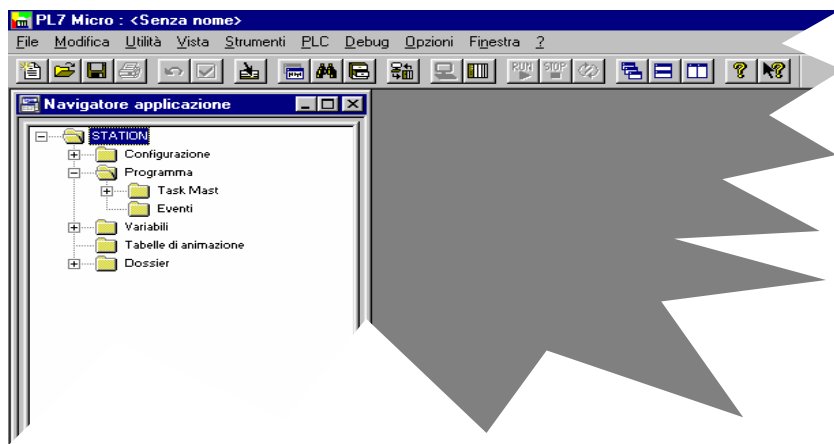


Dalla finestra principale selezionare il comando

File / Nuovo

oppure l'icona

Crea una nuova applicazione



Si estende sia la barra dei menù che degli strumenti, in basso compare la **barra di stato**, che indica lo stato di funzionamento

dell'applicazione, compare il navigatore applicazione che consente di presentare, con struttura ad albero, il contenuto di un'applicazione PL7.

Poiché il controllore TSX MICRO può avere diverse configurazioni(vedi Hardware TSX Micro, Classificazione PLC), quando si crea una applicazione bisogna configurarla in modo identico al modello di PLC con cui l'applicazione stessa deve lavorare.

Configurare vuol dire definire una forma per il controllore da usare scegliendo tra le varie opzioni offerte dal costruttore relative alle risorse hardware e software del controllore stesso. Così la configurazione hardware consente di definire tipo di processore e di schede di ingressi/uscite e loro caratteristiche: tipo di task, modo di funzionamento, filtraggio,ecc. La configurazione software permette di definire numero blocchi funzione predefiniti e dimensione variabili.

Allora occorre conoscere i dati relativi a base controllore e moduli allocati nel rack del nostro PLC e, prima di iniziare a lavorare col programma, configurare allo stesso modo l'applicazione.

A tale scopo selezionare dal menu il comando

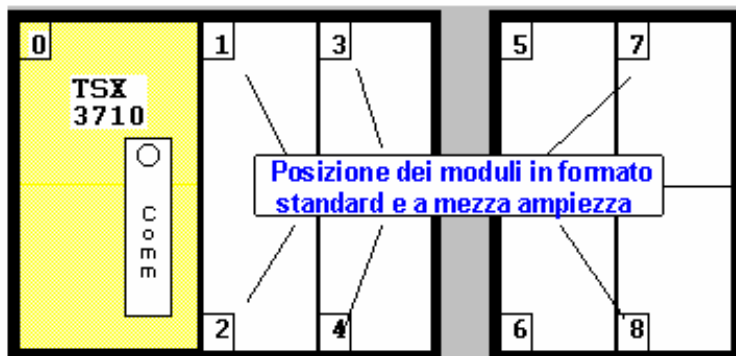
Strumenti/Configurazione

oppure il comando

Strumenti/Navigatore applicazione/Configurazione

Il programma visualizza la configurazione del PLC per il processore selezionato.

Per il processore TSX 37-10



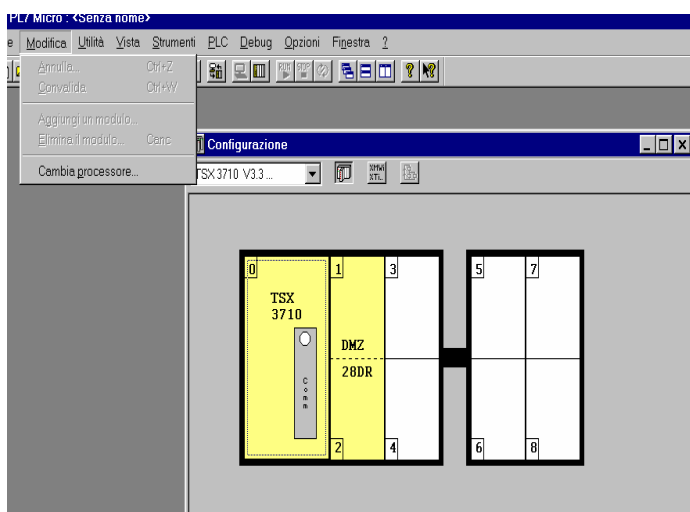
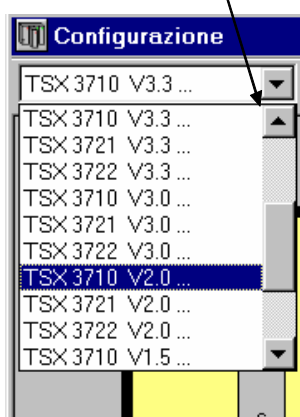
Dove:

- la posizione 0 è occupata dal processore
- le posizioni da 1 a 8 possono essere occupate dai moduli di I/U

Configurazione Hardware

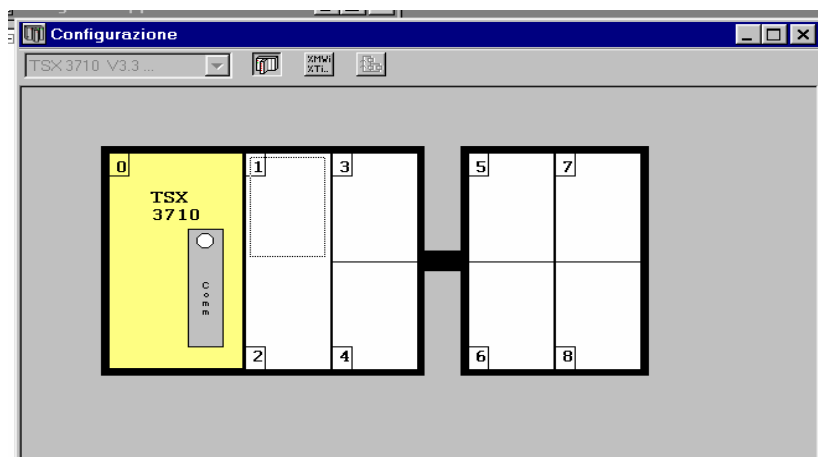
Scelta del processore TSX Micro:

- Selezionare il tipo di processore con il comando **Modifica/Cambia processore** oppure col pulsante
- Confermare la scelta.

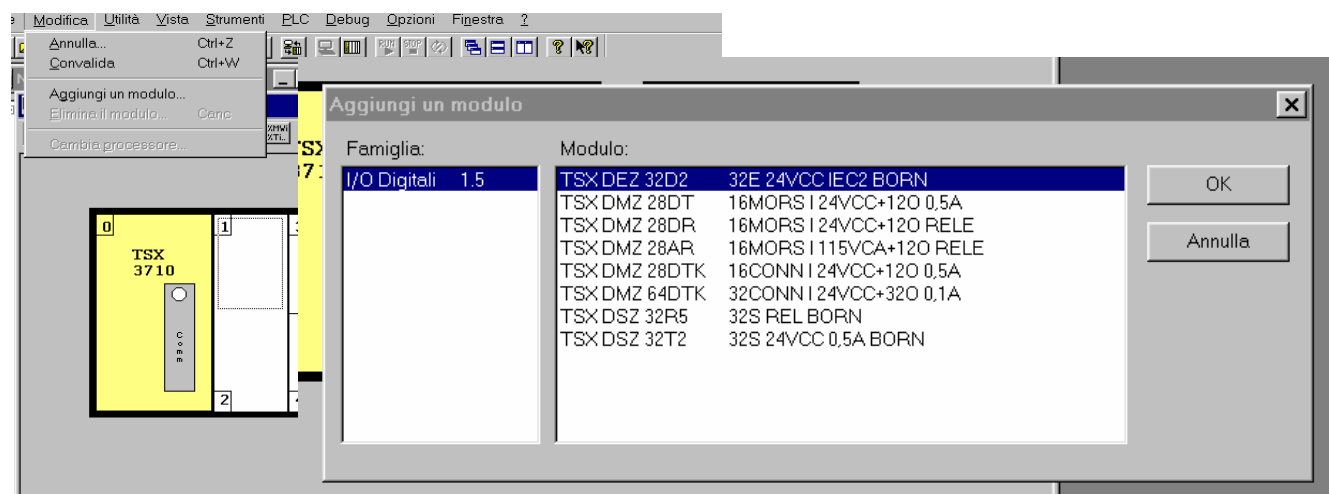


Scelta dei moduli I/O:

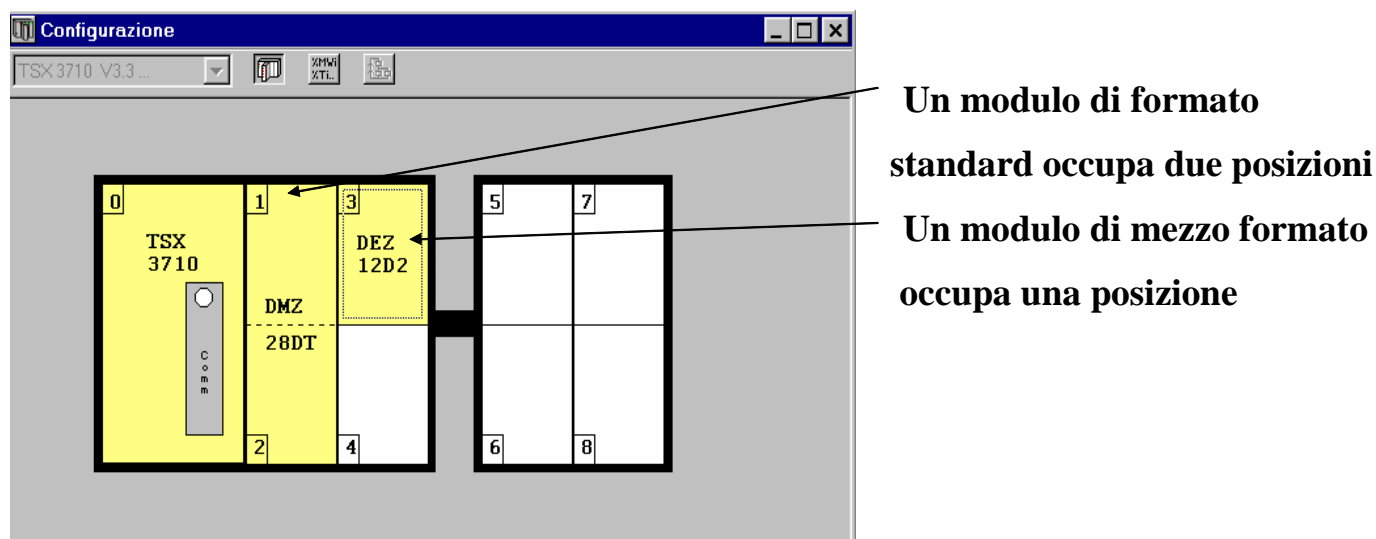
Selezionare la posizione del modulo da configurare con un clic sul mouse o coi tasti freccia



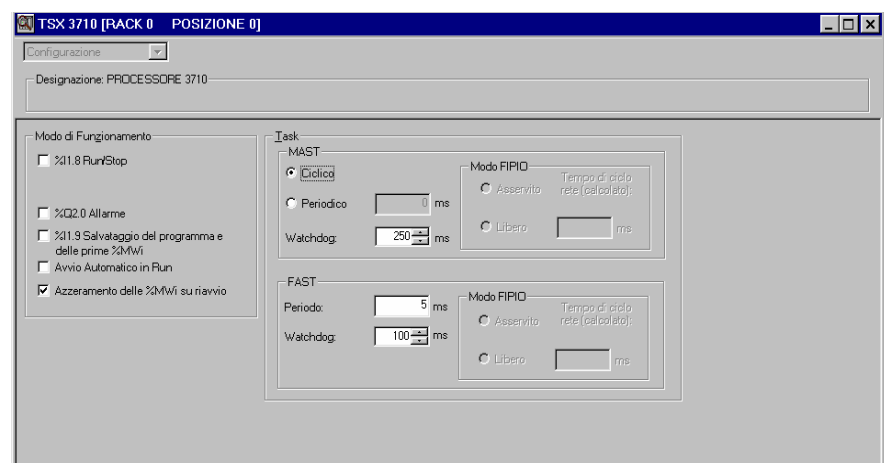
Selezionare il comando **Modifica/ Aggiungi un modulo**, oppure fare doppio clic o premere invio sulla posizione selezionata: compare la lista dei moduli configurabili



Selezionare famiglia e modulo(con mouse o tasti freccia) e confermare.



Caratteristiche processore: modo di funzionamento, tipo di task



Caratteristiche moduli I/U: task, filtraggio,ecc



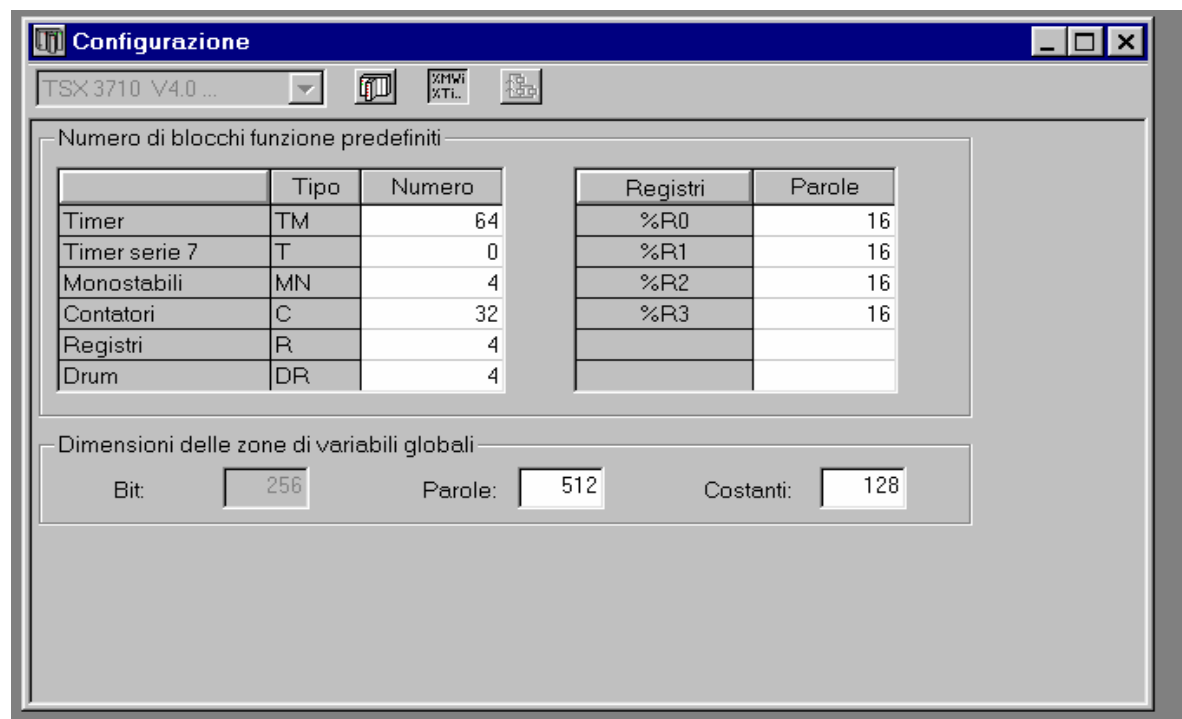
Memorizzazione dello stato su impulso da 0 a 1 e da 1 a 0:

Quando un ingresso è troppo “rapido” (tempo di permanenza dell’ ingresso < del tempo di ciclo) perchè sia memorizzato come stato logico per un tempo sufficiente ad essere registrato occorre aumentare il tempo di permanenza. Ciò si fa mediante filtraggio (si può fare sui primi 4 ingressi). Filtraggio programmabile sugli ingressi (es. proximiti con tempi di permanenza ridotti): ad es. tempo di filtraggio 0.1ms significa che è possibile leggere cambiamenti di stato > 0.1ms. Tempi troppo

ridotti possono essere influenzati da disturbi(es. un finecorsa con rimbalzi di tipo meccanico non può avere tempi di filtraggio troppo ridotti).

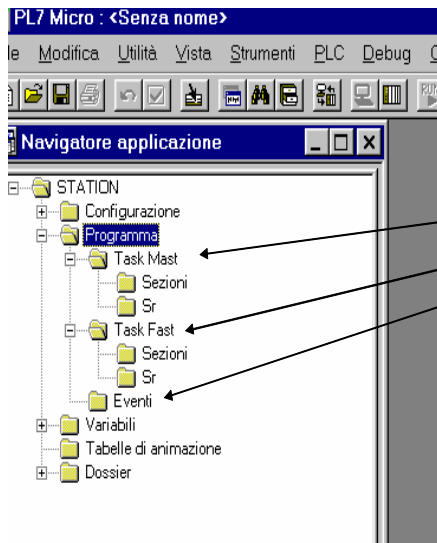
Configurazione Software

Caratteristiche: numero blocchi funzione predefiniti, dimensione variabili.

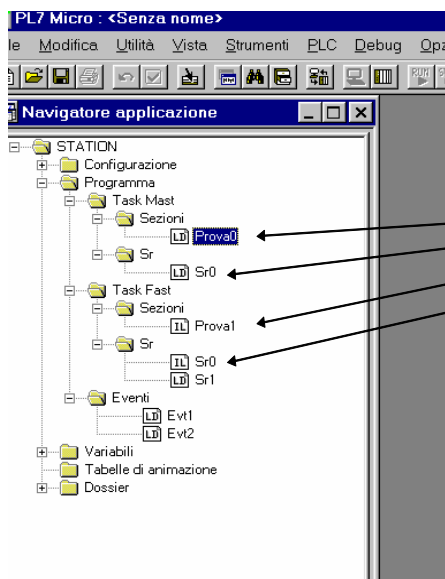


Creazione del programma

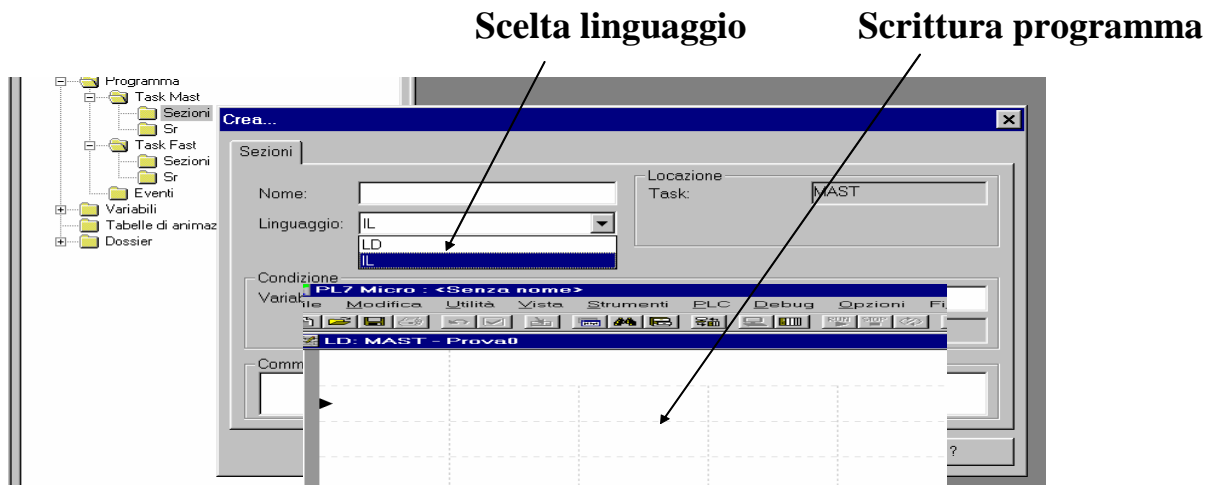
Questa fase consente di scrivere il programma scegliendo task, modulo, linguaggio.



Scelta task



Scelta modulo



Terminata l'applicazione, per verificarne il funzionamento il programma permette di creare le **tabelle di animazione**. In alternativa alla tabella d'animazione la verifica dell'efficienza dell'applicazione si effettua col simulatore.

Queste tabelle consentono di conoscere il valore delle variabili, di visualizzare, in modalità collegato, l'evoluzione del contenuto delle variabili e di forzare le variabili bit(cioè si può assegnare un valore d'ingresso ad ogni variabile).

Vi si accede da

Navigatore Applicazione/Tabella di animazione.

Ognuna di queste tabelle, che può contenere fino a 40 variabili, può essere creata manualmente o in modo automatico, in modalità locale o collegato.

Creazione manuale:

Premere il tasto destro e fare clic su **Crea**.

Immettere le variabili sotto forma di variabile o di simbolo e confermare con INVIO: verrà visualizzato il valore corrente della variabile.

Creata la tabella, chiudere la casella di immissione.

Indicare il nome della tabella, specificare l'eventuale commento e confermare con INVIO.

Creazione automatica:

Accedere al modulo di programma per il quale viene eseguita la creazione della tabella.

Posizionarsi sul rung(LD), la frase(IL) o l'istruzione(ST) e selezionare il comando **Modifica / Seleziona rung(LD)** o **Modifica / Seleziona frase(IL)**.

Selezionare il comando **Utilità / Inizializza tabella d'animazione**.

Le tabelle così create contengono l'insieme degli oggetti dell'elemento di programma selezionato e possono essere successivamente modificate eliminando o aggiungendo nuove variabili.

Visualizzazione dello stato delle variabili:

In modo **collegato** è possibile attivare (comando **Utilità / Anima**) o disattivare (comando **Utilità / Interrompi animazione**) l'animazione di una tabella.

L'animazione viene realizzata in modo asincrono: tutti i valori degli oggetti vengono letti nel processore a fine del ciclo del task Mast.

E' possibile modificare in modo unitario il valore di una variabile, forzare o liberare il valore di un bit.

Modifica di una variabile:

Selezionare la variabile e posizionarsi sul campo **Valore corrente**.

Immettere il valore della variabile e confermare con **Modifica**.

Forzatura e liberazione delle variabili:

Questa funzione consente di modificare lo stato di tutti canali di un modulo o di parte di essi. La forzatura dei bit viene effettuata tramite le tabelle di animazione.

Selezionare la variabile da forzare e spostarsi sul campo **Valore corrente**.

Selezionare **Forza a 0** per forzare il bit a 0;

Selezionare **Forza a 1** per forzare il bit a 1 (la forzatura viene indicata dalla lettera F).

Liberazione dei bit in una tabella di animazione:

Selezionare la variabile da liberare e spostarsi sul campo Valore corrente.

Selezionare **Libera**.

Il software PL7 consente inoltre una **diagnostica** a diversi livelli dell'applicazione:

il **sistema**, il **processo**, il **PLC**, i **moduli**, il **programma**.

Per accedere alla diagnostica occorre essere in modalità collegato.

Le funzioni di diagnostica di PL7 sono distribuite tra i vari editor e strumenti dei software PL7. I software PL7 offrono una serie di strumenti completi per il debug delle applicazioni.

Il debug è l'insieme delle operazioni che servono per individuare e correggere errori in un programma o in un processo automatizzato. Gli strumenti di debug che consentono di accedere alle funzioni principali sono:

una barra di debug programma che consente, ad esempio, l'impostazione di punti d'arresto, l'esecuzione passo passo del programma, il monitoraggio dell'applicazione.

Per accedere selezionare **Debug / Barra di debug programma**

Una schermata di debug PCU che offre: informazioni sullo stato dell'applicazione, l'accesso alla diagnostica del programma e ai moduli funzioni specifiche, l'accesso all'aggiornamento e alla visualizzazione dell'orodattario.

Per accedere selezionare **Debug /Accedi alla schermata di debug PCU**

Il navigatore di debug Grafcet, che consente una vista gerarchica del grafico.

Per accedere selezionare il comando **Servizi / Debug Grafcet**

Una barra di debug Grafcet, che consente di visualizzare e/o modificare lo stato del grafico.

Per accedere selezionare il comando **Debug / Barra di debug Grafcet**

Infine il PL7 permette di produrre una documentazione del progetto compilando il **dossier** a cui si accede da **Navigatore Applicazione / Dossier**.

Oggetti del linguaggio PL7 e loro indirizzamento

Oggetti del linguaggio

Gli oggetti del linguaggio si identificano come oggetti di tipo **bit** e di tipo **parola**.

I principali oggetti di tipo bit sono i **bit di Ingresso**, i **bit di Uscita**, i **bit Interni**, i **bit di Sistema**, i **bit dei Blocchi funzione**, i **bit Grafcet**. Tra gli oggetti di tipo parola troviamo tutti i suddetti più le parole **Costante** e **Comune**.

La tabella seguente descrive i principali oggetti **booleani**

Bit	Descrizione	Esempi	Accesso in scrittura
Valori immediati	0 o 1 (False o True)	0	—
Ingressi/uscite	Questi bit sono le "immagini logiche" degli stati elettrici di ingressi/uscite. Sono posizionati nella memoria di dati e vengono aggiornati ad ogni scansione del task in cui sono configurati. Nota: I bit di ingressi/uscite non utilizzati non possono essere impiegati come bit interni.	%I23.5 %Q51.2	no sì
Interni	I bit interni permettono di memorizzare degli stati intermediari durante l'esecuzione del programma.	%M200	sì
Sistema	I bit di sistema da %S0 a %S127 sorvegliano il corretto funzionamento del PLC e lo svolgimento del programma applicazione.	%S10	Secondo i
Blocchi funzione	I bit dei blocchi funzione corrispondono alle uscite dei blocchi funzione standard o all'istanza di DFB. Queste uscite non possono essere collegate direttamente e vengono utilizzate come oggetto.	%TM8.Q	No
Estratti da parole	Il software PL7 offre la possibilità di estrarre uno dei 16 bit di un oggetto parola	%MW10:X5	Secondo il tipo di parole
Fasi e macrofasi Grafcet	I bit Grafcet di stato delle fasi, delle macrofasi e delle fasi di macrofase permettono di conoscere lo stato della fase i, della macrofase j o della fase i della macrofase j di Grafcet	%X21%X5.9	Sì Sì

La tabella seguente descrive i principali oggetti **parola**

Parole	Descrizione	Esempi	Accesso in scrittura
Valori immediati	Si tratta di valori algebrici di formato omogeneo a quello delle parole lunghezza semplice e parola lunghezza doppia (16 o 32 bit), che consentono di assegnare dei valori a queste parole.	2542	—
Ingressi/uscite	Si tratta di "immagini logiche" dei valori elettrici degli ingressi/uscite (esempio: ingressi/uscite analogici). Sono classificati nella memoria dei dati e aggiornati ad ogni esame del task in cui sono configurati	%IW23.5 %QW51.1	no sì
Interni	Sono destinati a memorizzare i valori in corso di programma. Sono classificati all'interno dello spazio Dati nella stessa zona memoria.	%MW10 %MD45	sì sì
Costanti	Memorizzano le costanti o i messaggi alfanumerici. Il loro contenuto può essere scritto o modificato solo con il terminale. Sono memorizzate nello stesso punto che il programma e possono pertanto avere come supporto la memoria FLASH EPROM.	%KW30	sì esclusivamente tramite terminale
Sistema	Queste parole svolgono svariate funzioni: § alcune specificano lo stato del sistema (tempo di funzionamento sistema e applicazione...) altre permettono di agire sull'applicazione (modalità di funzionamento, ...)	%SW5	secondo i
Blocchi funzione	Queste parole corrispondono ai parametri o ai valori correnti dei blocchi funzione standard o istanza di DFB.	%TM2.P	sì
Comuni	Sono destinate ad essere scambiate automaticamente su tutte le stazioni collegate alla rete di comunicazione	%NW2.3	sì
Grafcet	Le parole Grafcet consentono di conoscere il periodo di attività delle fasi.	%X5,T	sì

Formato dei valori

I valori delle parole possono essere codificati nei seguenti formati:

Tipo	Dimensioni	Esempio di valore	Limite inferiore	Limite superiore
Intero base 10	Lunghezza semplice	1506	-32768	+32767
	Lunghezza doppia	578963	-2 147 483 648	2 147 483 647
Intero base 2	Lunghezza semplice	2#1000111011111011011	2#10...0	2#01...1
	Lunghezza doppia	2#10001110111110110111 11111011111011111	2#10...0	2#01...1
Intero base 16	Lunghezza semplice	16#AB20	16#0000	16#FFFF
	Lunghezza doppia	16#5AC10	16#000000000	16#FFFFFFFF
Virgola mobile		-1.32E12	-3.402824E+38 (1)	-1.175494E-38(1)
			-1.175494E-38 (1)	3.402824E+38 (1)

Indirizzamento degli oggetti

Quando si scrive il programma ad ogni componente di ingresso e di uscita impiegato, nonché ad eventuali altri elementi che realizzano la logica di funzionamento, bisogna assegnare un **indirizzo PLC**.

L'indirizzo PLC è un codice che identifica in modo univoco ogni elemento funzionale del PLC, cioè ogni istruzione del programma (contatti (ingressi esterni e bit interni), bobine (uscite esterne e uscite interne), temporizzatori, contatori, ecc.) e lo associa ad una cella di memoria.

Indirizzamento degli oggetti bit

In breve

L'indirizzamento dei bit interni, sistema e fasi segue le seguenti regole:

%	M, S o X	i
Simbolo	Tipo di oggetto	Numero

Sintassi

La seguente tabella descrive i vari elementi che costituiscono l'indirizzamento.

Famiglia	Elemento	Valori	Descrizione
Simbolo	%	-	-
Tipo di oggetto	M	-	Bit interni destinati a memorizzare gli stati intermedi durante l'esecuzione del programma. Sono classificati all'interno dello spazio dati nella stessa zona memoria.
	S	-	Bit sistema (Vedere Manuale di Riferimento, Volume 2), questi bit svolgono più funzioni: <ul style="list-style-type: none">• alcuni specificano lo stato del sistema tramite lettura dei bit %Si (superamento del watchdog, ...),• altri permettono di agire sull'applicazione (inizializzazione Grafcet, ...).
	X	-	Bit di fase , i bit di fase (Vedi <i>Oggetti Grafcet</i> , p. 53) riportano lo stato delle attività delle fasi.
Numero	i	-	Il valore massimo del numero dipende dal numero di oggetti configurati.

Esempi:

- %M25 = bit interno numero 25
- %S20 = bit sistema numero 20
- %X6 = bit fase numero 6

Bit estratti da parole

Il software PL7 consente di estrarre uno dei 16 bit dalle parole lunghezza semplice. La variabile della parola viene così completata dal rango del bit estratto secondo la sintassi riportata qui di seguito:

PAROLA : X	j
Indirizzo della parola	Posizione j= 0 a 15 rango del bit nella parola.

Esempi:

- %MW10:X4 = bit numero 4 della parola interna %MW10
- %QW5.1:X10 = bit numero 10 della parola di uscita %QW5.1

Nota: L'estrazione dei bit dalle parole può essere eseguita anche sulle parole indicizzate.

Indirizzamento degli oggetti di moduli di ingressi/uscite del TSX 37

Per quanto riguarda l'indirizzo degli oggetti di moduli di ingressi/uscite questo è di tipo geografico, cioè dipende dal numero del rack(nel caso si impieghi una configurazione con più rack), dalla posizione fisica del modulo nell'alloggiamento del rack, dal numero di via o canale del modulo.

L'indirizzamento è definito nel seguente modo:

%	I,Q,M,K	X,W,D,F	X	i	r
Simbolo	Tipo di oggetto	Formato	Posizione del modulo	N° canale del modulo	Posizione della parola

La seguente tabella descrive i diversi elementi che costituiscono l'indirizzamento:

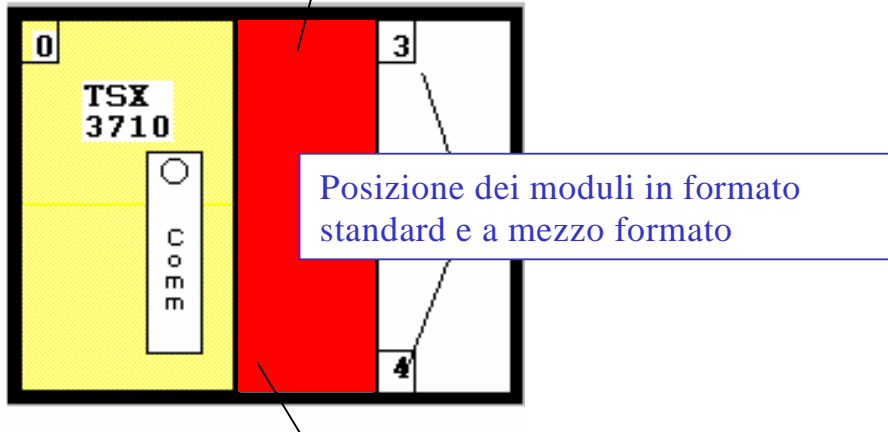
Famiglia	Elemento	Valori	Descrizione
Simbolo	%	-	-
Tipo di oggetto	I	-	Immagine dell'ingresso fisico del modulo
	Q	-	Immagine dell'uscita fisica del modulo Queste informazioni sono scambiate in modo automatico ad ogni ciclo del task ai quali esse sono associate.
	M	-	Variabile interna Queste informazioni di lettura o scrittura sono scambiate alla richiesta dell'applicazione
	K	-	Costante interna Queste informazioni di configurazione sono disponibili solo in lettura
Formato (dimensione)	X	-	Booleano Per gli oggetti di tipo booleano, la X può essere omessa
	W	16 bit	Lunghezza semplice
	D	32 bit	Lunghezza doppia
	F	32 bit	Virgola mobile. Il formato virgola mobile utilizzato è quello della norma IEEE Std 754-1985 (equivalente IEC 559).
Posizione del modulo	x	da 0 a 8 da 0 a 10	TSX 37-10 TSX 37-21/22 Nota: un modulo in formato standard (che occupa 2 posizioni) viene indirizzato come 2 moduli in 1/2 formato sovrapposti (vedere le spiegazioni qui di seguito).
N. canale	i	0 - 31oMOD	Numero di canale del modulo.MOD: canale riservato alla gestione del modulo e dei parametri comuni a tutti i canali.

Posizione	r	0 - 127oERR	Posizione del bit nella parola ERR: indica un errore del modulo o del canale.
------------------	---	-------------	--

Esempio: processore TSX 37-10

scheda TSX DMZ 28DR (16 ingressi a 24V cc , 12 uscite a relè)

16 ingressi (canali 0-15)



12 uscite (canali 0-11)

la CPU occupa la posizione 0 del rack ed ha indirizzo 0,

il modulo TSX DMZ 28DR formato standard con 16 ingressi e 12 uscite alloggia

all'indirizzo 1,2 (un modulo di formato standard viene indirizzato come due moduli di mezzo formato sovrapposti. Ad esempio un modulo di 28 I/U che occupa le posizioni 1e 2 è visto come due moduli di mezzo formato: un mezzo modulo di 16 ingressi in posizione 1 e un mezzo modulo di 12 uscite in posizione 2).

La sintassi degli I/U è dunque la seguente:

da %I1.0 a %I1.15 per gli ingressi

da %Q2.0 a %Q2.11 per le uscite

La tabella seguente riporta due esempi di indirizzamento di oggetti di un modulo standard 28 I/U che occupa le posizioni 3 e 4.

Oggetto	Descrizione
%I3.6	Canale di ingresso numero 6 del modulo in posizione 3
%Q4.2	Canale di uscita numero 2 del modulo in posizione 4

%	I,Q,M,K	X,W,D,F	X	i	r
---	---------	---------	---	---	---

Simbolo	Tipo di oggetto	Formato	Posizione del modul	N° canale del modulo	Posizione
%	I Q		3 4	6 2	

La tabella seguente riporta alcuni esempi di indirizzamento di oggetti.

Oggetto	Descrizione
%I1.5	Canale di ingresso numero 5 del modulo di ingressi/uscite che si trova in posizione 1
%MW2.0.3	Parola di stato di posizione 3 del canale 0 del modulo di ingressi/uscite che si trova in posizione 2
%I5.MOD.ERR	Informazione sull'errore del modulo di ingressi/uscite situato in posizione 5

%	I,Q,M,K	X,W,D,F	X	i	r
Simbolo	Tipo di oggetto	Formato	Posizione del modulo	N° canale del modulo	Posizione
%	M	W	2	0	3

Esempi

Caso dei moduli in formato standard

Sono indirizzati come 2 moduli in 1/2 formato sovrapposti.

Ad esempio, un modulo di 64 I/O che occupa le posizioni 5 e 6, è visto come 2 moduli 1/2 formato:

- un 1/2 modulo di 32 ingressi in posizione 5
- un 1/2 modulo di 32 ingressi in posizione 6

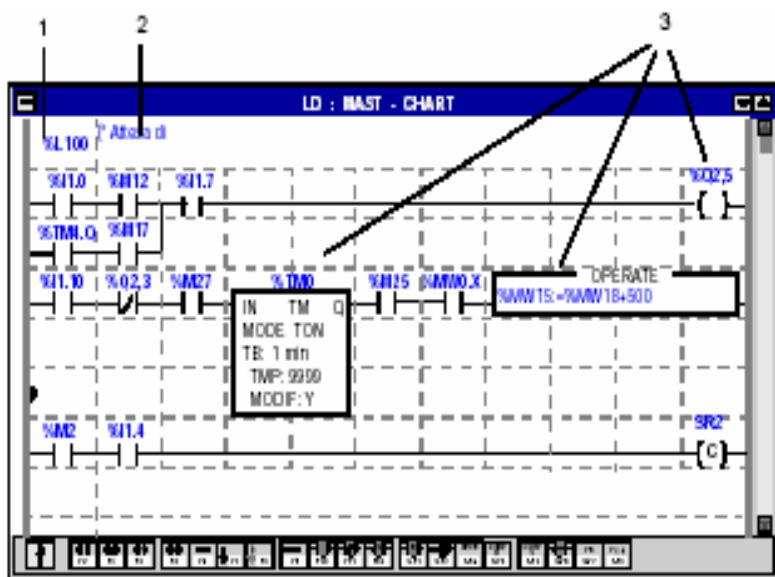
La tabella seguente descrive la codifica Posizione/14numero di canale a seconda del modulo.

Modulo	1/2 formato			Formato standard			
	4U	8I	12I	28I/U	32I	32U	64I/U
Numero di canale	da 0 a 3	da 0 a 7	da 0 a 11	da 0 a 15 (I)	da 0 a 15 (I)	da 0 a 15 (U)	da 0 a 31 (I)
				da 0 a 11 (U)	da 0 a 15 (I)	da 0 a 15 (U)	da 0 a 31 (U)
Indirizzamento: Posizione/14numero di canale (x=posizione)	x.0 verso x.3	x.0 verso x.7	x.0 verso x.11	x.0 verso x.15	x.0 verso x.15	x.0 verso x.15	x.0 verso x.31
				(x+1).0 verso (x+1). 11	(x+1).0 verso (x+1). 15	(x+1).0 verso (x+1).15	(x+1).0 verso (x+1).31

Linguaggio Ladder Diagram del PL7 Micro.

Un programma scritto in linguaggio Ladder si compone di una serie di rung eseguiti in sequenza dal PLC.

Esempio di un rung PL7



Componenti del rung :

1=etichetta (label)

2= commento

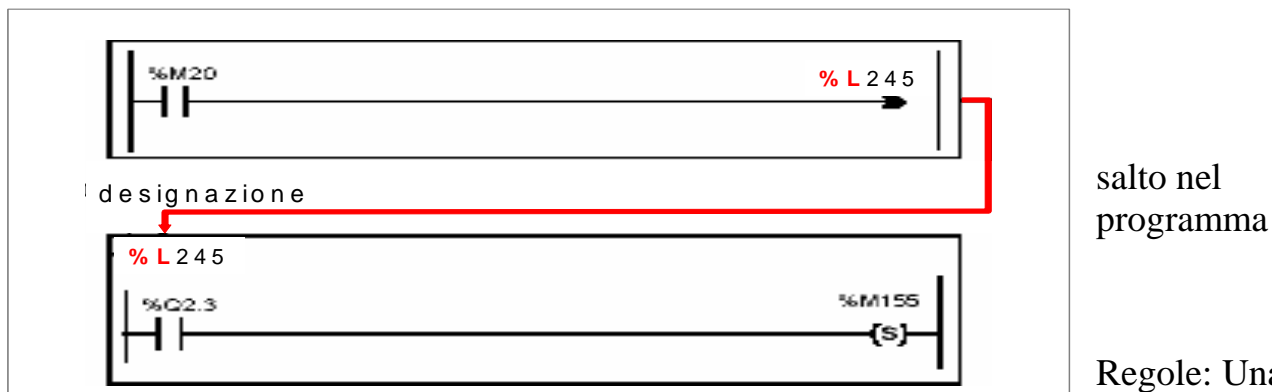
3= elementi grafici

Un rung PL7 Micro è costituito da un certo numero di elementi grafici disposti su una griglia di 7 linee e 11 colonne, delimitata da due barre di potenziale verticali e suddivisa in una zona test, in cui si scrivono gli ingressi, e una zona azione che applica il risultato della zona test e produce un'uscita(un rung può essere costituito da più rung alimentati e indipendenti).

I componenti di un rung sono l'etichetta, il commento e gli elementi grafici.

L'**etichetta(label)** individua un rung in un programma(per poterlo, ad esempio, collegare dopo un salto di programma da un altro rung). La sintassi di una etichetta è %Li, con i compreso tra 0 e 999 e si posiziona sulla parte superiore sinistra, davanti alla barra di potenziale.

I rung seguenti illustrano l'uso di una label:



variabile di label (etichetta di indirizzo) può essere assegnata ad un solo rung all'interno di una stessa entità di programma.

Occorre dare un indirizzo etichettando un rung per consentire il collegamento dopo un salto di programma.

L'ordine delle variabili delle label è indifferente, in quanto al momento dell'esame il sistema prende in considerazione l'ordine di immissione dei rung.

Il **commento** serve per specificare un programma o per aiutare a comprendere ciò che un rung deve eseguire. Sintassi : il commento è integrato al rung e contiene al massimo 222 caratteri alfanumerici(contenuti in tre righe: digitare CTRL+ENTER per andare a capo) preceduti e seguiti dai caratteri (* e *). I commenti vengono visualizzati nella zona riservata a tale scopo nella parte superiore del rung. In caso di eliminazione di un rung, viene eliminato anche il commento ad esso associato. I commenti vengono memorizzati nel PLC e l'utente può accedervi in qualsiasi momento.

Etichetta e commento sono facoltativi.

Gli **elementi grafici** sono istruzioni del linguaggio Ladder e sono costituiti da:

contatti che rappresentano gli ingressi(pulsanti, sensori, contatti ausiliari, ecc.) e le variabili interne del PLC.

bobine che rappresentano le uscite(relè, spie luminose, ecc.)e le variabili interne del PLC.

blocchi funzione che rappresentano le funzioni di automazione come temporizzatori, contatori, ecc.

blocchi operazione che rappresentano le funzioni aritmetiche e logiche.

Gli elementi grafici dei **contatti** sono programmati in zona **test** e occupano una cella del rung: 1 riga di altezza e 1 colonna di larghezza.

Contatto in chiusura		Contatto passante quando l'oggetto bit che lo pilota si trova allo stato 1.
Contatto in apertura		Contatto passante quando l'oggetto bit che lo pilota si trova allo stato 0.
Contatto di rilevamento del cambiamento di stato	 	<p>Fronte di salita: rilevamento del passaggio da 0 a 1 dell'oggetto bit che lo pilota.</p> <p>Fronte di discesa: rilevamento del passaggio da 1 a 0 dell'oggetto bit che lo pilota.</p>

Gli elementi

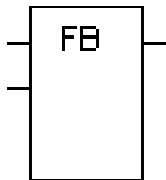
grafici delle **bobine** sono programmati in zona **azione** ed occupano una cella:

1 riga di altezza e 1 colonna di larghezza.

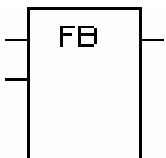
Designazione	Grafica	Funzioni
Bobina diretta		L'oggetto bit associato prende il valore del risultato della zona test
Bobina inversa		L'oggetto bit associato prende il valore inverso del risultato della zona test
Bobina di attivazione		L'oggetto bit associato è impostato a 1 quando il risultato della zona test è impostato a 1
Bobina di Reset		L'oggetto bit associato è impostato a 0 quando il risultato della zona test è impostato a 1
Salto condizionale ad un altro rung (JUMP)		<p>L'oggetto bit associato è impostato a 0 quando il risultato della zona test è impostato a 1. Consente di eseguire un collegamento ad un rung etichettato, a monte o a valle. I salti sono effettivi solo all'interno di una stessa entità di programmazione (programma principale, sottoprogramma,...).</p> <p>L'esecuzione di un salto provoca:</p> <ul style="list-style-type: none"> ▪ l'arresto dell'esame del rung in corso ▪ l'esecuzione del rung etichettato richiesto <p>il non esame della parte del programma che si trova tra l'azione del salto e il rung designato.</p>
Bobina diesis		Proposta in linguaggio Grafset, utilizzata durante il diesis, la programmazione delle ricettività associate alle transizioni provoca il passaggio alla fase seguente.
Bobina che chiama un sottoprogramma (CALL)		Consente di eseguire un collegamento all'inizio del sottoprogramma quando il risultato della zona di test sottoprogramma è impostata a 1.

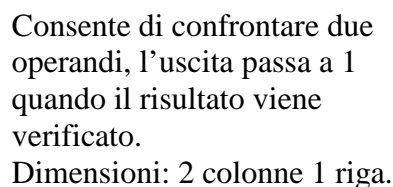
		L'esecuzione di una chiamata ad un sottoprogramma provoca: <ul style="list-style-type: none"> ▪ l'arresto dell'esame del rung in corso, ▪ l'esecuzione del sottoprogramma, ▪ la ripresa dell'esame del rung interrotto.
Ritorno dal sottoprogramma	< RETURN >	Riservata al sottoprogramma SR, permette il ritorno al modulo chiamante quando il risultato della zona di test è impostato a 1
Arresto programma	< HALT >	Provoca l'arresto dell'esecuzione del programma quando il risultato della zona di test è impostato a 1

Gli elementi grafici dei **blocchi funzione standard** si programmano in zona **test** ed occupano una dimensione la cui altezza è di massimo 16 righe e la cui larghezza di 3 colonne:

Designazione	Grafica	Funzione
Temporizzatore, Contatore, Programmatore ciclico, Registro, Monostabile		Utilizzano ingressi e uscite che consentono di collegarli agli altri elementi grafici per realizzare le funzioni di automazione

Gli elementi grafici dei **blocchi di confronto** e dei **blocchi operazione** si programmano rispettivamente in zona test e zona azione ed occupano le dimensioni di seguito riportate:

Designazione	Grafica	Funzione
Blocco di confronto verticale		Consente di confrontare due operandi e secondo il risultato l'uscita corrispondente passa a 1. Dimensioni: 2 colonne, 4 righe.
Blocco di confronto orizzontale		



Realizza le operazioni aritmetiche e logiche. Richiama la sintassi del linguaggio letterale strutturato. Contiene al massimo 4096 caratteri. Dimensioni: 4 colonne, 1 riga.

IN TM Q

MODE: TON

TB: 1mn

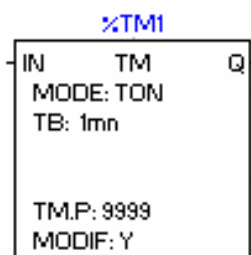
TM.P: 9999

MODIF: Y

Ton: questa modalità consente di gestire i ritardi del collegamento. Il ritardo è programmabile e può essere modificato o meno da terminale.

Charakteristika

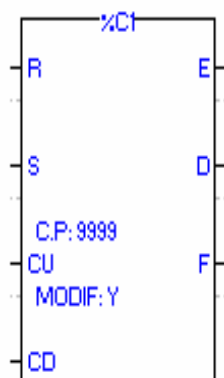
50



Input "Attivazione"	IN	Avvia il temporizzatore sul fronte di salita (modalità TON o TP) o sul fronte di discesa (modalità TOF).
Output "Temporizzatore"	Q	Bit associato %TMI.Q; l'impostazione a 1 dipende dalla funzione realizzata: TON, TOF o TP.

(1) %TMI.V può essere modificata da terminale.

Contatore avanti/indietro %Ci

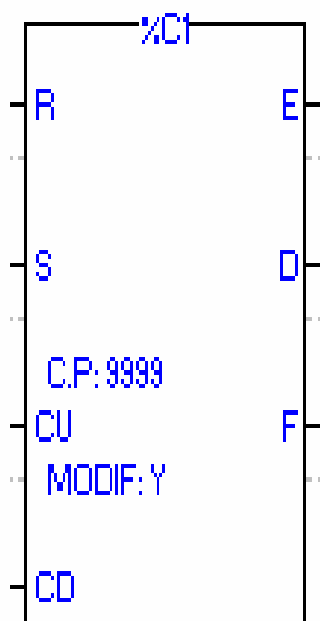


Il blocco funzione Contatore avanti/indietro consente di eseguire il conteggio avanti o indietro degli eventi. Le due operazioni possono essere eseguite contemporaneamente.

Contatore avanti/indietro %Ci

Caratteristiche

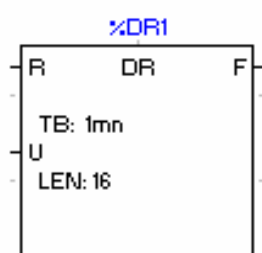
Numero del contatore	%Ci	Da 0 a 31
Valore corrente	%Ci.V	Parola incrementata o decrementata in funzione degli input CU e CD. Può essere letta e testata, ma non scritta con il programma (1).
Valore di preselezione	%Ci.P	$0 \leq \%Ci.P \leq 9999$. Parola che può essere letta, testata, scritta. Il valore predefinito è 9999.
Regolazione da terminale (MODIF)	Y/N	Y: possibilità di modifica del valore di preselezione in regolazione. N: nessun accesso in regolazione.
Input azzeramento	R	Nello stato 1 : %Ci.V = 0.
Input preselezione	S	Nello stato 1: %Ci.V = %Ci.P.
Input conteggio	CU	Incremento %Ci.V sul fronte di salita.
Input conteggio inverso	CD	Decremento %Ci.V sul fronte di salita.



Contatore avanti/indietro %Ci		
Overflow dell'uscita	E (Empty)	Il bit associato %Ci.E=1 (impostato a 1 quando %Ci.V è uguale a 9999; impostato a 0 se il contatore continua il conteggio indietro); quando il conteggio supera %Ci.V passa da 0 a 9999, %S18=1. Quando il conteggio supera (%Ci.V passa da 9999 a 0) %S18=1.
Raggiunta uscita preselezione	D (Done)	Il bit associato %Ci.D=1, quando %Ci.V=%Ci.P.
Overflow dell'uscita	F (Full)	Il bit associato %Ci.F=1 quando %Ci.V passa da 9999 a 0 (impostato a 1 quando %Ci.V è uguale a 0; impostato a 0 se il contatore continua il conteggio).

(1) %Ci.V può essere modificata da terminale.

Programmatore a tamburo - Drum %DRi

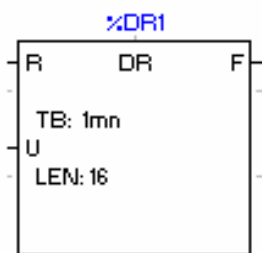


Il programmatore a tamburo (Drum) cambia passo in funzione degli eventi esterni e presenta un principio di funzionamento simile al programmatore a camme. A ciascun passo, il punto superiore di una camma indica un ordine utilizzato dall'automatismo. Nel programmatore a tamburo, i punti superiori saranno contrassegnati da uno stato 1 al livello di ciascun passo e saranno assegnati a dei bit di output %Qi.j o interni %Mi denominati bit d'ordine..

Programmatore a tamburo - Drum %DRi

Caratteristiche

Numero	%DRi	Da 0 a 7
Numero del passo	LEN	Da 1 a 16 (valore predefinito 16)
Base dei tempi	TB	1mn, 1s, 100ms, 10ms (valore predefinito 1mn)
Tempo sequenza o durata del passo in corso	%DRi.V	0 ≤ %DRi.V ≤ 9999. Parola impostata a 0 a ciascuna modifica del passo. Può essere letta, testata, ma non scritta. La durata è uguale a %DRi.V ~ TB.



Programmatore a tamburo - Drum %DRI		
Input "avanzamento"	U (UP)	Sul fronte di salita determina l'avanzamento di un passo del programmatore e l'aggiornamento dei bit d'ordine.
Output	F (FULL)	Indica che l'ultimo passo definito è in corso. Il bit %DRI.F associato può essere testato (%DRI.F=1 se %DRI.S=numero del passo è configurato - 1).
Stato di un passo	%DRI.Wj	Parola di 16 bit che definisce gli stati del passo j del programmatore i. Può essere letto, testato, ma non scritto.
Bits d'ordine		Output o bit interni associati al passo (16 bit d'ordine).



I PLC, come tutte le apparecchiature digitali, operano secondo una logica binaria in cui esistono due soli stati: "1" e "0".

I PLC realizzano un automatismo risolvendo delle funzioni logiche secondo un programma(inserito dall'operatore) e le variabili di ingresso il cui stato logico è definito dalla posizione dei pulsanti, finecorsa, ecc.

Una funzione logica o booleana è una relazione che lega il valore di una variabile di uscita (var. dipendente) ai valori di una o più variabili di ingresso (var. indipendenti).


Per un programma scritto in linguaggio Ladder valgono, dunque, le regole dell'algebra di Boole in cui agli elementi grafici della zona test(variabili di ingresso)si applicano gli operatori logici fondamentali: AND, OR, NOT(prodotta logico, somma logica, negazione)
Ad ogni contatto dello schema Ladder (così come ad ogni bobina) è associato un bit di memoria.


Convenzioni:

- il **contatto aperto**  assume lo stato logico 0



- il **contatto negato** assume lo stato logico 1
- si indica con livello logico 1 il segnale elettrico che fa cambiare di stato un contatto
- si indica con livello logico 0 la mancanza di segnale o il segnale che non fa commutare il contatto.

Ad esempio il livello logico 1 si identifica con la pressione del dito sul pulsante che chiude il circuito di alimentazione di un ingresso. Allora:

Mandare il segnale 1 ad un contatto aperto  (cioè impostare a 1 il bit associato a tale contatto) vuol dire chiuderlo e fare uscire il segnale 1

Mandare il segnale 1 ad un contatto NC  (cioè impostare a 1 il bit associato a tale contatto) vuol dire aprirlo e fare uscire il segnale 0.

I simboli 1 e 0 non sono numeri in senso aritmetico, ma indicano lo stato logico della variabile di ingresso e/o dell'uscita.

I contatti fronte di salita e fronte di discesa si utilizzano per ovviare a problemi che possono insorgere quando il segnale d'ingresso viene mantenuto per un tempo relativamente lungo, più lungo rispetto alla durata di un ciclo di scansione. Infatti: il contatto aperto  riporta lo stato logico dell'ingresso per tutta la sua durata, il contatto P (N), rilevando il cambiamento di stato, mantiene lo stato logico dell'ingresso solo per la durata di un ciclo.

Ad esempio, nel conteggio di scatole che passano su un nastro, con l'uso di un contatto aperto il passaggio di una scatola lunga mantiene a 1 il livello del segnale d'ingresso anche nel ciclo successivo e la stessa scatola viene contata due volte; allora, per evitare un conteggio errato, con il contatto P l'ingresso va a 0 anche se il pezzo è ancora presente nel ciclo successivo.

Gli elementi grafici di **collegamento** consentono di collegare gli elementi grafici di test e di azione.

Designazione	grafica	Funzione
Connessione orizzontale		Consente di collegare in modo seriale gli elementi grafici di test e di azione tra due

barre di potenziale

Connessione verticale

Consente di collegare in parallelo gli elementi grafici di test e di azione

Derivazione corto circuito

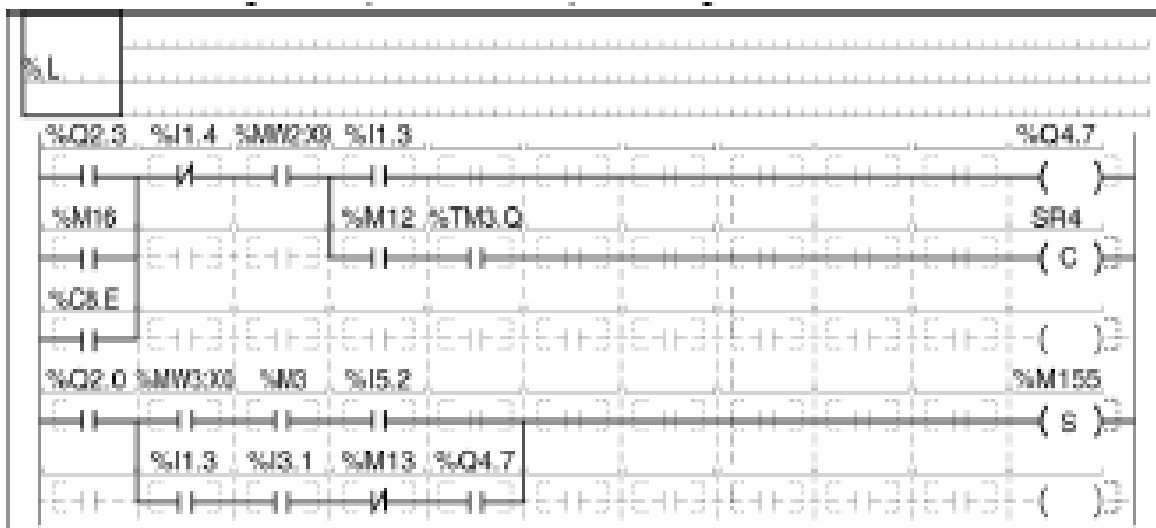
Consente di collegare 2 oggetti tramite svariate connessioni

Regole di programmazione di un rung

La programmazione di un rung avviene tramite elementi grafici, nel rispetto delle regole di programmazione seguenti:

- Gli elementi grafici semplici di test e di azione occupano ciascuno una cella all'interno di un rung.
- Le linee di contatto cominciano sempre sulla linea di potenziale sinistra e terminano obbligatoriamente sulla linea di potenziale destra.
- I test sono sempre situati sulle colonne da 1 a 10.
- Le azioni sono sempre situate sulla colonna 11.
- Il senso di circolazione della corrente è il seguente: per i collegamenti orizzontali, da sinistra a destra, per i collegamenti verticali, nei due sensi.

La schermata seguente riporta un esempio di rung



Regole di programmazione dei blocchi funzione

I blocchi funzione standard si posizionano nella zona di test dei rung. Indipendentemente dal blocco funzione utilizzato, questo deve essere collegato obbligatoriamente in ingresso alla

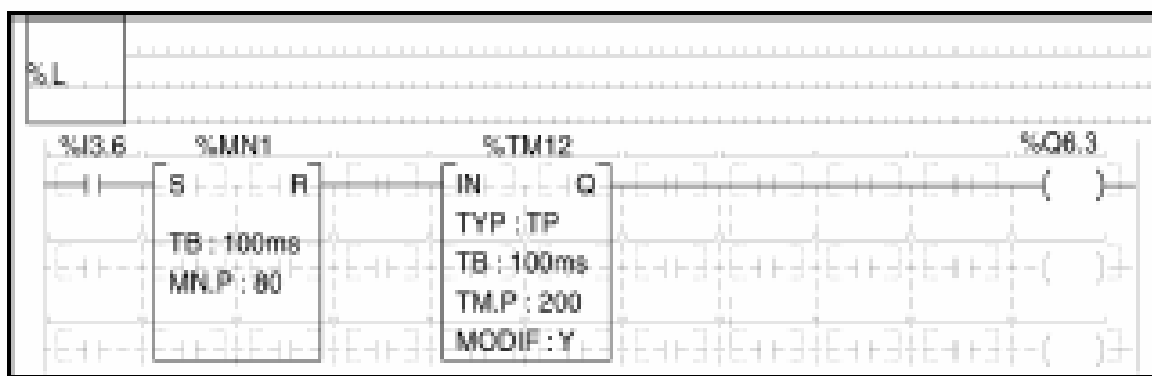
barra di potenziale sinistro, oppure attraverso altri elementi grafici.

Uscite "scoperte": non occorre collegare le uscite dei blocchi funzione ad altri elementi grafici

Uscite verificabili: le uscite dei blocchi funzione sono accessibili all'utente sotto forma di oggetto bit

Le variabili interne dei blocchi e le uscite grafiche sono oggetti utilizzabili a distanza da un'altra parte del programma. Gli ingressi non cablati dei blocchi funzione standard sono posti a 0. Come per gli elementi grafici di tipo contatti, è possibile realizzare delle combinazioni di blocchi funzione.

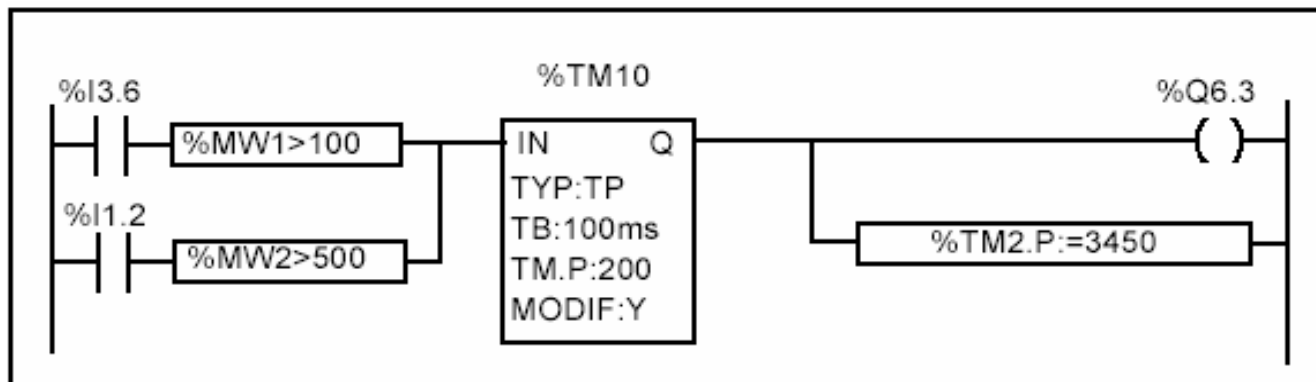
La seguente illustrazione riporta un esempio di rung contenente 2 blocchi funzione.



Regole di programmazione dei blocchi operazione

I blocchi di confronto si posizionano nella zona test, mentre i blocchi operazione si posizionano nella zona azione. Indipendentemente dal blocco operazione utilizzato, questo deve essere collegato obbligatoriamente in ingresso alla barra di potenziale sinistro, oppure attraverso altri elementi grafici. Come per gli elementi grafici di tipo contatti, è possibile realizzare delle combinazioni di blocchi funzione e di blocchi operazione.

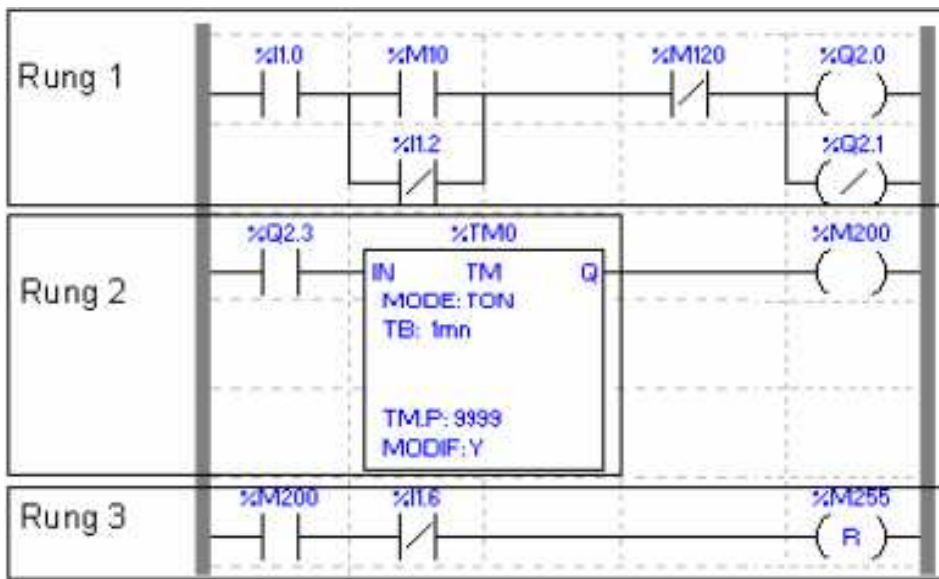
La seguente illustrazione riporta un esempio di rung contenente 2 blocchi di confronto e un blocco operazione.



Rung alimentato

Un rung alimentato contiene elementi grafici collegati tutti tra loro da elementi di collegamento, esclusa la barra di potenziale, ma indipendenti dagli altri elementi grafici del rung (assenza di collegamenti verticali verso l'alto o verso il basso al limite del rung alimentato).

Il rung seguente si compone di tre rung alimentati:



Regola di esecuzione dei rung alimentati

Il primo rung alimentato valutato è quello il cui angolo sinistro si trova nel punto in alto a sinistra. Un rung alimentato è valutato nel senso dell'equazione: valutazione del rung dall'alto in basso, riga per riga, e in ogni riga da sinistra a destra. Se si riscontra un collegamento verticale di convergenza, viene valutata la sottorete ad esso associata (secondo la stessa logica) prima di continuare con la valutazione del rung che la ingloba.

Esecuzione degli elementi in un rung alimentato

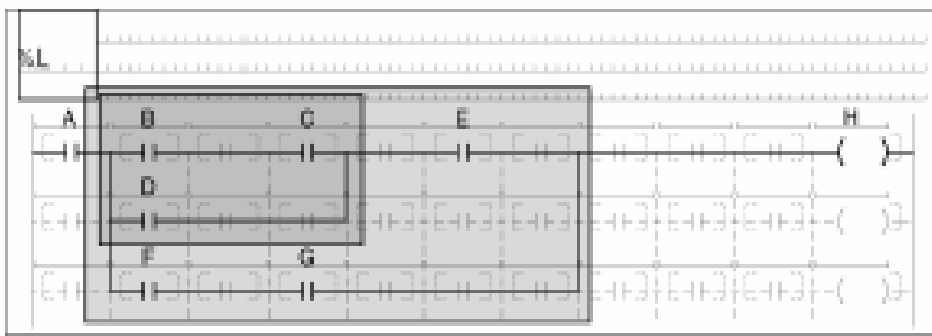
L'ordine di esecuzione degli elementi in un rung alimentato è il seguente:

- 1: il sistema valuta lo stato logico di ciascun contatto in funzione del valore corrente degli oggetti interni dell'applicazione e dello stato degli ingressi dei moduli di I/U acquisiti all'inizio del ciclo

- 2: il sistema esegue i trattamenti associati alle funzioni, ai blocchi funzione e ai sottoprogrammi
- 3: il sistema aggiorna gli oggetti bit associati alle bobine(l'aggiornamento delle uscite dei moduli I/U avviene alla fine del ciclo)
- 4: il sistema viene scollegato verso un altro rung etichettato dello stesso modulo programma(salto ad un altro rung - >> % Li, ritorno al modulo chiamante < RETURN >, oppure arresto del programma< HALT > .

Esempio:

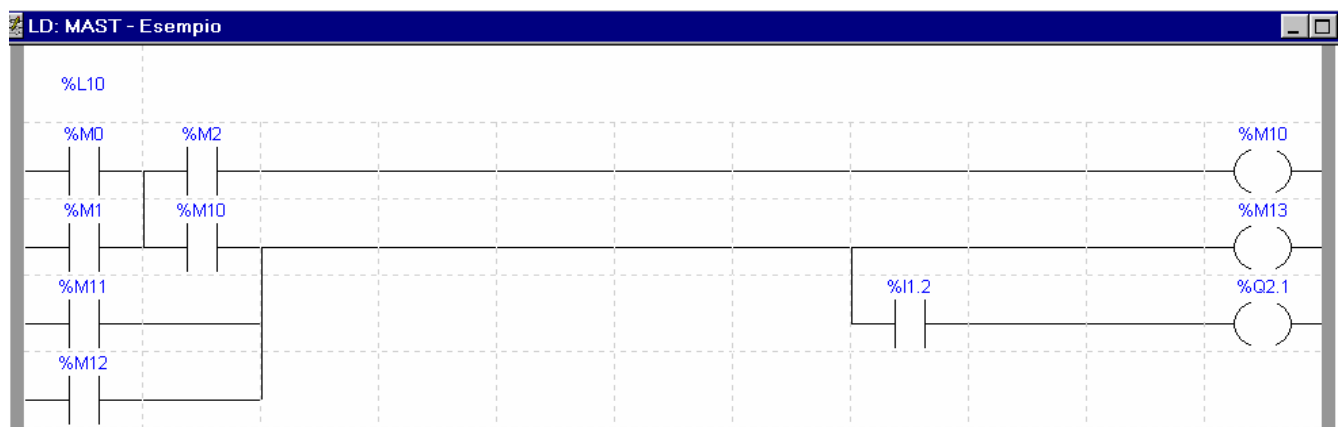
Il seguente disegno visualizza l'ordine di esecuzione degli elementi grafici.



- 1- valutazione del rung fino a quando non si incontra il 1° collegamento verticale di convergenza: contatti A, B, C
- 2- valutazione della prima sottorete: contatto D
- 3- seguito della valutazione del rung fino a quando non si incontra il 2° collegamento verticale di convergenza: contatto E
- 4- valutazione della seconda sottorete: contatti F e G
- 5- valutazione della bobina H

Esempio:

Il seguente disegno visualizza l'ordine di esecuzione degli elementi grafici.



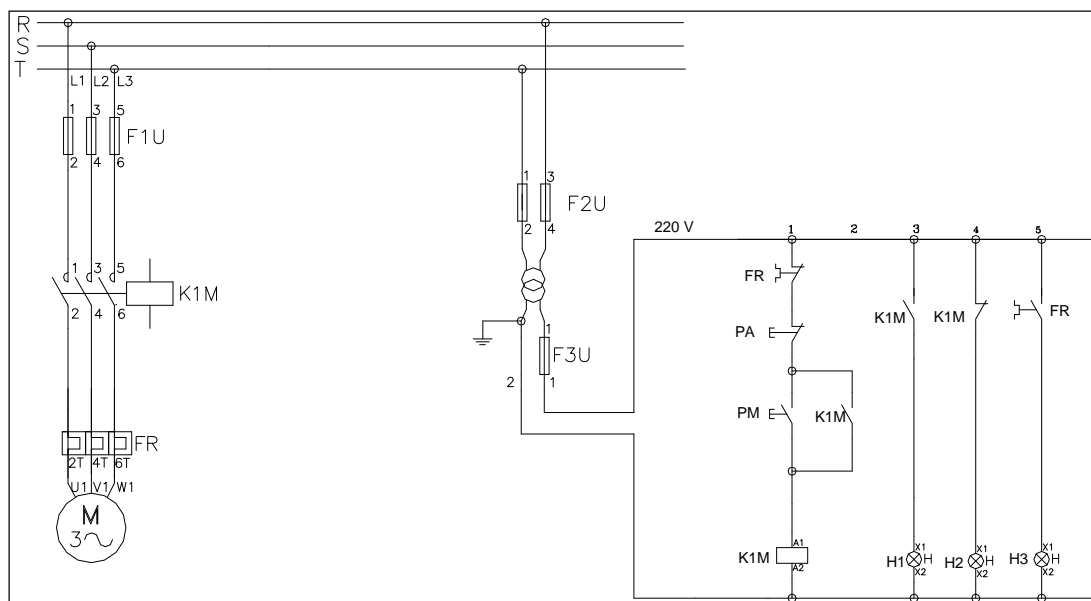
- | | |
|---------------------------|------------------------------------|
| 1- %M0, %M1, %M2, %M10 | aggiornamento prima bobina(%M10) |
| 2- %M10, %M11, %M12, %M13 | aggiornamento seconda bobina(%M13) |
| 3- %I1.2, %Q2.1 | aggiornamento terza bobina(%Q2. |

Fasi per la progettazione di un programma per PLC

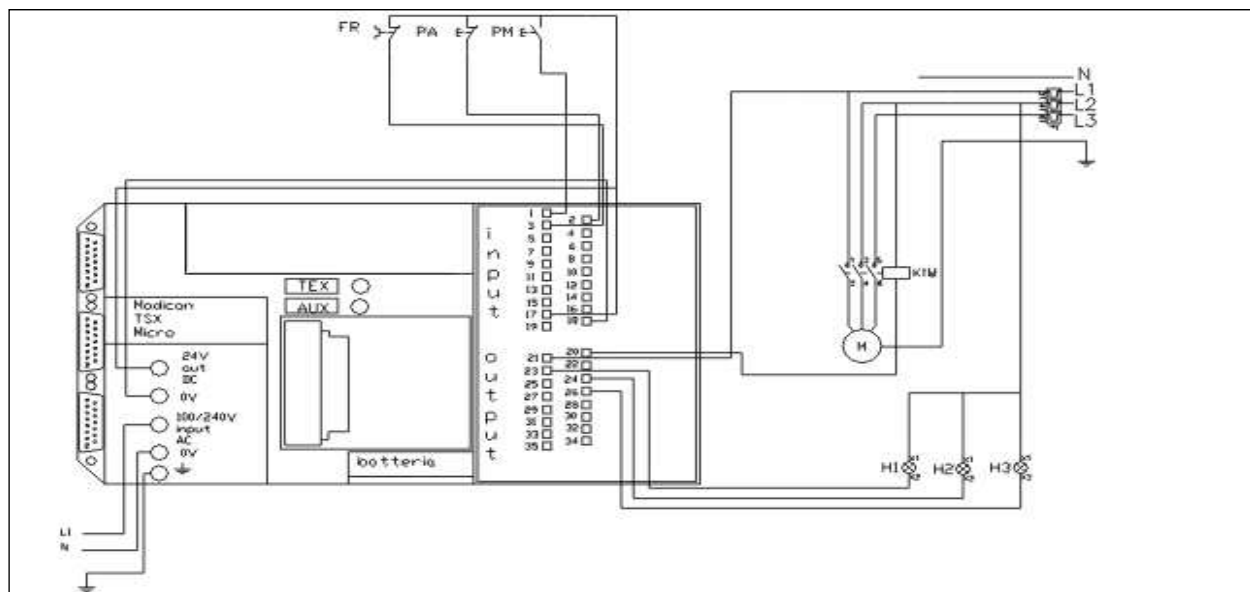
Quando si realizza un impianto con la logica programmabile con PLC, per eseguire la progettazione del programma è consigliabile osservare i seguenti passi:

- 1) Definizione problema, stabilendo ingressi, fasi di elaborazione e uscite.

- 2) Schematizzazione del ciclo di lavoro dell'impianto con uso dello schema elettrico funzionale (o altro: flow chart, Grafcet, schemi logici, ecc., a seconda della preparazione che si possiede).



- 3) Schema collegamenti ingressi / uscite (o schema di montaggio).



N.B.: Il cablaggio degli ingressi/uscite sul PLC non segue quello della logica cablata, rappresentato nello schema funzionale, dove gli ingressi e le uscite sono collegati in cascata. Sul PLC ogni ingresso esterno è alimentato singolarmente perché lo stato logico possa cambiare mediante un contatto; le uscite esterne sono relè, transistor, triac, in modo tale che il circuito sia aperto a uscita non eccitata.

- 4) Compilazione lista ingressi /uscite (lista I/O), con cui si definiscono gli ingressi e le uscite utilizzando una apposita tabella in cui per ogni elemento si riportano:

sigla, descrizione, cablaggio e indirizzo PLC.

SIGLA	DESCRIZIONE	CABLAGGIO	INDIRIZZO PLC
PM	Puls.marcia	NO	% I1.0
PA	Puls.arresto	NC	% I1.1
FR	Relè termico	NC	% I1.2
K1M	Contattore	-	% Q2.0
H1	Segnalazione motore fermo	-	% Q2.1
H2	Segnalazione motore in marcia	-	% Q2.2
H3	Segnalazione intervento termico	-	% Q2.3

L'indirizzo è un codice che identifica in modo univoco ogni elemento funzionale del PLC, cioè ogni istruzione del programma(contatti, relè, lampade di segnalazione, uscite interne, temporizzatori, contatori, ecc.) e lo associa ad una cella di memoria. La lista I/O deve essere compilata prima di scrivere il programma perchè sia chiaro quale indirizzo dare ad ogni istruzione in fase di stesura del programma stesso. Ogni costruttore ha adottato un codice di identificazione, numerico o alfanumerico, delle funzioni disponibili sul PLC(vedi "magazzino funzioni").

- 5) Stesura del programma con uno dei linguaggi previsti(grafico o letterale), ad esempio col linguaggio Ladder, su carta(con matita e gomma, perché nessuno è in grado di scrivere un programma che funziona al primo colpo). Commento del In questa fase è consigliabile aggiungere all'inizio del programma una sintetica descrizione dei compiti che esegue(ciò vale per eventuali subroutine) e delle righe di programma ogni qualvolta si inizia una nuova funzione o si commenta un diverso dispositivo.
- 6) Battitura del programma (con uso del software opportuno) e salvataggio su hard disk e floppy disk.
- 7) Caricamento in memoria RAM del PLC ON LINE: si effettua il collegamento del PC al PLC, si trasferisce il programma da PC a PLC e si mette il PLC in RUN.
- 8) Verifica efficienza del programma. Si effettua la simulazione che può essere fatta utilizzando il software o col simulatore.

- 9) Correzione eventuale, memorizzazione definitiva nel controllore e archiviazione su floppy disk e supporto cartaceo.
- 10) Documentazione: accompagna il progetto, di cui è parte integrante, aiuta il progettista a interpretarlo anche a distanza di tempo. Essa comprende una relazione tecnica con descrizione dell'impianto e riferimenti legislativi e normativi; il dimensionamento di cavi, apparecchiature e protezioni; gli schemi; la lista I/U; il programma; l'elenco del materiale.

Esempio di documentazione tecnica relativa ad un progetto di automazione con l'uso del PLC Modicon TSX Micro Telemecanique.

Impianto di marcia arresto per un motore asincrono trifase con protezione termica e segnalazione controllato da PLC.

Relazione tecnica

Descrizione impianto:

Si vuole realizzare l'avviamento e l'arresto di un motore asincrono mediante l'uso del PLC. Il motore dovrà essere avviato tramite il pulsante di marcia e si dovrà arrestare mediante il pulsante di arresto o per intervento del relè termico.

I pulsanti di marcia e arresto e un contatto NC del relè termico costituiscono gli ingressi del PLC, la bobina di un contattore e tre lampade di segnalazione, che indicano la marcia, l'arresto e l'intervento termico, sono collegate alle uscite del controllore.

L'alimentazione dell'impianto è composta da:

- una linea trifase 400 VAC (per alimentare il motore)
- una linea monofase 24 VAC (per alimentare la bobina del contattore)
- una linea monofase 230 VAC (per alimentare il PLC)

Riferimenti normativi:

- Norma CEI EN 60204-1 (Classificazione CEI 44-5. Pubblicazione IEC 204-1): "Sicurezza del macchinario. Equipaggiamento elettrico delle macchine. Parte I: regole generali".

Questa Norma è utile per la corretta installazione del PLC e di tutto l'equipaggiamento elettrico delle macchine.

- Norma CEI EN 61131-1 IEC 1131-1 CEI 65-23: "Controllori programmabili – Informazioni generali".

Ha la funzione di regolamentare la standardizzazione dei Controllori a Logica Programmabile.

- Norma CEI EN 61131-3 (CEI 65-40, IEC 1131-3): "PLC programmabili - Parte 3: Linguaggi di programmazione"

Specifica la sintassi e la semantica degli elementi software implementati per programmare i PLC.

Dimensionamento:

Per un motore asincrono trifase con $P_n = 0.75 \text{ Kw}$; $I_n = 1,93 \text{ A}$; $V_n = 380 \text{ V}$, dalla tabella di coordinamento degli apparecchi di manovra e protezione dei motori del manuale di elettrotecnica determino:

- F1u: aM $I_n = 4 \text{ A}$ $V = 380 \text{ V}$
- Fr: $I_n = 1.6 - 2.5 \text{ A}$
- Contattore (K1M contattore in AC3 tipo CO 24Vac $I_e = 6 \text{ A}$ $I_{th} = 16 \text{ A}$)

Avendo scelto un trasformatore 380/24V- 50VA:

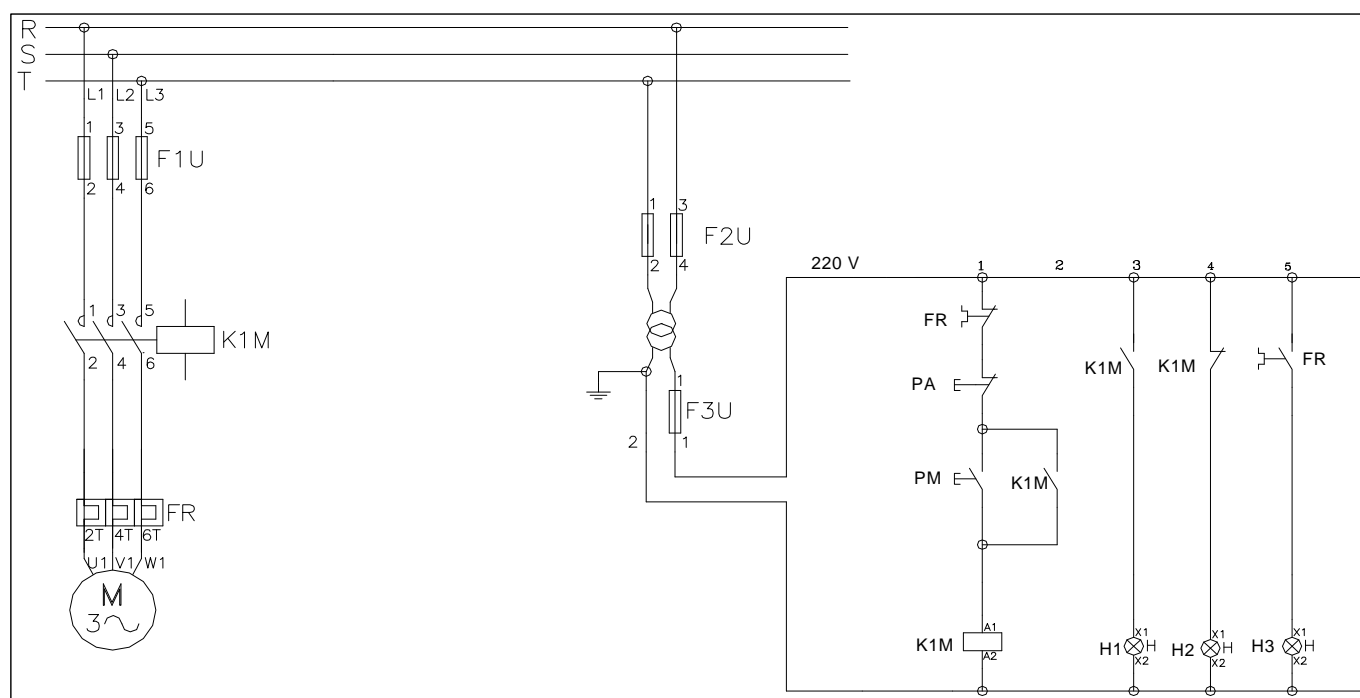
- F2u tipo gL $I_n = 131 \text{ mA}$
- F3u tipo gL $I_n = 2 \text{ A}$

Sezione conduttori circuito di potenza da tab CEI – UNEL 35024: 1.5 mm^2 . In previsione di futuri ampliamenti, per contenere ulteriormente la c.d.t. e poiché non si utilizza molto cavo, per cui il risparmio sarebbe esiguo, si sceglie di impiegare cavi di sezione $2,5 \text{ mm}^2$

Elenco materiale:

- Trasformatore 400/24 VAC
- Trasformatore 400/230VAC
- PLC Tsx-micro Telemecanique
- Contattore (K1M contattore in AC3 tipo CO 24Vac $I_e=6\text{ A}$ $I_{th}=16\text{ A}$)
- Motore asincrono trifase ($P=0.75\text{ Kw}$ $I_n=1,73\text{ A}$ $V=380\text{ V}$)
- Pulsanti di arresto 250V-5A (colore rosso)
- Pulsanti di marcia 250V-5A (colore verde)
- Lampade segnalazione 24V-4W
- Relè termico motore
- Fusibili
- Cavo N07-VK

Schema dei circuiti di potenza comando e segnalazione



Schema dei collegamenti esterni o schema di montaggio:

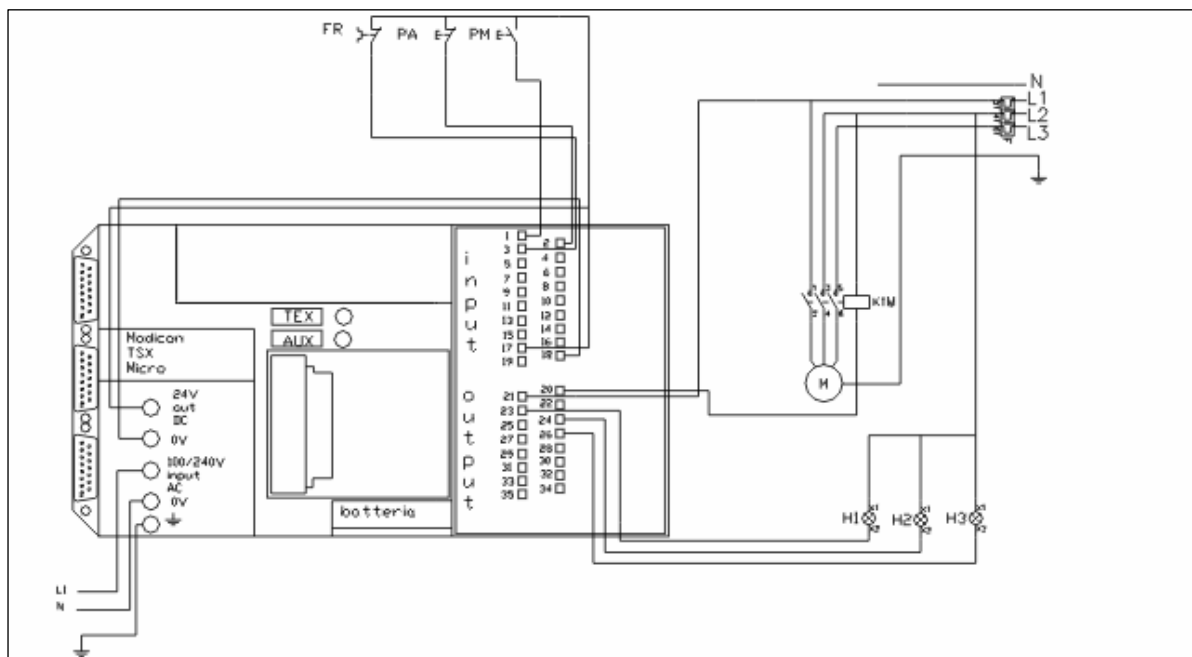
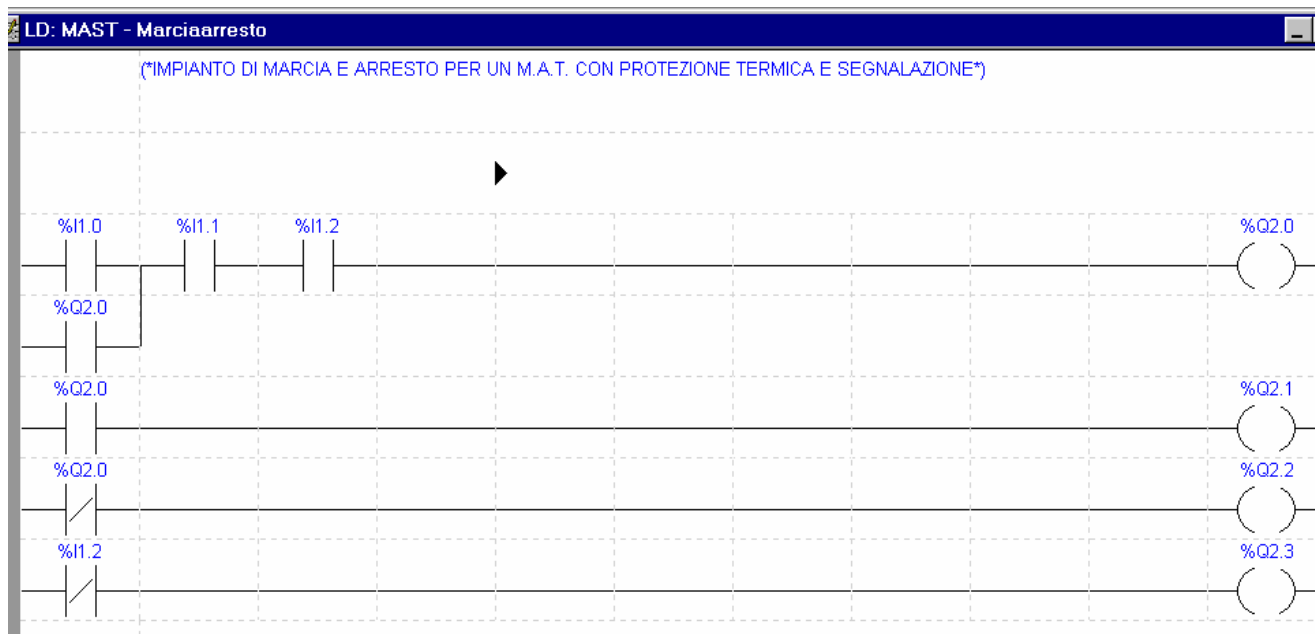


tabella I/U

SIGLA	DESCRIZIONE	CABLAGGIO	IND. PLC
PM	Pulsante marcia	NO	%I1.0
PA	Pulsante arresto	NC	%I1.1
FR	Relè termico	NC	%I1.2
K1M	Contattore	-	%Q2.0
H1	Segnalazione marcia	-	%Q2.1
H2	Segnalazione arresto	-	%Q2.2
H3	Segnalazione intervento relè termico	-	%Q2.3

Schema Ladder: vista variabili



Funzionamento del programma:

Analizzando lo schema ladder si vede che il programma funziona nel seguente modo: premendo il pulsante di marcia chiudiamo il suo contatto NO e mandiamo il segnale 1 al bit associato al contatto di ingresso aperto %I1.0 che si chiude e fa uscire il segnale 1.

Essendo a 1 anche i bit associati ai contatti di ingresso %I1.1 e %I1.2, che rappresentano il pulsante di arresto e il contatto del relè termico, cablati NC, il risultato della zona test è 1 e la bobina %Q2.0 si attiva.

L'attivazione della bobina %Q2.0 provoca la chiusura dell'uscita collegata al contattore K1M e la marcia del motore.

Il cambiamento dello stato logico della bobina %Q2.0 commuta lo stato logico di tutti i suoi contatti, associati a dei bit interni (tutti i contatti con lo stesso indirizzo commutano), col seguente effetto:

- si mantiene il passaggio di corrente nel rung alimentato che collega l'uscita %Q2.0 anche dopo il rilascio del pulsante di marcia e la riapertura dell'ingresso %I1.0 ad esso associato (l'oggetto bit che pilota il contatto di ingresso %I1.0 assume lo stato logico 0 ma essendo in OR col bit interno %Q2.0, che assume lo stato logico 1, il risultato della zona test è ancora 1);
- si interrompe il passaggio di corrente nel rung alimentato che collega l'uscita %Q2.2 (%Q2.2 si disattiva e spegne la lampada H2 di motore fermo, inizialmente accesa);
- si permette il passaggio di corrente nel rung alimentato che collega l'uscita %Q2.1 (%Q2.1 si disattiva e accende la lampada di segnalazione H1 di motore in marcia).

Premendo il pulsante di arresto l'ingresso %I1.0 va a 0 e manda a 0 il risultato della zona test che disattiva l'uscita e arresta il motore, con lo spegnimento della lampada H1 l'accensione della lampada H2.

La marcia del motore si disattiva anche quando, per intervento relè termico, va a 0 l'oggetto bit che pilota l'ingresso %I1.2 e manda a 0 il risultato della zona test; ciò determina, per effetto del bit interno associato al contatto negato %I1.2, nell'ultimo rung alimentato, l'attivazione dell'uscita %Q2.3 e la segnalazione della lampada H3.

Verifica delle conoscenze

Generalità

1. Che cos'è un PLC?
2. Quali sono le differenze tra PC e PLC?
3. Quali sono le principali caratteristiche di un PLC?
4. Che cosa si intende per flessibilità di un PLC?
5. Qual è la differenza tra la logica cablata e la logica programmabile con cui può essere realizzato un automatismo?
6. Come si classificano i PLC?
7. Quali sono i criteri per scegliere un PLC?

Normativa

8. Quali sono le norme tecniche più importanti che si applicano ai PLC?

Hardware

9. Da quali componenti è costituito l'hardware di un PLC?
10. Quali sono le funzioni della CPU?
11. Che cos'è la scansione?
12. A che cosa servono le unità di I/U? Da quali dispositivi sono costituite?

Software

13. Che cos'è un linguaggio di programmazione?
14. Come si classificano i linguaggi di programmazione?
15. Com'è strutturata una pagina di programma in linguaggio ladder?
16. In cosa consiste la configurazione di una applicazione?
17. Che cos'è l'indirizzo PLC?
18. Descrivere il significato dei seguenti indirizzi: %I3.6; %Q4.2.
19. Dire se è corretta, motivando, la sintassi dei seguenti indirizzi: %I2.0; %Q1:15.