CORSO PLC

Automazione

“Complesso di tecniche dirette ad azionare macchine con altre macchine”

Dalla “Meccanizzazione” ossia dall’impiego di macchine azionate dall’uomo , si passa all’azionamento automatico delle macchine, senza l’intervento dell’uomo

L’Automazione si occupa di tutti i problemi inerenti il controllo attivo di un processo

Per processo s’intende l’evoluzione nel tempo di un sistema fisico

Il controllo di processo

Essendo lo scopo principale di un processo l'ottenimento di un prodotto avente determinate caratteristiche, chi lo progetta e chi lo gestisce deve fare in modo che esso possa essere opportunamente controllato.

Possiamo allora definire «CONTROLLO DI PROCESSO» l'interazione di un insieme di mezzi avente lo scopo di far evolvere il processo come previsto in base alle specifiche del prodotto da ottenere.

Tale controllo può essere effettuato secondo diversi livelli gerarchici quali:

- **singola macchina,**
- **insieme di macchine,**
- **reparto di produzione,**
- **insieme di reparti.**

Sia il processo che il controllo di processo possono essere effettuati in due modi profondamente diversi:

- **Manuale**
- **Automatico**

Soprattutto in questi ultimi anni le industrie hanno privilegiato il ricorso a processi automatici nei quali un numero sempre maggiore di operazioni viene effettuato da macchine che sostituiscono il lavoro umano.

PER «AUTOMAZIONE» SI DEVE PERTANTO INTENDERE L'INTERAZIONE DI UN INSIEME DI MEZZI CHE CONSENTE DI EVITARE ATTIVITÀ MANUALI SIA PER QUANTO RIGUARDA LE SINGOLE FASI CHE PER QUANTO RIGUARDA IL CONTROLLO DEI PROCESSI.

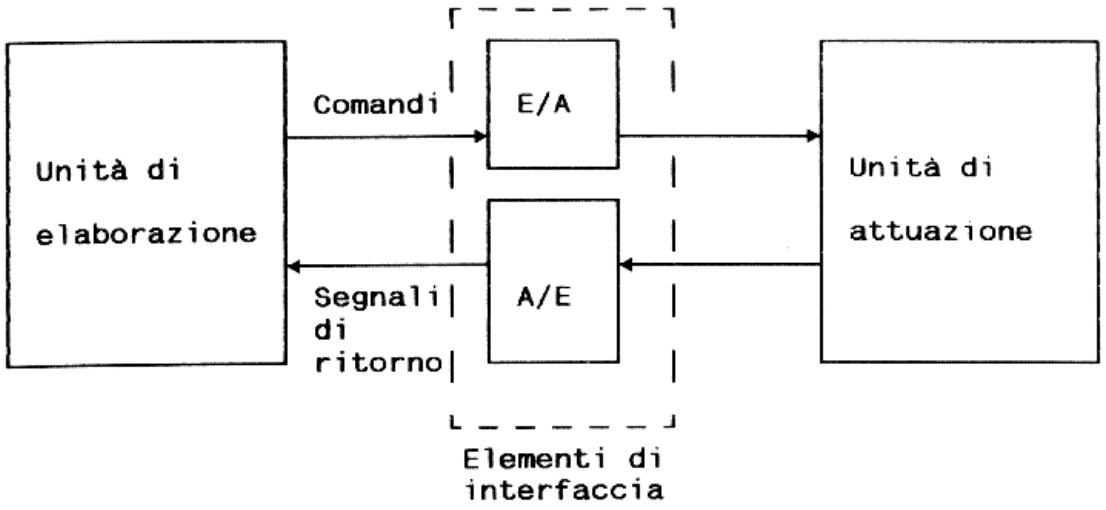
Comando automatico

L’automazione di processo viene realizzata con un insieme di apparecchiature cui si dà il nome di «sistema di comando» o «comando automatico».

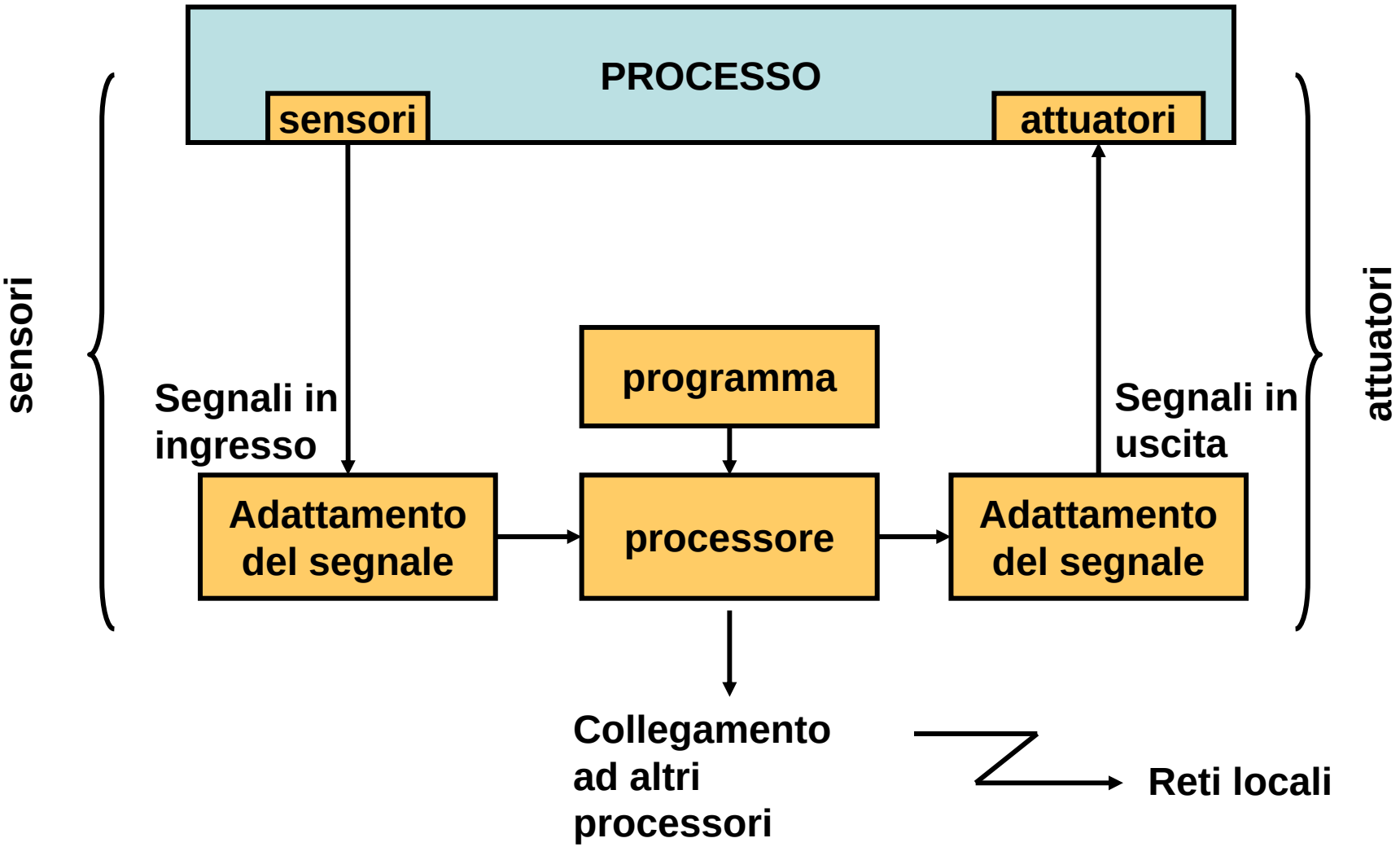
Nella prima fase si ha un prelievo di informazioni sull’andamento del processo, grazie ad opportuni dispositivi genericamente indicati con il nome di «sensori».

La fase successiva è quella che avviene nel cervello del sistema di comando e consiste nell’elaborazione delle informazioni ricevute (calcoli matematici, calcoli logici, decisione delle azioni da comandare).

La terza fase consiste nell’invio dei risultati dell’elaborazione agli organi attuatori i quali eseguono, a conclusione del ciclo, le istruzioni ricevute.



Elementi del sistema di controllo di un processo



Comando

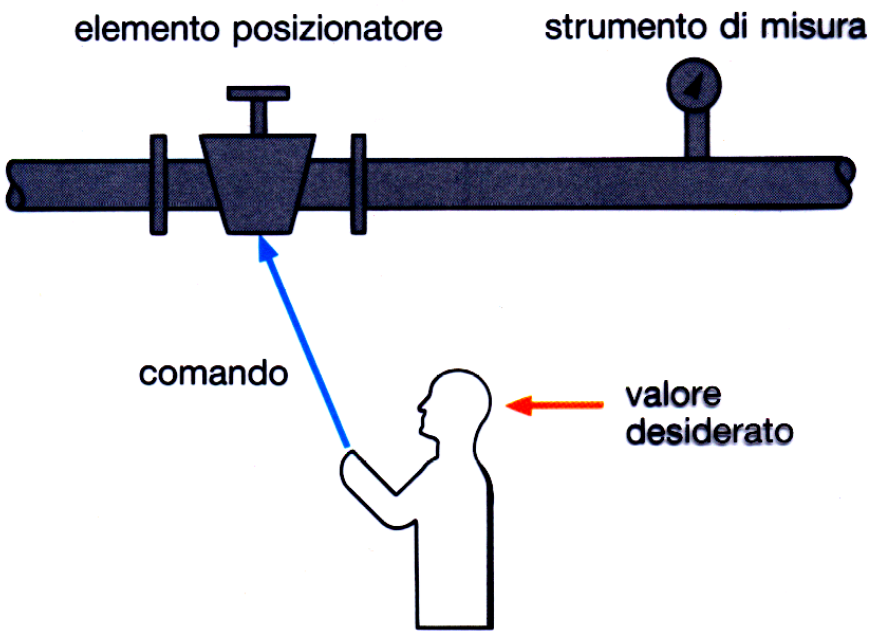
Il comando è ciò che avviene in un sistema nel quale una o più grandezze di ingresso influenzano altre grandezze d'uscita in base alle leggi fisiche intrinseche del sistema considerato

La caratteristica del comando è il fatto che lo svolgimento dell'azione, cioè la catena di comando è aperta

Comando manuale di una portata

La quantità di una determinata sostanza che nell'unità di tempo fluisce attraverso una tubazione (portata) può essere variata mediante una valvola (elemento posizionatore) e rilevata con uno strumento di misura

Per una data posizione della valvola la portata può comunque variare a causa di disturbi (variazioni di pressione, prelevamenti irregolari) senza che ciò venga rilevato

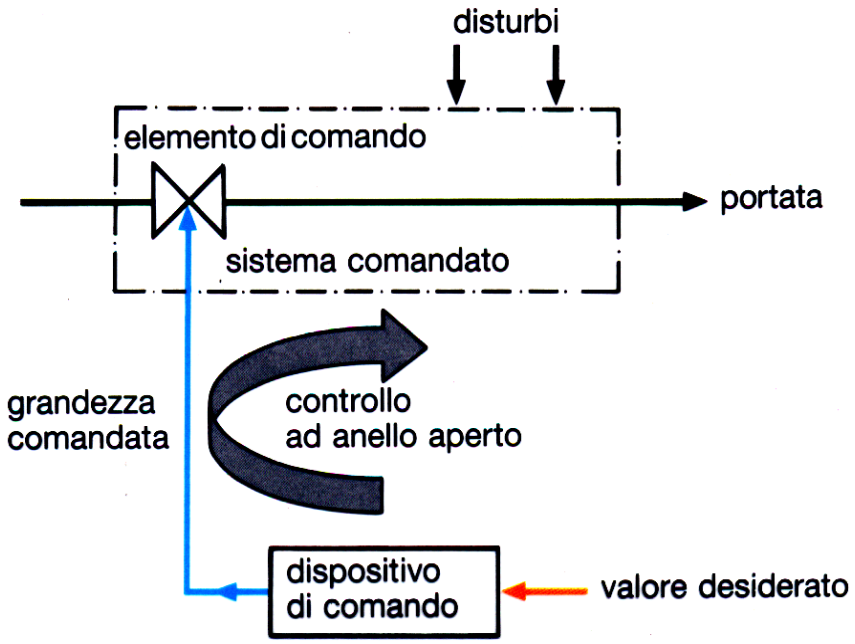


Sistema di comando per il controllo della portata ad anello aperto

Se il valore effettivo (portata istantanea) ed il valore desiderato (valore di riferimento) sono diversi l'uno dall'altro, il sistema non può correggersi automaticamente.

Una correzione può avvenire solo tramite intervento dall'esterno, mediante riposizionamento della valvola.

Per questo motivo si parla di un circuito ad anello aperto, come rappresentato in figura.



Regolazione

La regolazione è un processo durante il quale la grandezza da controllare viene costantemente rilevata, confrontata con un’altra grandezza, quella di riferimento, e, in base all’esito di questo confronto, modificata fino a raggiungere il valore della grandezza di riferimento.

Lo svolgimento delle azioni che ne derivano viene effettuato in un circuito chiuso chiamato anello di regolazione.

Regolazione manuale di una portata

Se a causa di disturbi si producono indesiderate variazioni del valore registrato in uscita, questo valore effettivo deve essere regolato, ed i disturbi devono, se necessario, essere eliminati mediante compensazione della valvola.

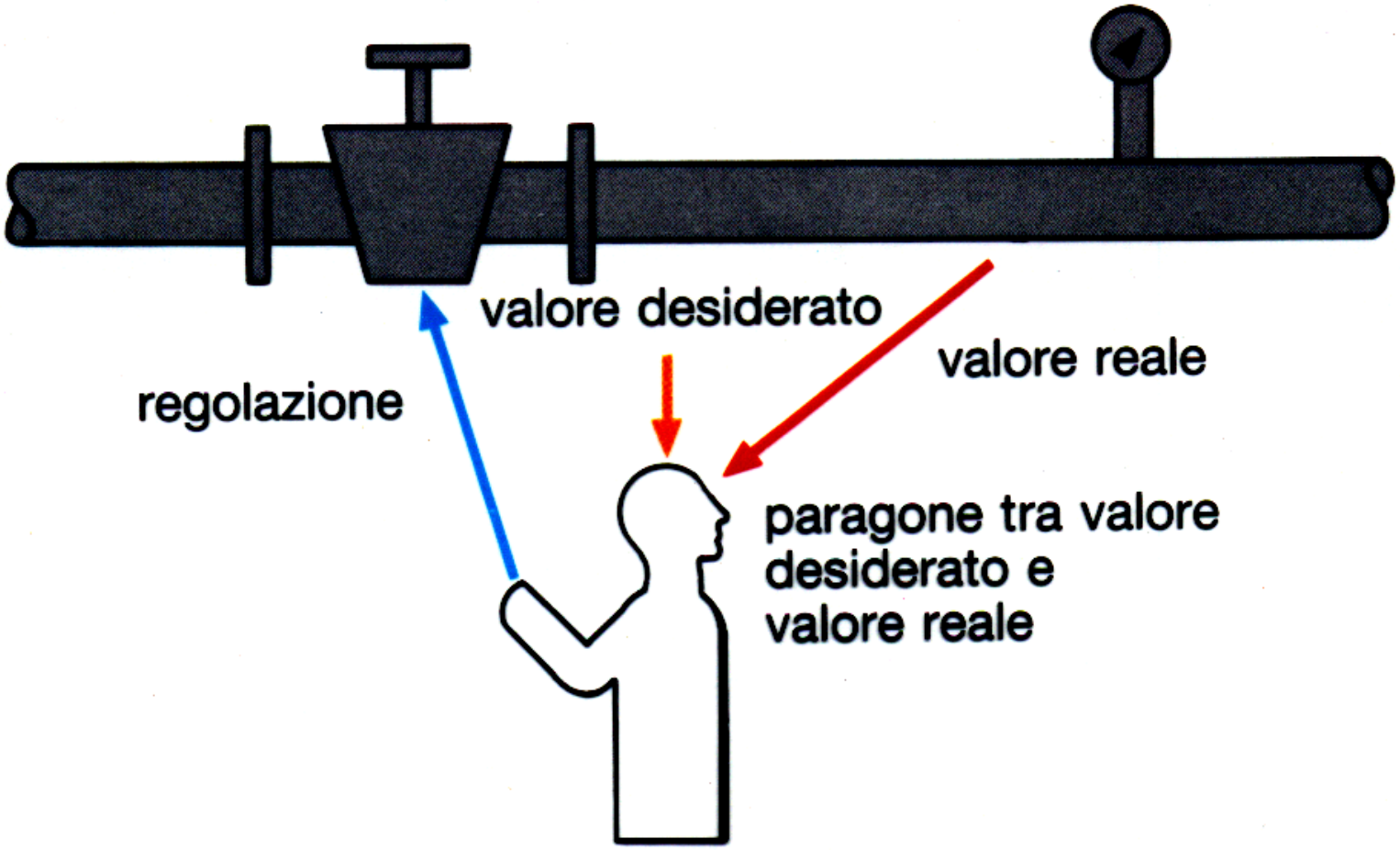
In caso di differenza tra valore effettivo e valore desiderato diventa indispensabile una regolazione correttiva.

In questo modo si ottiene la regolazione del processo dove l'anello di regolazione risulta essere chiuso.

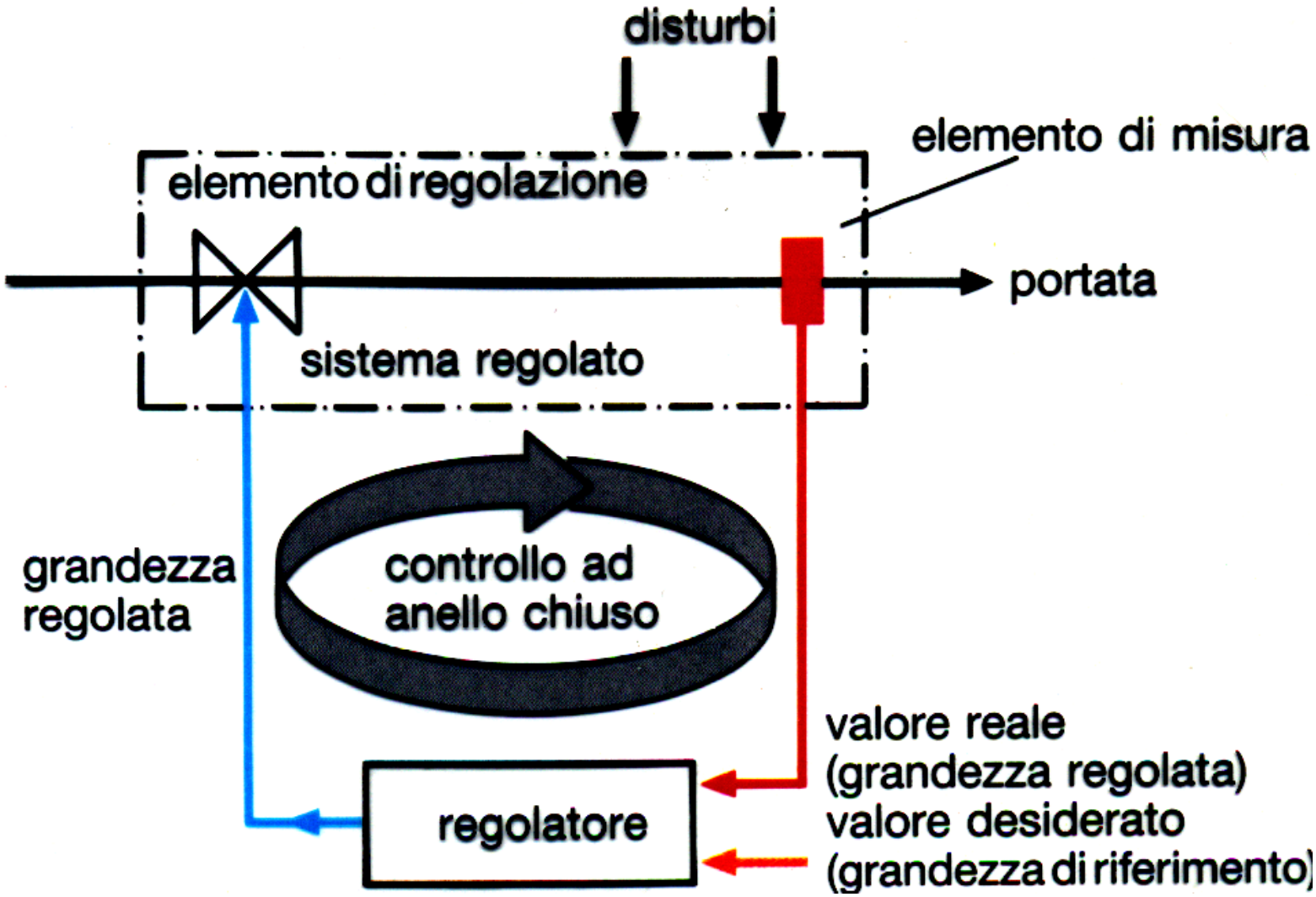
Se le attività di misurazione, confronto e compensazione vengono svolte dall'uomo, si tratta di una regolazione manuale.

Se invece vengono svolte da un dispositivo, si parla di regolazione automatica.

Regolazione manuale di una portata



Sistema di regolazione per il controllo ad anello chiuso della portata



Avvio dell’automazione

La procedura tipica di automazione richiedeva:

- La costruzione di un pannello
- Il cablaggio dei componenti elettromeccanici e dei moduli elettronici scelti
- La verifica del funzionamento.

Non essendo generalmente disponibili delle strutture di simulazione, il controllo del funzionamento doveva basarsi sull'effettiva situazione impiantistica, collegando fisicamente il controllore al sistema da governare.

Se l'architettura di controllo si rivelava inefficace, spesso occorreva modificare l'intero schema di cablaggio, riassemblando i componenti utilizzati e collaudare nuovamente il tutto.

Se dopo un certo tempo il ciclo di funzionamento dell'impianto doveva subire dei cambiamenti (per motivi di produzione) era necessario effettuare delle modifiche al quadro comando, non sempre facili e veloci o addirittura rifare completamente il quadro stesso con un costo non indifferente:

- In pratica, per ogni macchina o per ogni tipo di lavorazione, occorreva un quadro costruito su misura.

Si parla di circuiti a “Logica Cablata” ……

Circuiti a logica cablata

I circuiti a logica cablata sono quindi:

- Rigidi (cioè non possono essere facilmente modificati)
- Scarsamente riutilizzabili.

La caratteristica saliente di un sistema a logica cablata è data dall'elaborazione parallela dei segnali:

- il verificarsi di una qualsiasi condizione di ingresso determina l'immediata modifica della corrispondente configurazione dei segnali di uscita.

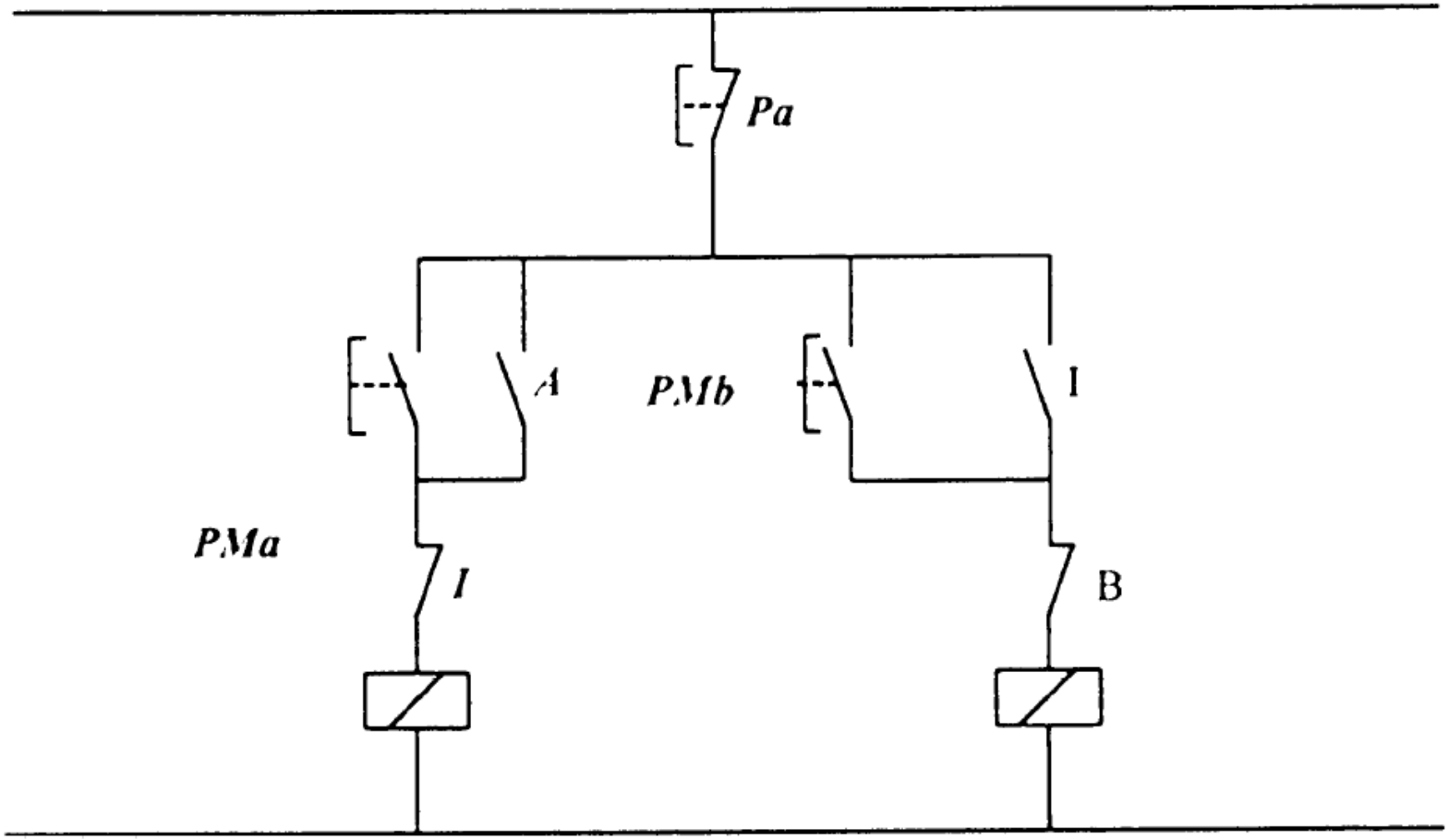
LOGICA CABLATA

Un circuito che svolge determinate sequenze viene detto LOGICO. Un circuito logico non è altro che un insieme di elementi che svolgono una determinata funzione.

In un circuito a LOGICA CABLATA i vari componenti vengono collegati tra loro seguendo uno schema ben preciso.

I componenti di un sistema a logica cablata possono essere di varia natura, in base alla tecnologia adottata per il comando di quel dato sistema o impianto, possono essere componenti pneumatici oppure elettromeccanici, oppure ancora componenti elettronici (come circuiti integrati).

Logica cablata con componenti elettromeccanici



Verso il PLC

In una visione moderna del mondo del controllo, l'elettronica rappresenta il cammino verso la futura automazione.

La fabbrica del futuro è oggi possibile , essa può dare:

- Una maggiore affidabilità del sistema;
- Una maggiore qualità del prodotto;
- Un maggior flusso di informazioni;
- Un costo più ridotto;
- Un maggior grado di efficienza e flessibilità.

Uno degli elementi fondamentali di una fabbrica così concepita è un'apparecchiatura elettronica chiamata:

Controllore a **L**ogica **P**rogrammabile ➔ **PLC**

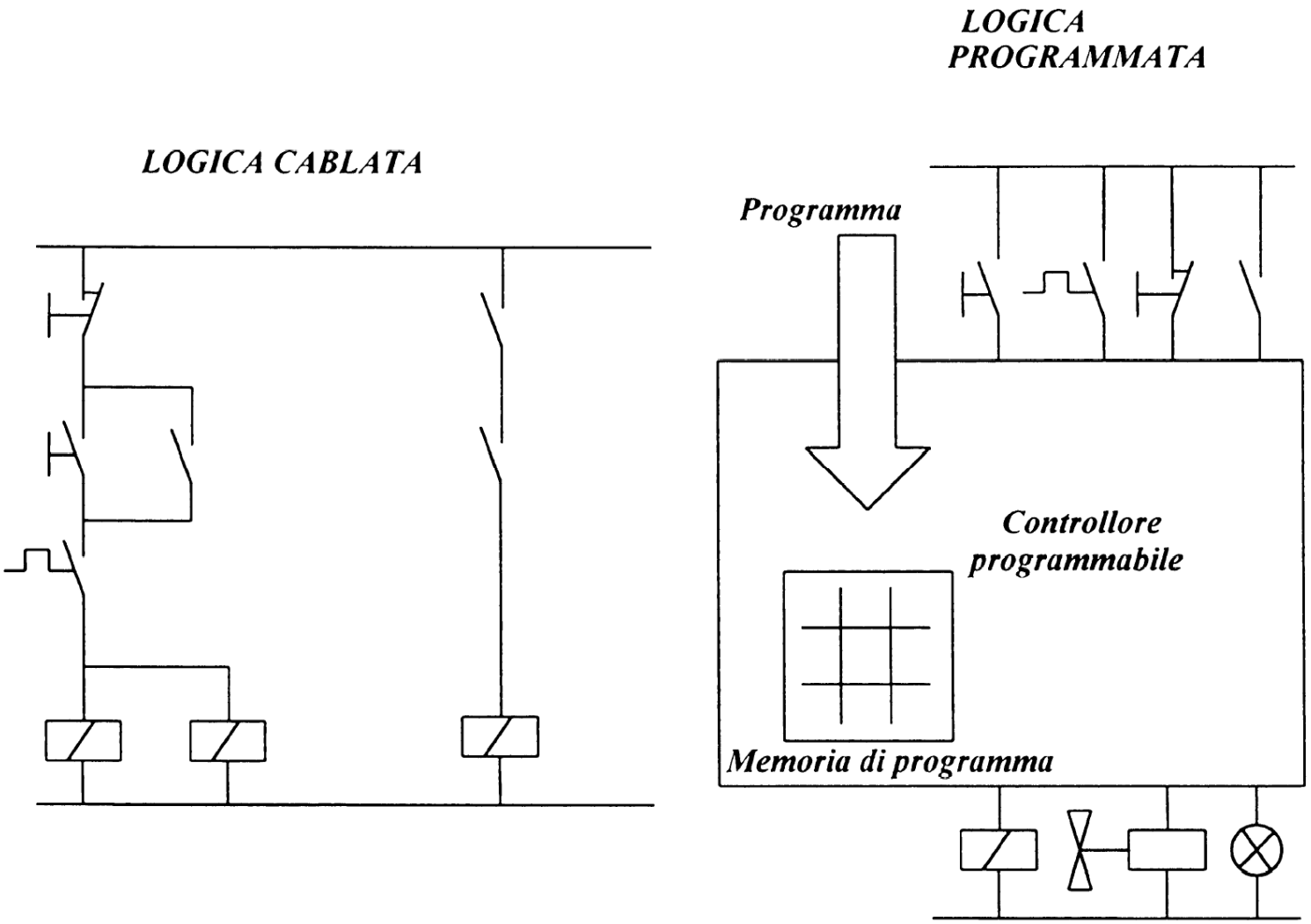
L’introduzione del PLC

Questa apparecchiatura fu introdotta sul mercato per la prima volta nel 1970 ed è stata da allora perfezionata grazie alla disponibilità di componenti elettronici sempre più sofisticati, come il microprocessore.

I PLC oggi vengono progettati utilizzando le tecniche più evolute (basate sui microprocessori e sui circuiti elettronici integrati) e consentono un funzionamento molto affidabile in applicazioni industriali, dove esistono parecchi fattori di rischio come disturbi elettrici, alte temperature, alimentazioni AC poco affidabili, vibrazioni meccaniche.

Il PLC può essere definito come un sistema digitale elettronico destinato ad operare in ambiente industriale, che utilizza una memoria programmabile per memorizzare informazioni e istruzioni atte a implementare specifiche funzioni, come logica combinatoria, controlli sequenziali, temporizzazioni, conteggi, calcoli aritmetici, con riferimento al controllo di macchine e processi.

Dalla logica cablata alla logica programmata



LOGICA PROGRAMMATA

In un circuito a logica programmata le varie funzioni (relè porte logiche, flip-flop, temporizzatori, contatori, ecc.) vengono ottenute mediante opportuni circuiti elettronici integrati, attivati con determinate istruzioni scritte e conservate nella memoria del sistema.

Con l'introduzione del controllore logico programmabile per gestire automaticamente impianti o macchine occorre

- stabilire la sequenza e le condizioni di attivazione delle varie funzioni;
- scrivere nella memoria le istruzioni atte ad attivare le funzioni richieste al verificarsi delle condizioni stabilite (programma).

Risulta pertanto evidente che, un intero ciclo produttivo o un controllo di processo può essere modificato a piacere anche in fase operativa semplicemente modificando il contenuto della memoria (cioè il programma). Inoltre non esiste più un vero e proprio cablaggio in quanto le varie funzioni vengono eseguite tramite circuiti elettronici, interni al PLC, opportunamente predisposti.

I collegamenti al PLC

Agli ingressi del PLC occorre però sempre collegare i pulsanti, gli impostatori, i finecorsa e i rilevatori di vario tipo per segnalare:

- in quale stato deve essere portato l'impianto (pulsanti, impostatori);
- in quale stato si trova istante per istante l'impianto (fine corsa, rivelatori).

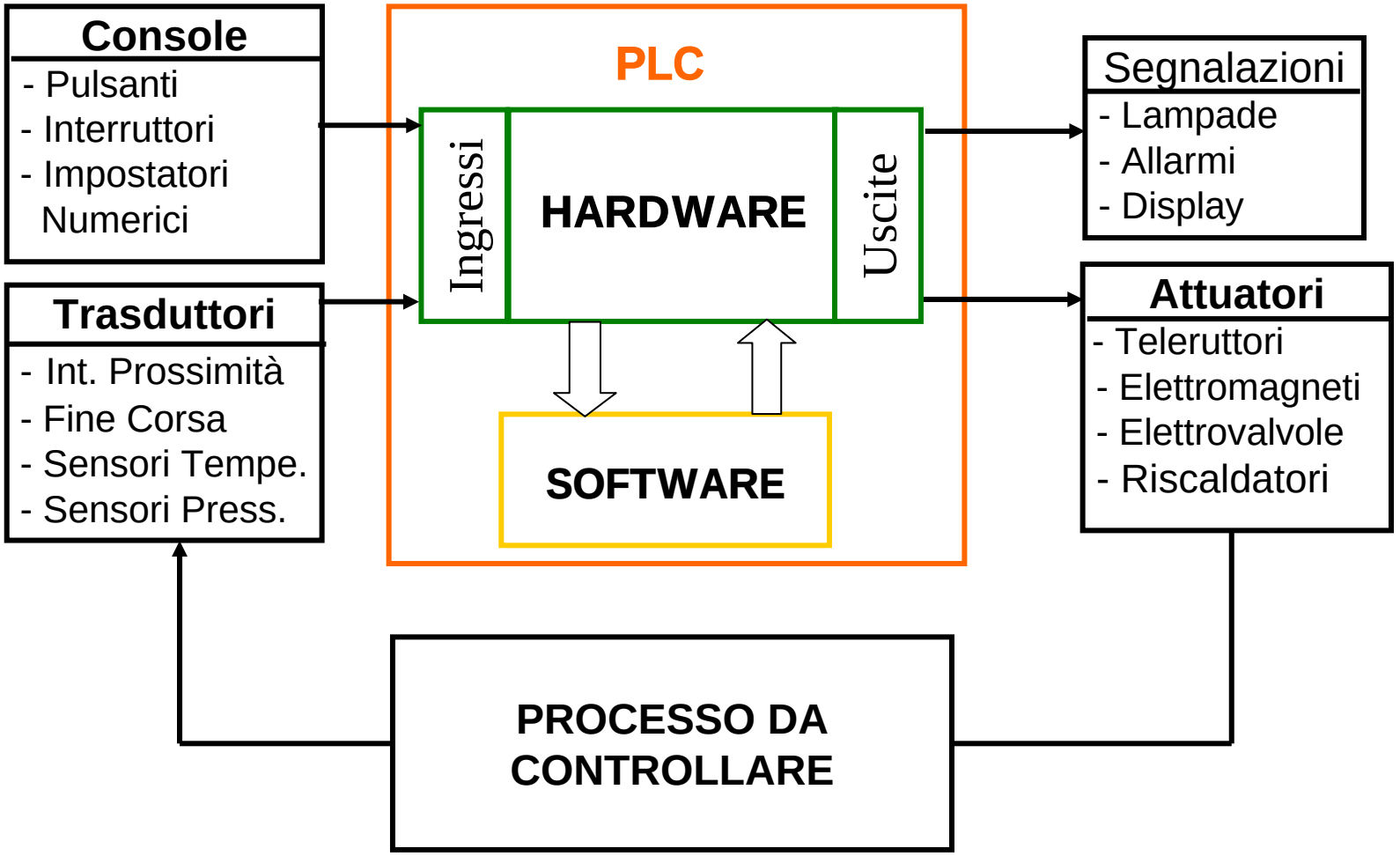
Alle uscite del PLC vanno invece collegati:

- i contattori per il comando di motori elettrici;
- I circuiti di comando degli attuatori (es. le elettrovalvole per il comando dei cilindri);
- i dispositivi di segnalazione;
- dispositivi vari per il comando degli attuatori.

Le istruzioni eseguibili da un PLC dipendono dall'"intelligenza" della sua unità centrale o CPU. Nei casi più semplici un PLC può limitarsi a eseguire tutte le funzioni di tipo logico altrimenti realizzabili tramite un sistema cablato a relè, oppure un sistema di moduli elettronici tra loro collegati in modo da dare funzioni fisse e definite.

Schema funzionale del PLC

Controllo di Processo



Classificazione dei PLC (I/O)

Con riferimento alla quantità di ingressi ed uscite che gestisce, un PLC si dice:

- di gamma (o taglia) bassa quando può controllare un numero massimo di 64 I/O
- di gamma media quando controlla un numero di I/O compreso tra 64 e 512
- di gamma alta quando controlla più di 512 I/O (normalmente 1024 o 2048)

Classificazione dei PLC (architetturale)

In base al criterio costruttivo adottato i PLC possono essere:

- monoblocco (o compatti)
- Modulari

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

- Sequenziali
- multifunzione

PLC Monoblocco

Si dicono monoblocco (o compatti) quando vengono offerti in una configurazione rigida che non può essere modificata (Siemens Simatic S5 modello 101U: PLC monoblocco)

In taluni casi il numero degli I/O può essere aumentato con il collegamento ad una unità d'espansione, anch'essa di tipo rigido e generalmente uguale, sia nella forma che nelle prestazioni, all'unità base.

I PLC compatti sono generalmente di gamma bassa.

PLC Modulare

Un PLC si dice modulare quando è configurabile dall'utente, a secondo delle specifiche esigenze assemblando in un rack, o su una base, varie schede di concezione modulare aventi ciascuna una determinata funzione.

(Siemens Simatic S5 modello 105R PLC modulare con alimentatore montato e con 9 slot liberi di cui uno per CPU e 8 per schede I/O)

Classificazione dei PLC (Funzionale)

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

- Sequenziali
- Multifunzione

PLC Sequenziali

I PLC sequenziali possono essere compatti o modulari, di taglia piccola o media. Sono impiegati nella realizzazione degli automatismi che funzionano secondo una logica sequenziale; in pratica questi sono i controllori della prima generazione ovvero quelli nati per sostituire i quadri elettromeccanici.

Gli attuali PLC sequenziali sono migliorati molto e svolgono, oltre a quelle logiche, anche altre funzioni come ad esempio:

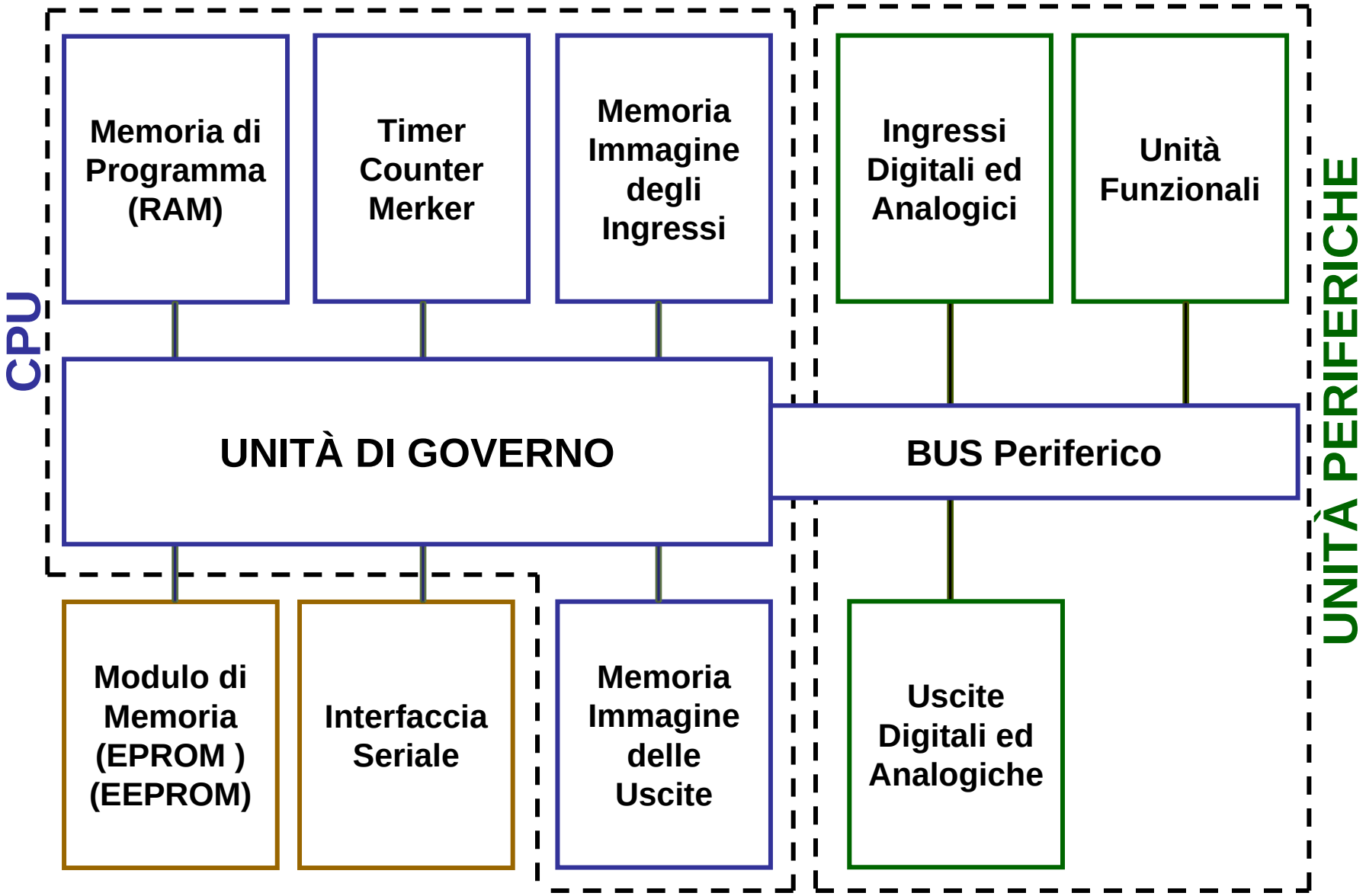
- calcoli matematici
- elaborazione di segnali analogici oltre che On-Off
- conteggio veloce

PLC Multifunzione

I PLC multifunzione sono impiegati in tutti quei casi in cui, oltre alle funzioni caratteristiche della logica sequenziale, sono richieste alcune delle seguenti prestazioni:

- misura
- regolazione (PID)
- posizionamento, controllo assi
- dialogo tra PLC e periferiche
- comunicazione tra PLC

Schema generale di un sistema PLC



Hardware di un PLC

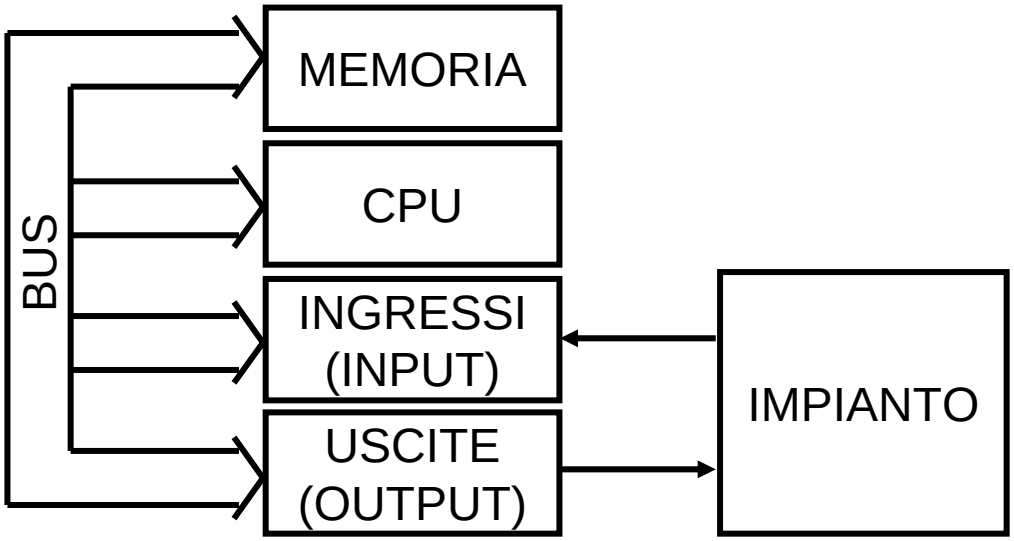
Memoria: Supporto fisico dove sono registrate le istruzioni che costituiscono il Programma applicativo e i dati necessari per le funzioni ausiliarie

CPU: Legge lo stato dei segnali di ingresso ed esegue le istruzioni (in sequenza) registrate in memoria. In base all'elaborazione provvede ad aggiornare lo stato delle uscite.

Ingressi: Riceve i segnali (Fine corsa, Pulsanti, etc.)

Uscite: Riceve i segnali prodotti dall'elaborazione e li adatta per comandare i vari organi attuatori (Elettrovalvole, Teleruttori, spie luminose, etc.)

Bus: Via di comunicazione e transito tra i vari blocchi



Unità Centrale

L’Unità Centrale di un PLC risulta costituita dalle seguenti parti fondamentali:

- scheda processore (CPU)
- memorie
- alimentatore

Scheda Processore (CPU)

La scheda processore, denominata anche CPU (Central Processor Unit) o ALU (Arithmetic-Logic-Unit) è la parte più importante del PLC poiché, servendosi di un microprocessore, consente l'organizzazione e il coordinamento di tutte le attività del comando.

La CPU legge i segnali d'ingresso che recano le informazioni provenienti dal campo e che sono rilevate per mezzo di pulsanti, finecorsa, sensori e trasduttori vari.

Qualora ci siano delle variazioni in tali segnali, la CPU reagisce, elabora tali nuovi dati secondo la logica interna programmata dall'utente e genera gli opportuni segnali d'uscita.

Questi ultimi pilotano i dispositivi di campo che comandano gli attuatori permettendo l'effettuazione dei movimenti secondo la sequenza desiderata.

Funzionamento della CPU

La CPU esegue le varie istruzioni del programma in memoria:

- sequenzialmente (cioè una alla volta e una dopo l'altra);
- in modo ciclico (cioè terminata l'ultima istruzione riparte dalla prima)

A causa di questo tipo di funzionamento è praticamente impossibile una rilevazione contemporanea di tutti i segnali di ingresso.

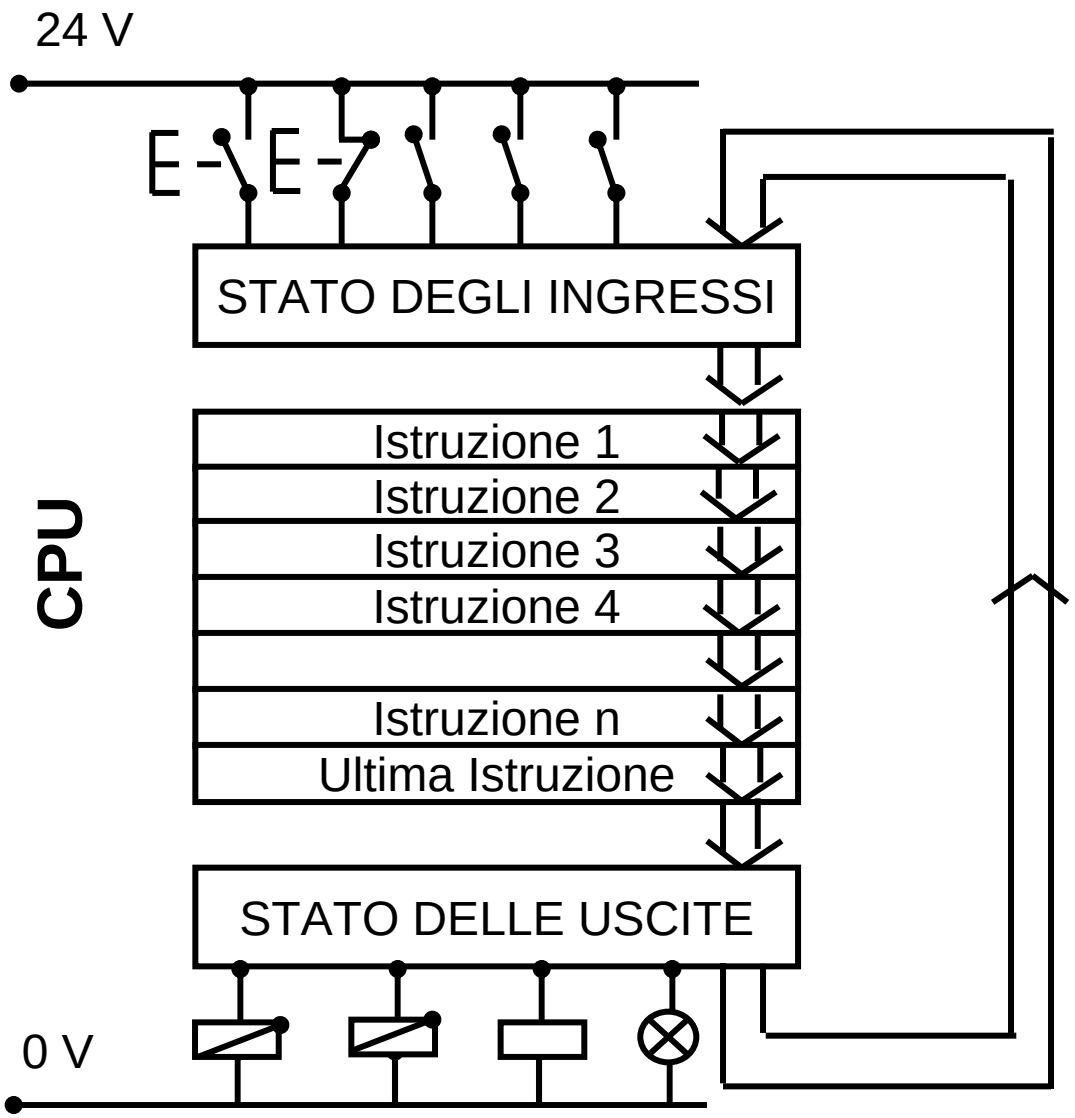
Tra l'elaborazione di una riga di programma e la successiva intercorre necessariamente un tempo, che, pur breve, è sensibile e comunque diverso da zero.

Questa caratteristica è alla base dei metodi di acquisizione dello stato degli ingressi da parte dei PLC.

Alcuni realizzano quella che viene definita un'immagine degli ingressi all'inizio della elaborazione. Tale immagine è conservata inalterata per tutto un ciclo di elaborazione, come se effettivamente gli ingressi non cambiassero, a differenza della realtà fisica, introducendo un errore noto e di solito non problematico.

In altri casi i PLC aggiornano periodicamente l'immagine nel corso dell'elaborazione

Funzionamento della CPU

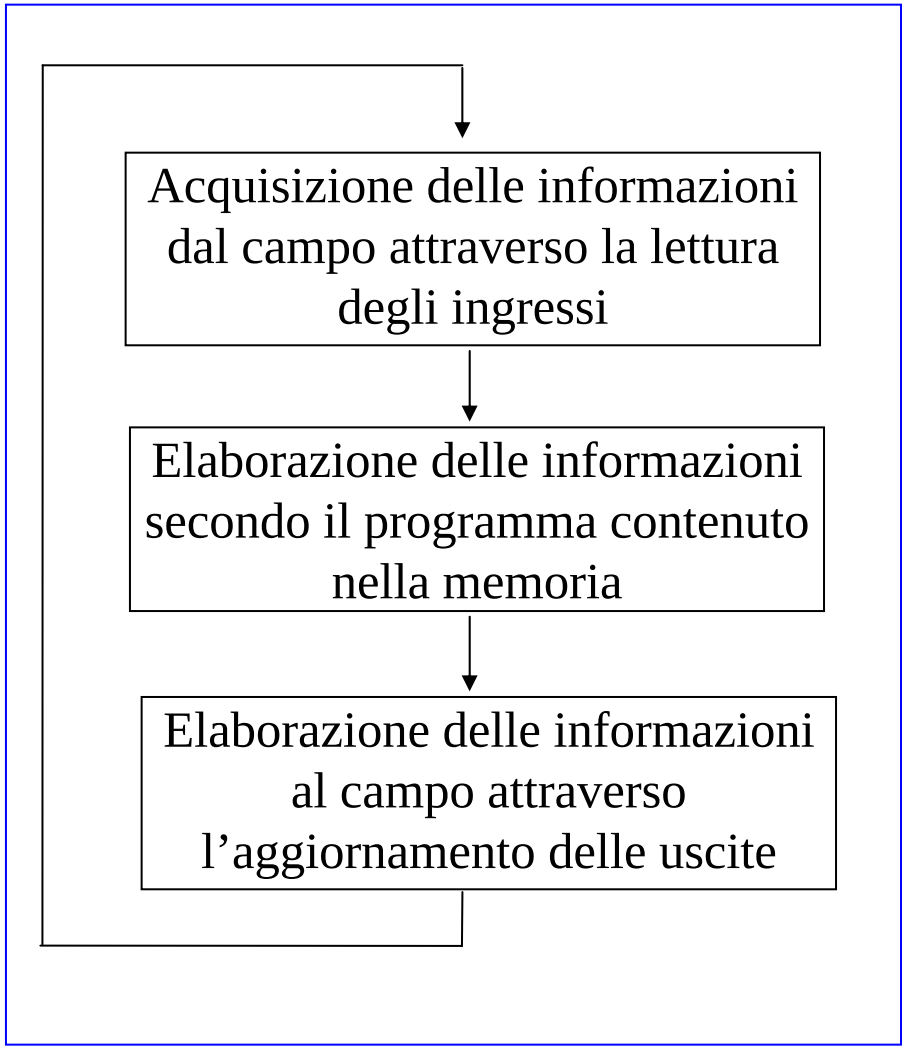


La CPU legge lo stato dei segnali di Ingresso provenienti dalla Macchina o dall'Impianto da controllare, esegue in sequenza le istruzioni registrate in memorie e, in base ai risultati logici dell'elaborazione, provvede ad aggiornare lo stato delle Uscite del Sistema.

Scansione

La CPU svolge le proprie mansioni secondo un ordine sequenziale, a cui si dà il nome di “Scansione”, che può essere schematizzato come in figura

- Acquisizione dagli ingressi
- Esecuzione del Programma
- Generazione uscite



Tipi di Scansione

Ciascun PLC è costruito per operare secondo uno dei seguenti tipi di scansione:

- sincrona d’ingresso e di uscita
- sincrona d’ingresso e asincrona d’uscita
- asincrona d’ingresso e d’uscita

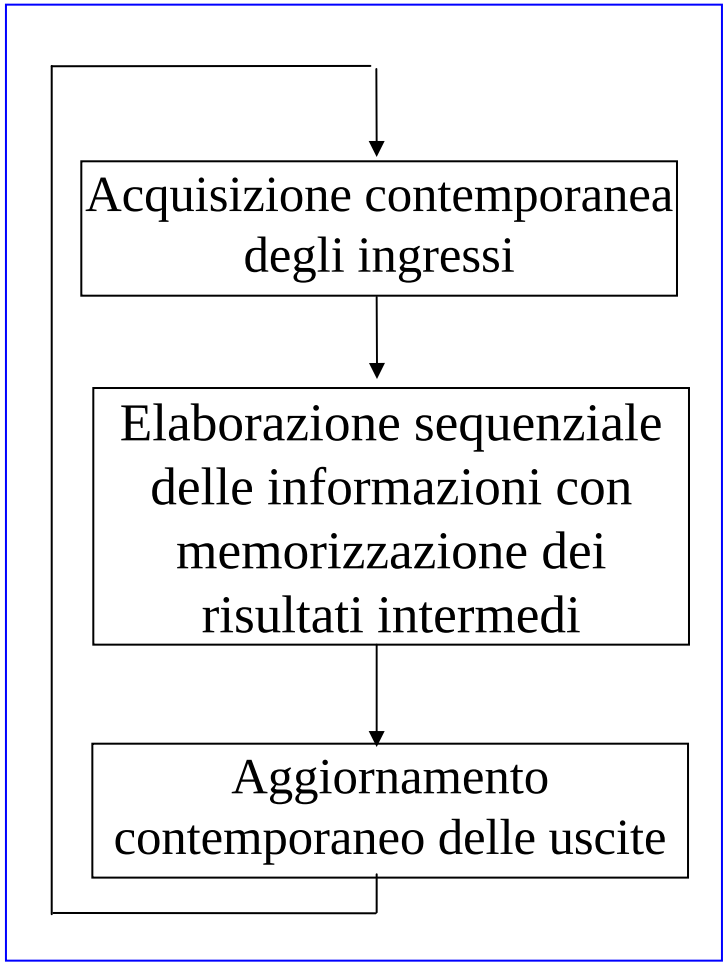
Sincrona d'Ingresso e d'Uscita

Essa deve il suo nome al fatto che sia la lettura che l'invio dei dati avvengono in sincronismo, ovvero contemporaneamente per tutti gli ingressi e per tutte le uscite.

Il PLC legge tutti gli ingressi all'inizio del ciclo e crea in memoria un immagine del processo in quel momento.

Tale immagine resta invariata per tutto il suo ciclo, anche se durante lo stesso alcuni ingressi potrebbero essere modificati.

Vengono poi eseguite tutte le elaborazioni previste dal programma e alla fine del ciclo sono attivate o aggiornate le uscite, per poi dar seguito, all'inizio del ciclo successivo, a una nuova lettura degli ingressi.

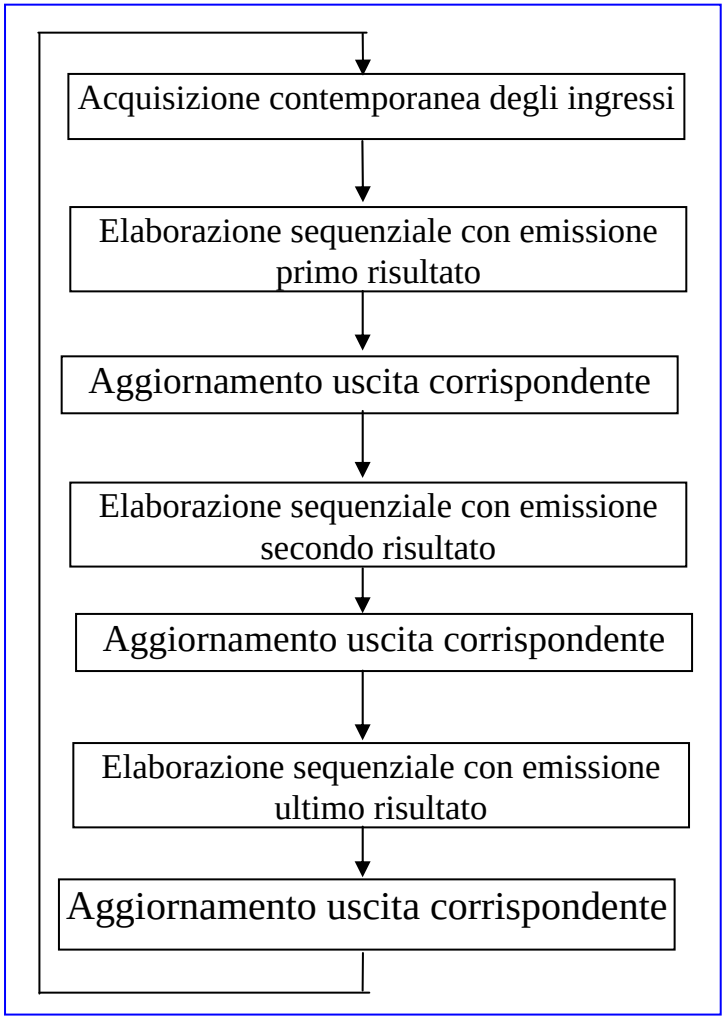


Scansione Sincrona d'Ingresso e Asincrona d'Uscita

Questo tipo di ciclo si dice sincrono d'ingresso, perché tutti gli ingressi vengono letti contemporaneamente, ed asincrono d'uscita perché l'invio alle uscite viene fatto in tempi diversi.

Gli ingressi vengono letti una sola volta all'inizio dell'elaborazione ma le uscite vengono attivate di volta in volta quando viene risolta la funzione logica corrispondente.

In questo caso l'aggiornamento è più rapido, ma si verificano problemi nel caso di un'uscita che ridiventa ingresso condizionando a sua volta un'ulteriore uscita. In questo tipo di ciclo, l'uscita che ridiventa ingresso non ha modo di agire se non con il ciclo successivo.



Scansione Asincrona d'Ingresso e d'Uscita

La scansione si dice asincrona d'ingresso e d'uscita poiché sia la lettura che l'invio sono effettuati in tempi diversi.

In questo caso sia le uscite che gli ingressi vengono aggiornati ogni volta che si ottiene un risultato durante l'esecuzione del programma.

Sistema Operativo

Tra le principali operazioni che tale sistema operativo gestisce ricordiamo:

- Scansione (già esaminata)
- autodiagnosi
- protezione dei dati
- funzione d’Interruzione (Interrupt)

Autodiagnosi

Alla fine di ogni scansione, immediatamente prima di ricominciare quella successiva, la CPU effettua un test autodiagnostico, ovvero un controllo sul funzionamento dei suoi circuiti interni.

Se viene riscontrata qualche anomalia il sistema emette un segnale di allarme, per avvertire gli operatori che qualche cosa non funziona nella CPU e contemporaneamente vengono disabilitate le uscite, per evitare eventuali danni alle persone o alle apparecchiature.

Il tipo ed il numero dei controlli dipendono dal particolare modello tuttavia quasi tutti rilevano:

- malfunzionamento del microprocessore
- malfunzionamento della memoria
- malfunzionamento delle schede I/O
- errori nella sintassi del programma utente
- tempo di scansione anomalo
- livello insufficiente della tensione fornita dalla batteria tampone

Protezione Dati

In caso di improvvisa mancanza d’alimentazione elettrica il sistema operativo provvede, automaticamente, a far alimentare le memorie, da una batteria tampone; ciò è indispensabile per evitare la perdita dei dati con conseguenze facilmente immaginabili.

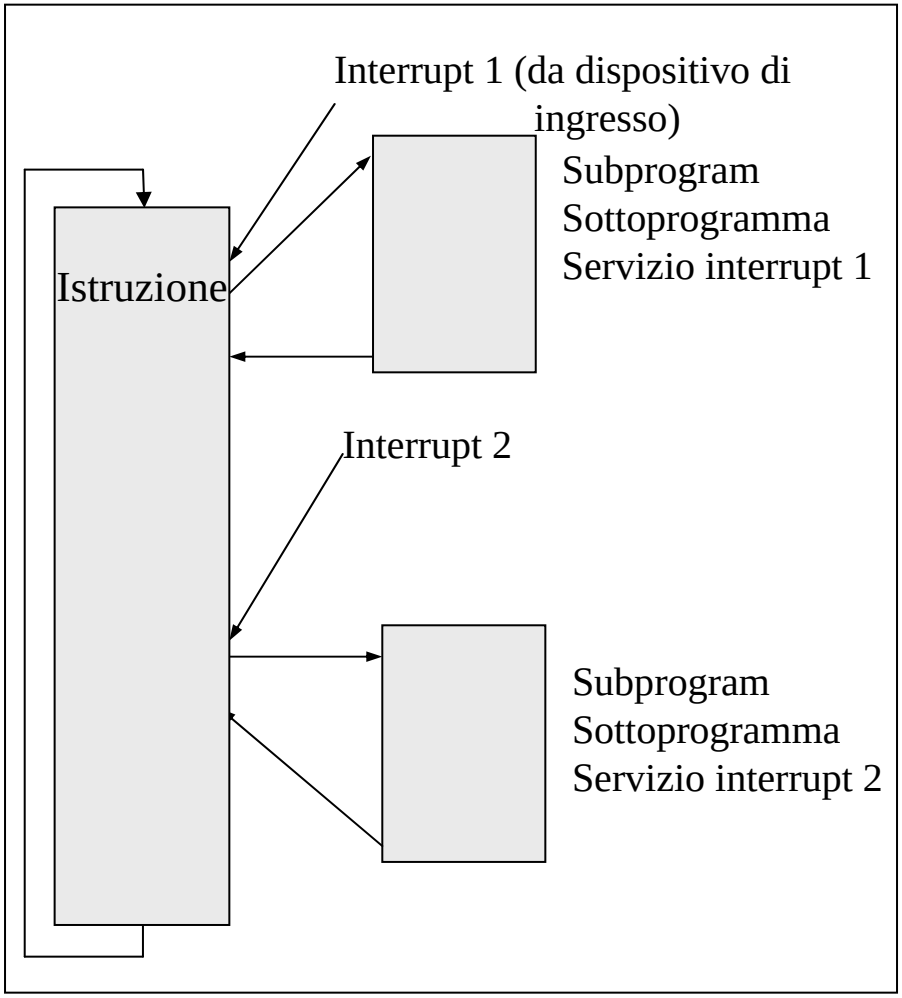
Per disporre di questa protezione occorre necessariamente mantenere la batteria ausiliaria in condizioni di perfetta efficienza; è questo il motivo per cui la verifica di tali prestazioni è inserita nel test di autodiagnosi.

Naturalmente le condizioni di avaria sono segnalate all’esterno attraverso una spia luminosa.

La Funzione d'Interrupt

L'elaborazione del «programma principale» viene immediatamente interrotta in seguito all'eventuale commutazione degli ingressi veloci per provocare l'esecuzione del cosiddetto «sottoprogramma»; ciò consente una più immediata reazione del microprocessore alle variazioni di quei segnali che vengono indicati come prioritari

Ad intervento effettuato la scansione del programma principale riprende dal punto in cui era stata interrotta.



BIT

Il bit è il valore logico di una variabile che può assumere solamente due stati: vero o falso (presente o assente), normalmente identificati mediante le prime due cifre decimali:

- 0 “per essere falso” (assente - non verificarsi di un evento)
- 1 “per essere vero” (presente - verificarsi di un evento)

Il bit può essere assegnato a una variabile di uscita o ad una variabile di ingresso o, ancora, ad una variabile interna e ne rappresenta il comportamento logico

- Un insieme di bit è detto Word (o parola)
- Si hanno parole di 8, 16, 32 bit
- Una parola costituita da 8 bit viene denominata byte

L’unità di misura della quantità di memoria è la parola che viene però espressa con il multiplo K (kilo) e pertanto si ha:

- 1 K byte = 1024 byte, dove 1024 nel sistema binario è la potenza di due più prossima al valore 1000: $2^{10} = 1024$

Memorie

I dispositivi che consentono registrare informazioni sono le memorie, sulle quali si possono scrivere e leggere dati

Le memorie si classificano sulla base dei seguenti parametri principali:

- possibilità di lettura e/o scrittura
- velocità di scrittura
- modalità di cancellazione
- comportamento in mancanza di alimentazione elettrica (volatilità)
- quantità di informazioni memorizzabili (capacità)

L'unità utilizzata dai costruttori di PLC, per definire la quantità di memoria di cui dispone il controllore è la KWord (chiloparola) abbinata alla lunghezza di word, che può essere di 4, 8, 16 o 32 bit

Esempio di Memorie

Dire che la memoria RAM di un PLC è di 3,5 Kword di 16 bit significa che essa è costituita da:

3,5 x 1024 = 3584 parole di 16 bit

3584 x 16 = 57344 bit

57344 : 8 = 7168 byte = 7 Kbyte oppure anche

3,5 Kw x 2 = 7 Kb essendo 1 word = 2 byte

Tipi di Memorie

Le memorie presenti nel PLC sono di tipo:

- ROM
- RAM
- EPROM
- EEPROM

Memoria ROM

È una memoria di sola lettura (Read Onlv Memory) è scritta dal costruttore per compiti specifici e mantiene il proprio contenuto anche in assenza di alimentazione elettrica (memoria non volatile)

La memoria ROM è utilizzata per la gestione del PLC (memoria del sistema) perché non può essere cancellata dall'utente

Memoria RAM

È una memoria ad accesso diretto (Random Access Memory), è scritta e letta dalla CPU ed è volatile, ma il suo contenuto può essere mantenuto attraverso una batteria tampone.

Memoria EPROM

È una memoria di sola lettura, cancellabile e riprogrammabile (Erasable Programmable Read Only Memory)

Non volatile, può essere letta dal processore e può essere scritta e cancellata dall'utente con l'impiego di dispositivi specifici (programmatore Eprom per la scrittura e raggi ultravioletti per la cancellazione)

Memorie del PLC

Le memorie del PLC si possono distinguere, in base al loro impiego, in:

- memoria di sistema
- memoria di programma
- memoria dati

Memoria di Sistema

La memoria di sistema serve a conservare tutte quelle particolari istruzioni necessarie per la gestione ed il controllo del funzionamento della CPU e che pertanto costituiscono il vero e proprio «sistema operativo» del PLC.

La memoria di sistema, dato il contenuto, di primaria importanza per il controllore, viene realizzata dal costruttore mediante ROM, allo scopo di impedire la sua involontaria cancellazione

Memoria di Programma

In questa memoria vengono registrate le istruzioni del programma che il PLC deve eseguire.

Per svolgere tale funzione essa deve essere accessibile all'utente e pertanto viene realizzata con RAM o con EPROM.

Normalmente il controllore offre la possibilità, attraverso la commutazione di un selettore, di usare, in alternativa tra loro, una RAM o una EPROM per sfruttare le caratteristiche positive di entrambe.

Per sviluppare e mettere a punto il programma si utilizza la memoria RAM, che può essere scritta e corretta, restando installata nel PLC, con l'unità di programmazione del controllore.

Nel funzionamento normale conviene sfruttare le caratteristiche delle EPROM che offrono i notevoli vantaggi di non poter essere accidentalmente modificate, di non aver bisogno di batteria tampone in caso di mancanza di alimentazione elettrica e di poter essere duplicate.

Quest'ultima caratteristica si dimostra utile per l'archiviazione dei programmi e per il trasporto degli stessi da un PLC ad un altro.

Memoria Dati

La memoria dati è suddivida in due parti:

- memoria d'ingresso
- memoria d'uscita

Nella prima viene memorizzato lo stato, continuamente aggiornato, dei sensori di campo collegati al controllore.

Nella seconda il processore scrive, ad ogni scansione, lo stato delle uscite che deve essere trasmesso all'esterno.

Sono necessariamente realizzate con memorie RAM; nel funzionamento normale sono accessibili solo al processore ma per favorire la fase di messa a punto è possibile attivare una modalità di lavoro che consente l'accesso alla memoria dati anche all'utente che può, simulando lo stato, o come si dice in gergo «forzando gli I/O», verificare la rispondenza del programma alle specifiche.

Alimentatore

Il microprocessore necessita di alimentazione ad una tensione continua e stabilizzata di pochi Volt (in genere 5 V), mentre per i circuiti di ingresso uscita è necessaria una tensione di 12 o 24 Volt sempre in corrente continua.

Queste tensioni si ottengono con un alimentatore di adeguate caratteristiche.

Sistema BUS

L’interconnessione tra le varie parti del PLC avviene tramite un “SISTEMA BUS”, cioè un sistema di collegamenti interni per la trasmissione e lo scambio di segnali, tensione di alimentazione e potenziale di massa

Il sistema BUS è suddiviso in più gruppi di segnali:

- bus degli indirizzi, tramite il quale il processore specifica l’indirizzo della cella di memoria o del dispositivo di I/O a cui vuole accedere
- bus dati, tramite il quale è possibile lo scambio di dati tra il processore, le memorie e i moduli di I/O
- bus di alimentazione, tramite il quale l’alimentatore fornisce le tensioni di alimentazione ai vari elementi del PLC

Selettore della Modalità Operativa

Tutti i PLC possono usarsi almeno in due diverse modalità operative, il modo esecuzione ed il modo apprendimento o programmazione.

Per selezionare le modalità di funzionamento è disponibile esternamente un selettore.

Quando esso è in posizione di programmazione le uscite sono disabilitate e pertanto possono essere introdotti o editati programmi.

In modo esecuzione le uscite vengono abilitate e quindi il programma contenuto in memoria è operante.

Connettore per Unità di Programmazione

Per inserire i programmi nella memoria del controllore occorre collegare ad esso una delle unità di programmazione previste dal costruttore.

E’ necessario quindi un opportuno dispositivo di comunicazione tra la memoria di programma ed il dispositivo programmatore.

Selettore RAM / EPROM

Abbiamo già detto della possibilità di funzionamento sia con RAM che con EPROM; per rendere possibile questa prestazione il PLC deve essere dotato di uno zoccolo porta-memoria estraibile, per l'interscambio RAM/EPROM e viceversa, oppure di uno zoccolo porta EPROM supplementare e di un selettore che avvisa il processore circa il tipo di memoria che deve leggere.

Relè di RUN o Inibitore delle Uscite

Si tratta di un dispositivo di sicurezza che in modo esecuzione è attivato e pertanto abilita le uscite.

Quando si verifica una qualche irregolarità rilevata dall'autodiagnosi o viene a mancare tensione, il relè si disattiva automaticamente interrompendo il programma e disabilitando le uscite.

Tale dispositivo è fornito di due contatti esterni utilizzabili per realizzare un allarme per gli operatori in caso di guasto.

Circuiti di Autodiagnosi

Il controllore esegue ad ogni ciclo un test autodiagnostico.

Se si verifica un guasto ad uno qualsiasi dei dispositivi controllati si interrompe la scansione del programma, si apre il contatto del relè di RUN e si disabilitano le uscite.

Indicatori led I/O

Sul pannello frontale dei controllori monoblocco sono sempre previsti led luminosi, in quantità pari al numero dei punti I/O, che si accendono quando gli ingressi o le uscite corrispondenti sono attivi.

Morsettiere I/O

Ogni controllore deve avere una morsettiera cui cablare gli ingressi e le uscite esterne.

Allo scopo di non dover ricablare tutti i punti I/O ogni volta che si deve smontare il PLC, spesso tali morsettiere sono separabili da esso per mezzo di un connettore speciale appositamente predisposto.

Unità d’Ingresso / Uscita

L’unità ingresso / uscita è composta dai dispositivi adatti a consentire il dialogo del PLC con il gruppo di potenza.

Moduli d’ingresso

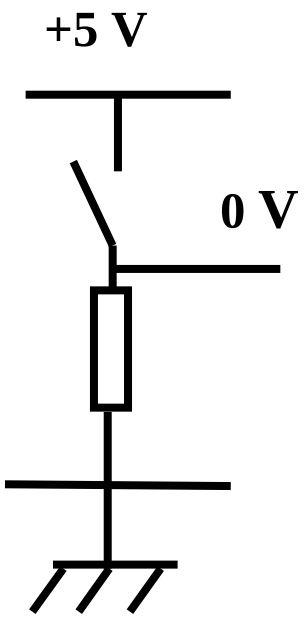
I moduli d’ingresso operano sui segnali provenienti dai sensori dell’impianto per renderli compatibili con la CPU.

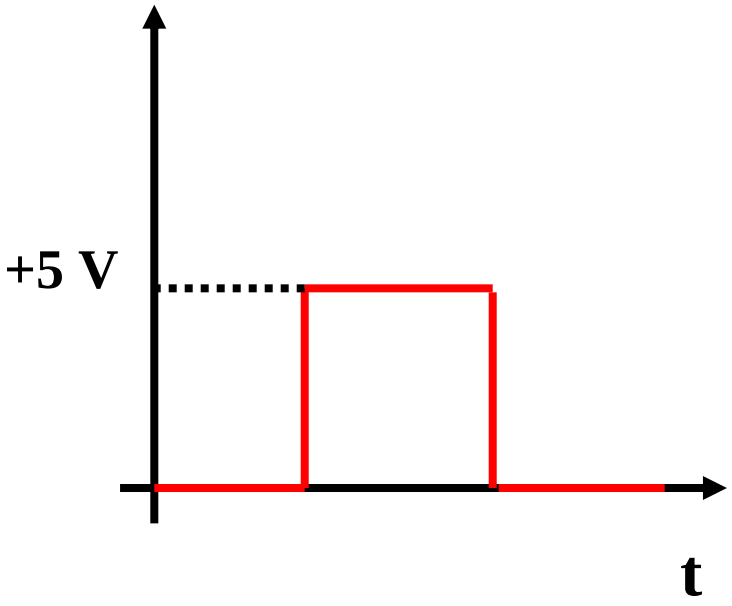
Le informazioni provenienti dal processo controllato (tramite organi di comando, di impostazione, trasduttori, sensori) possono essere definite con il termine generico di Segnale.

I segnali provenienti dai dispositivi in campo si possono distinguere in analogici o digitali.

Segnali Digitali

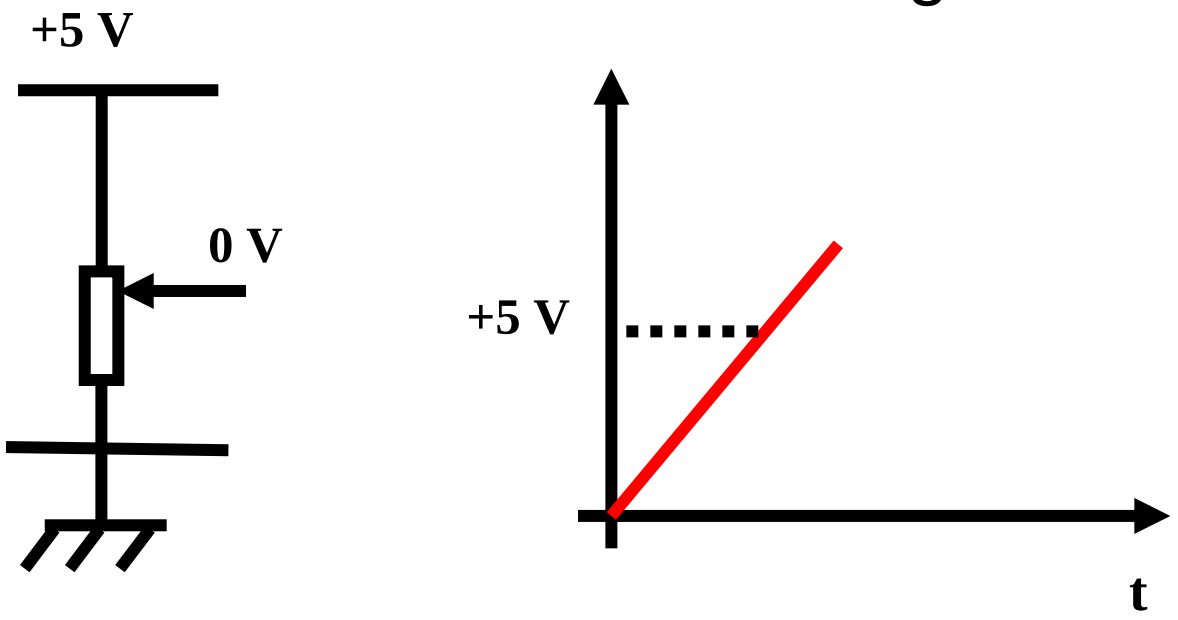
Possono assumere solo due livelli di tensione denominati convenzionalmente alto (H o 1) e basso (L o 0)





Segnali Analogici

Possono assumere infiniti livelli di tensione all'interno di un certo intervallo o range



Moduli d'Interfaccia

L'unità centrale è realizzata con circuiti integrati digitali che operano con la tensione di + 5 Volt c.c. e sono in grado di eseguire delle elaborazioni solo su segnali binari che assumono livelli di tensione di 0 Volt o 5 Volt

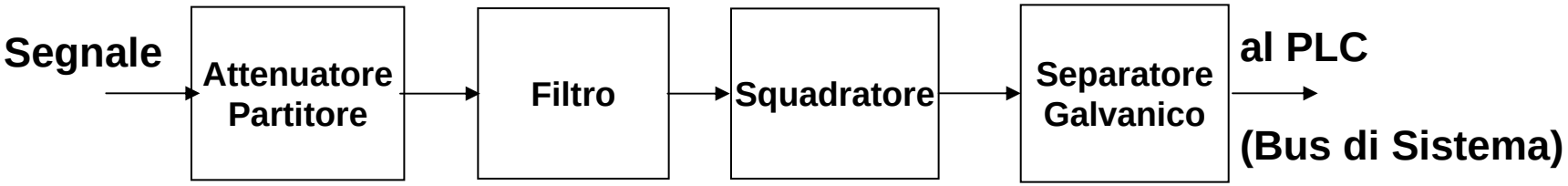
I moduli di ingresso fungono quindi da interfaccia tra la logica interna del PLC ed i segnali esterni, comunque generati.

Moduli d'Interfaccia

Questi moduli, come tutte le interfacce, attuano non solo un semplice collegamento, ma anche e soprattutto:

- un adattamento dei livelli di tensione e delle caratteristiche dei segnali
- il filtraggio dei disturbi
- la squadratura dei segnali
- la separazione galvanica dei segnali

Modulo d'Interfaccia



I segnali provenienti dall'esterno possono presentarsi con livelli di tensione diversi, ad esempio 12 Volt, 24 Volt ma anche 110 o 220 Volt e vengono quindi attenuati con opportuni partitori resistivi.

Per evitare che i disturbi elettrici (picchi di tensione sulla linea di alimentazione ritorni induttivi dagli attuatori, interferenze provenienti dal cablaggio) introducano false informazioni nel modulo di ingresso, vengono inseriti opportuni filtri che permettono di valutare non solo il livello logico del segnale, ma anche la sua durata e quindi il suo contenuto energetico.

Il segnale in uscita dal filtro viene successivamente squadrato perfettamente mediante uno speciale circuito elettronico a doppia soglia (trigger di Schmidt o comparatore di isteresi).

Gli ingressi sono normalmente isolati galvanicamente dalla logica interna del PLC, al fine di avere una separazione dei potenziali. Infatti, eventuali sbalzi e sovraccarichi potrebbero essere causa di danni irreparabili per i circuiti integrati interni.

Modulo d'Interfaccia

- La tecnica maggiormente adottata per questa separazione fa uso di optoisolatori: il segnale in arrivo attiva un LED (Light Emitting Diode - diodo emettitore di luce), il quale, emettendo luce, attiva un elemento fotosensibile (in genere un fototransistor) che genera il segnale di livello adatto verso la logica interna del PLC.
- Se il segnale è di tipo analogico viene sottoposto ad una conversione da analogico a digitale ricevendo una rappresentazione binaria. Come si vede, non esiste continuità galvanica (o metallica) tra ingressi e logica interna, ma i segnali si propagano ugualmente grazie all'accoppiamento ottico tra LED e fototransistor.

Funzioni di un Modulo d'Ingresso

Riassumendo, le funzioni di un modulo di ingresso sono:

- riconoscimento sicuro ed affidabile del livello logico 0 oppure 1 dei segnali
- trasferimento di questa informazione alla logica interna del PLC
- protezione della logica interna da eventuali sovratensioni, quindi isolamento galvanico tramite optoisolatori
- soppressione dei segnali di disturbo
- adattamento del segnale di ingresso ai livelli di tensione della logica interna della CPU del PLC

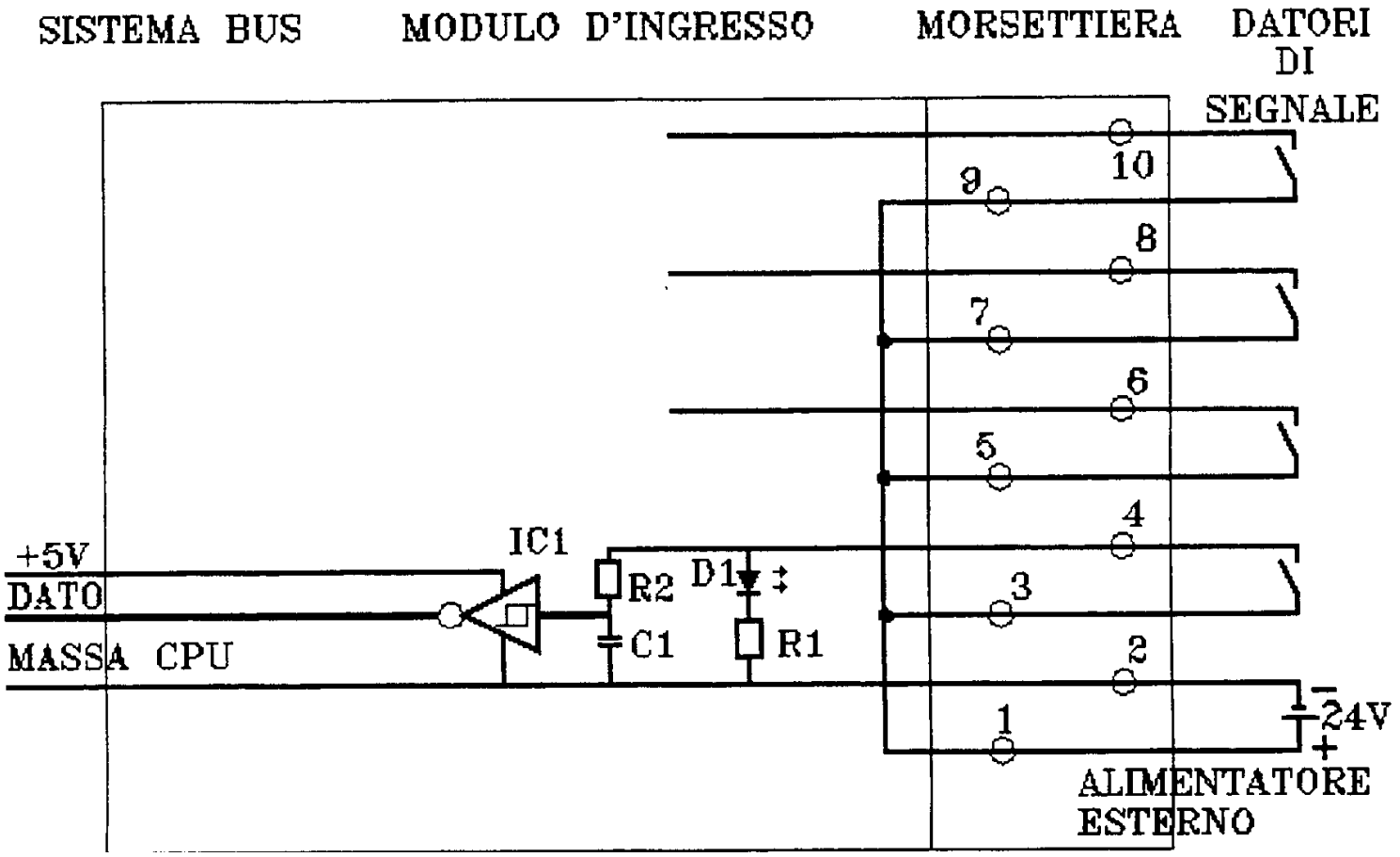
Moduli d'Ingresso Disponibili

In generale l'Unità d'ingresso è modulare ed è in grado di trattare vari tipi di segnali.

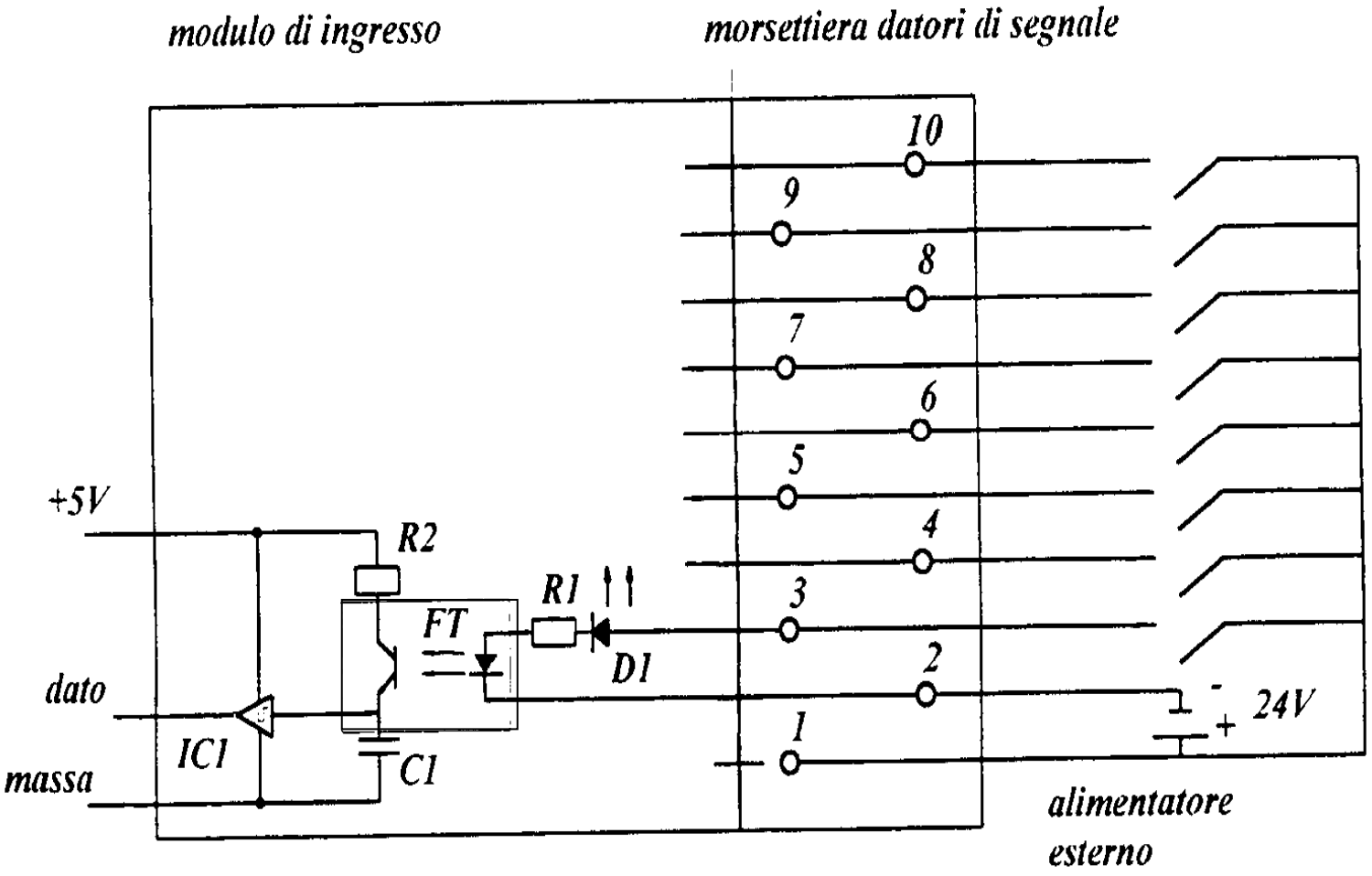
Sono normalmente disponibili:

- moduli di ingresso per segnali digitali in C.C. (con e senza separazione galvanica)
- moduli di ingresso per segnali digitali in C.A. (con e separazione galvanica)
- moduli di ingresso per segnali analogici (in tensione o corrente)

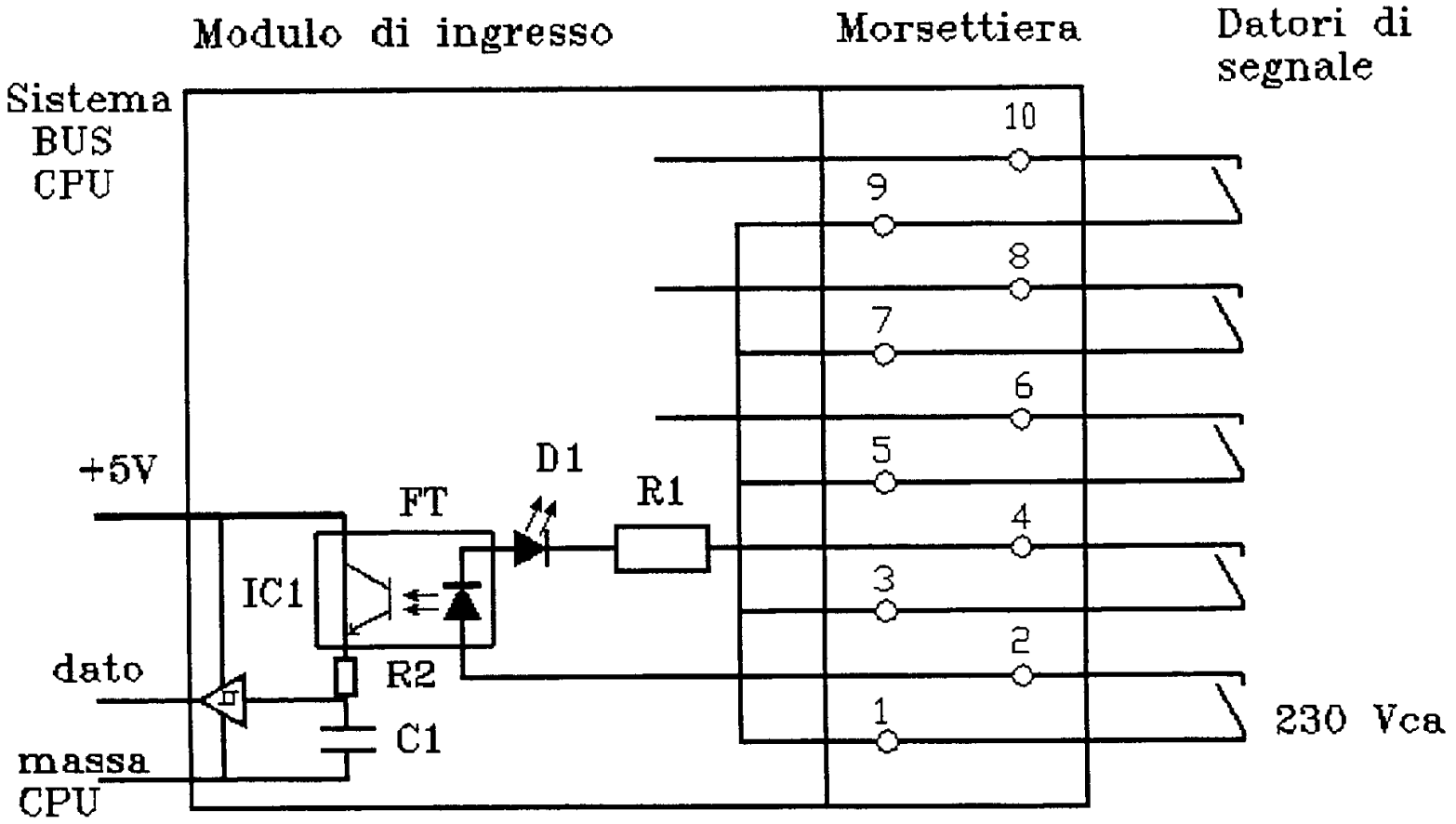
Modulo di ingresso digitale senza separazione galvanica (4 x 24 V DC)



Modulo di ingresso digitale con separazione galvanica (8 x 24 V DC)



Modulo di ingresso digitale con separazione galvanica (4 x 230 V AC)



Moduli di Uscita

I moduli di uscita rappresentano sostanzialmente l'interfaccia tra l'elaborazione del programma, attuata dalla CPU del PLC e gli attuatori che costituiscono il sistema di comando verso l'impianto controllato.

Le decisioni logiche formulate dal processore vengono inviate tramite il modulo di uscita agli attuatori di comando e agli elementi di segnalazione.

Il modulo di uscita:

- converte il livello dei segnali, che escono dalla CPU negli opportuni livelli richiesti dai dispositivi in campo
- isola la CPU dai transitori che si possono trovare nella rete esterna

Segnali di Uscita

I segnali richiesti dagli attuatori si possono suddividere in due categorie:

- segnali digitali: per gli attuatori ON - 0FF la cui “risposta” può assumere solo due valori (es.: fermo / in movimento)

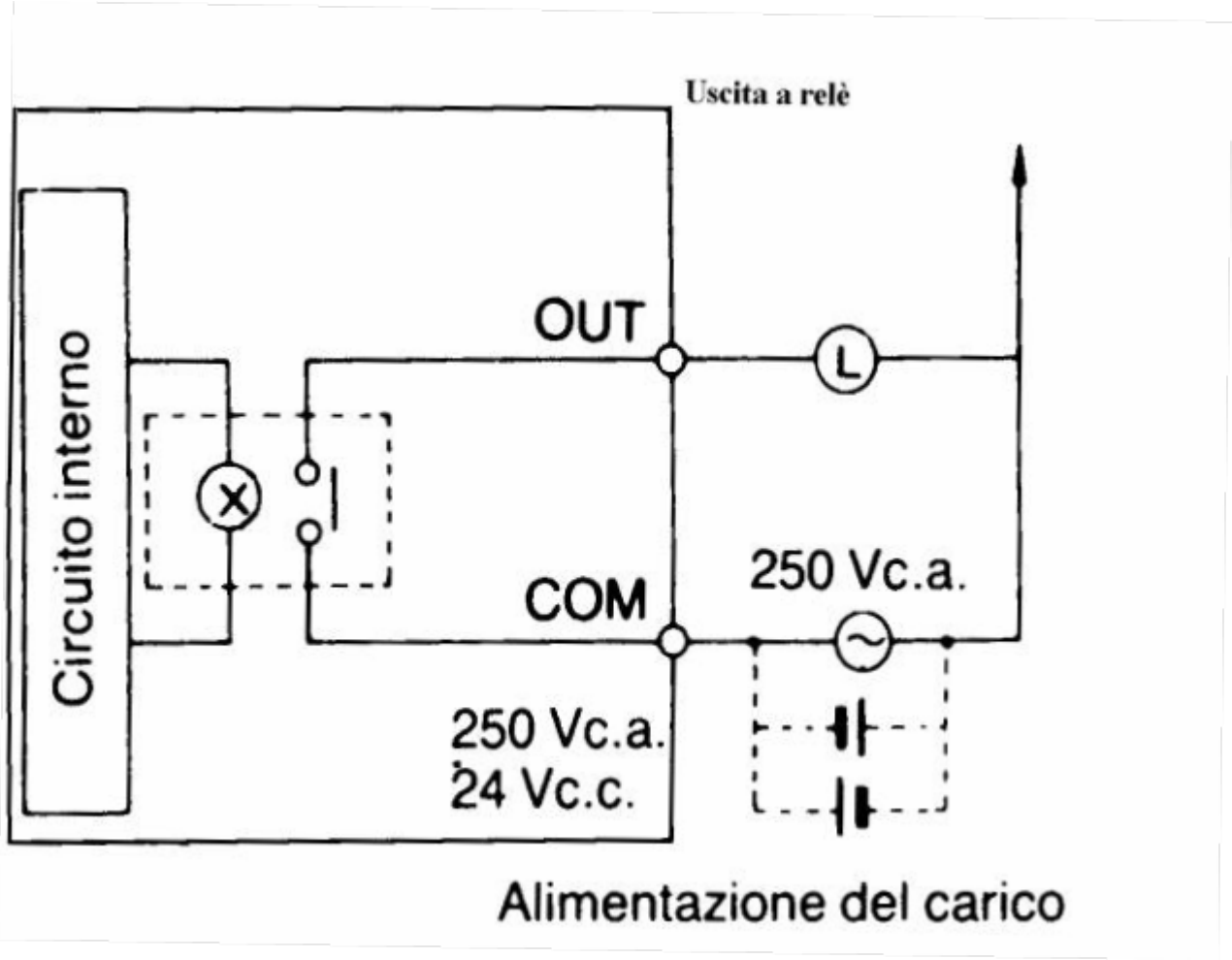
- segnali analogici: per gli attuatori proporzionali la cui “risposta” può assumere infiniti valori all’interno di un certo “range” (es.: movimento con velocità proporzionale al segnale ricevuto)

Moduli di Uscita Digitali

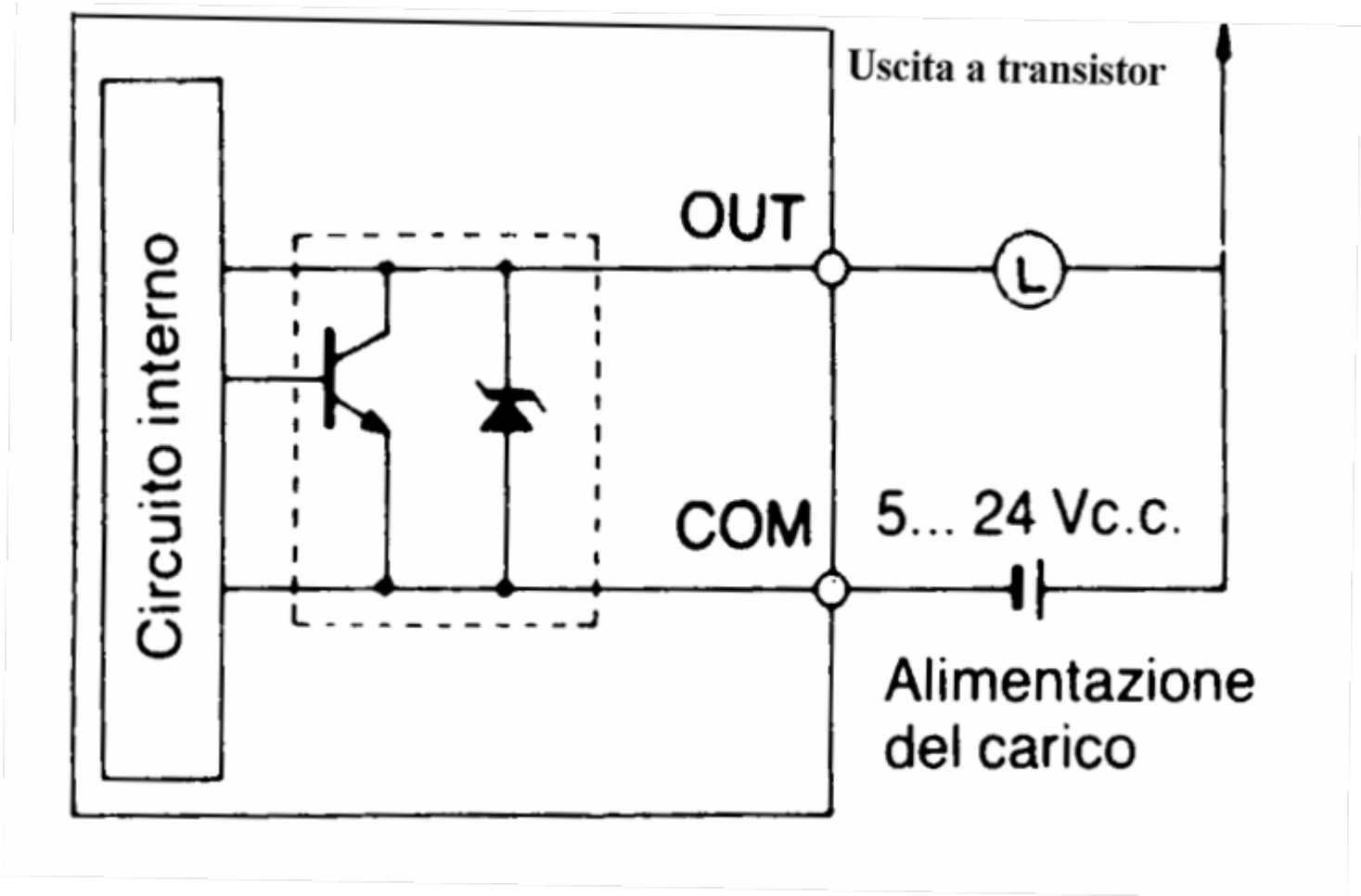
Sono normalmente disponibili moduli di uscita:

- a transistori (con e senza separazione galvanica) per apparecchiature a C.C. (5-12-24 V)
- a relè per apparecchiature a C.C. (5-12-24 V)
- a relè per apparecchiature a C.A (110- 220 V)
- a triac per apparecchiature a C.A. (110 - 220 V)

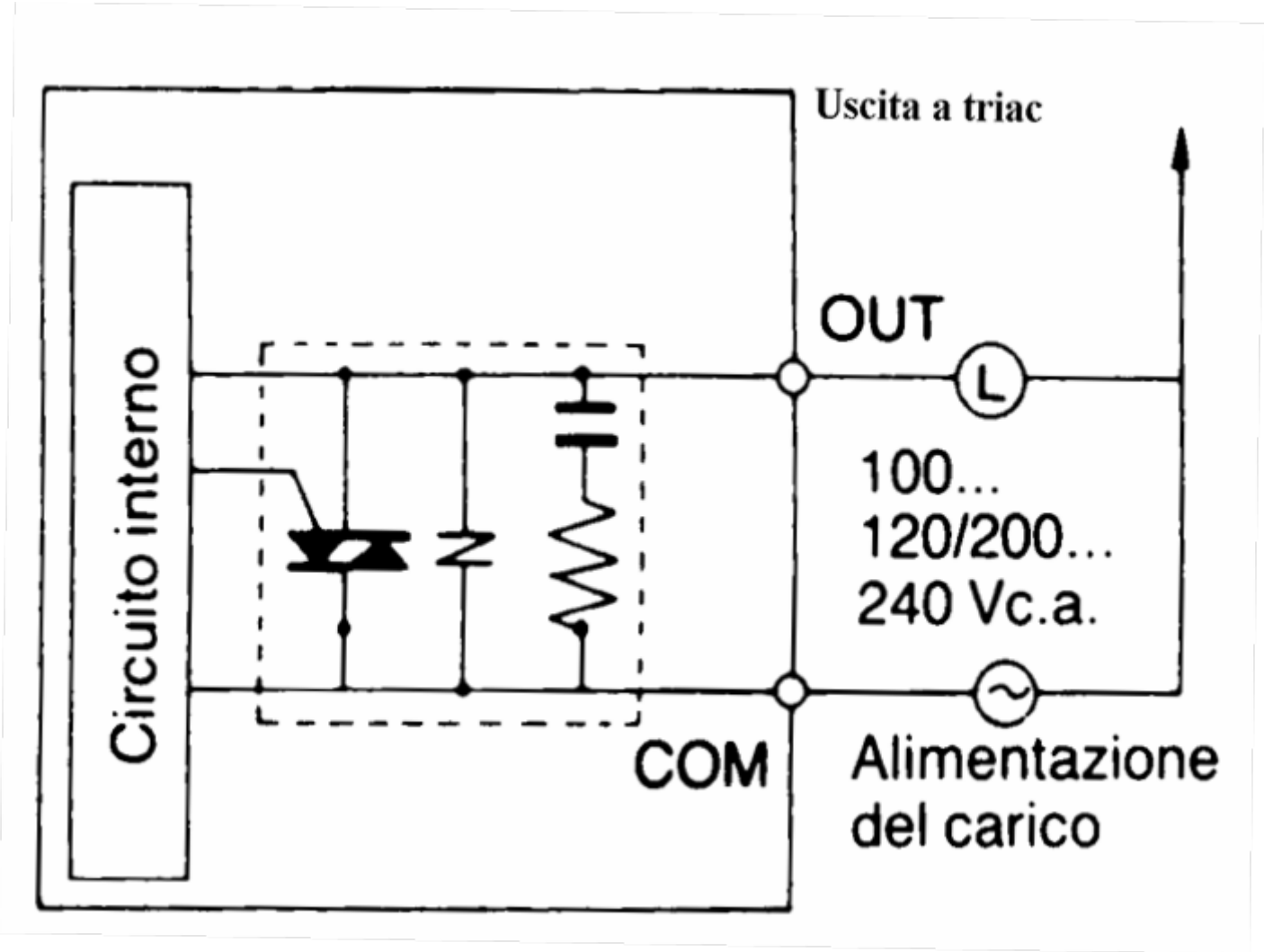
Uscite a Relè



Uscita a Trasistor



Uscita a TRIAC



Interfacce per Comunicazione e Trasmissione Dati

I PLC utilizzati in sistemi di controllo complessi devono essere collegati in rete tra loro, e con calcolatori elettronici, allo scopo di rendere possibile l'integrazione tra i vari livelli dell'automazione.

La circolazione dei dati avviene grazie a particolari dispositivi, come adattatori, diramatori, ripetitori, modem, ecc..

Sistemi di Comunicazione

I sistemi di comunicazione più usati sono:

- RS422, RS232, fibre ottiche

La RS422 si usa per il collegamento tra PLC cui si deve ricorrere quando è richiesta una quantità di punti I/O, o di memoria, maggiore di quella permessa da un solo controllore.

La RS232 si rende necessaria per collegare il PLC ad un adattatore di linea o direttamente ad un PC.

L'adattatore serve nel caso in cui si devono trasformare in livelli RS232 i dati trasmessi su fibre ottiche o su cavo RS422.

Si ricorre alle fibre ottiche quando occorre garantire un'elevata immunità ai disturbi del PLC.

Infine i principali parametri da considerare sono i seguenti:

- velocità di trasmissione
- distanze massime di trasmissione
- numero massimo di controllori collegabili

Individuazione degli Ingressi e delle Uscite

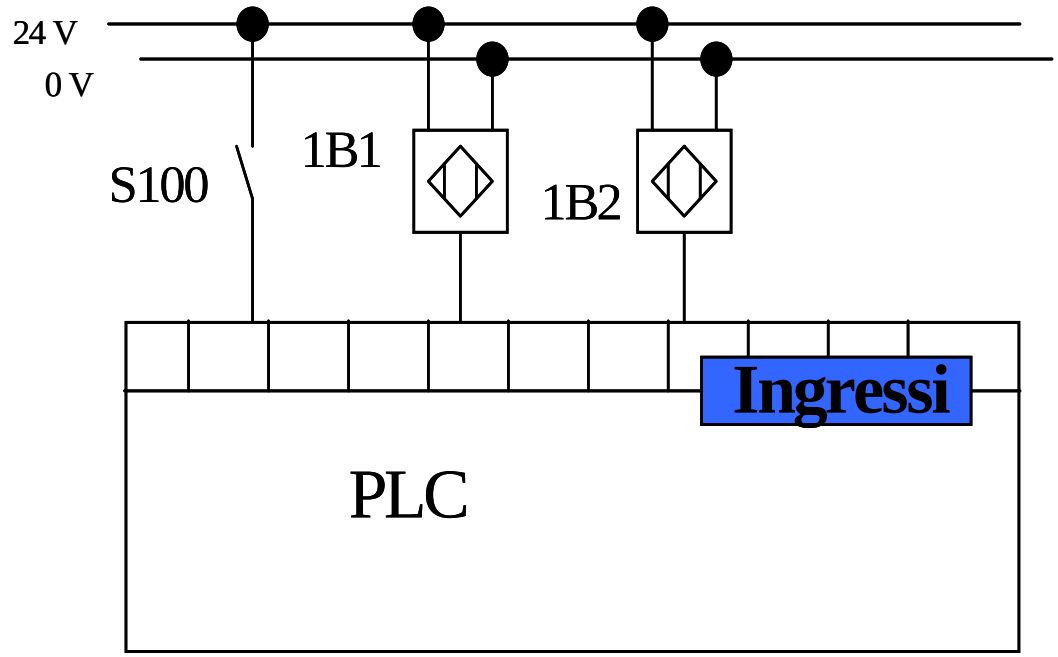
Ad ogni ingresso e ad ogni uscita viene assegnato un indirizzo univoco al quale occorre fare riferimento durante la stesura del programma.

Ogni ingresso o uscita possiede un indirizzo relativo al modulo stabilito dalla struttura circuitale e riportato sulla morsettiera (es.: per un modulo di 8 elementi l'indirizzo relativo va da 0 a 7) e un indirizzo di base che può dipendere dalla posizione del modulo all'interno del rack.

Per individuare l'indirizzo assoluto di un ingresso o di una uscita occorre sommare all'indirizzo di base del modulo l'indirizzo relativo dell'I/O.

Nomenclature dei Morsetti d'Ingresso

Le nomenclature dei morsetti di ingresso / uscita dipendono dal costruttore.



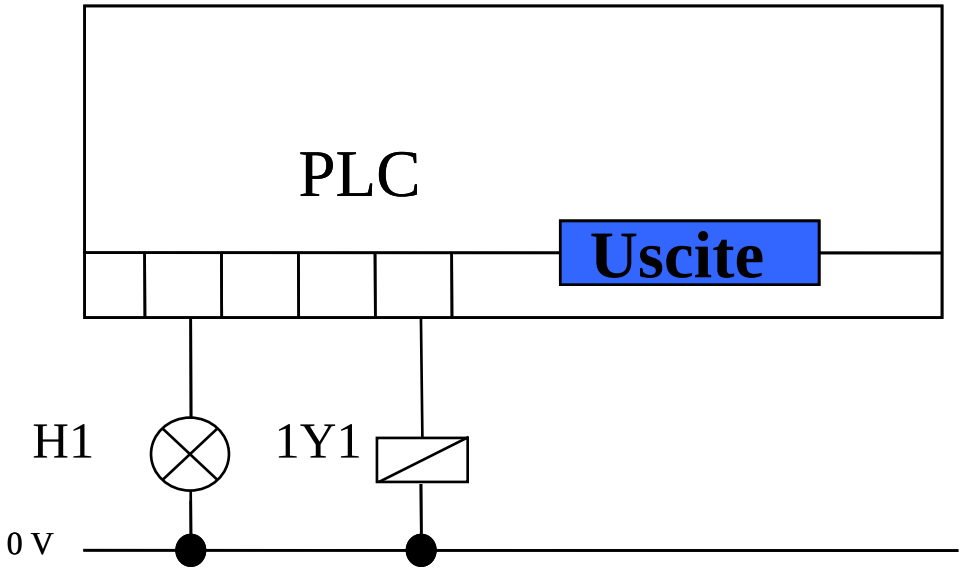
Ad esempio:

I 1.0

Indica la connessione al morsetto 0 del modulo 1, dove

I sta per *Input*.

Nomenclature dei Morsetti di Uscita



Ad esempio:

Q 1.0

Indica la connessione al morsetto 0 del modulo 1.

Dialogo Operatore - Macchina

Per gestire i sistemi automatizzati è necessario stabilire un dialogo tra operatore e macchina.

È quindi indispensabile un'interfaccia che, permettendo il dialogo tra l'operatore ed il PLC, consenta l'introduzione in memoria del suddetto programma.

Tale funzione è svolta dall'unità di programmazione chiamata anche «console» del «programmatore».

Funzioni Fondamentali della Console

L’unità di programmazione deve garantire almeno le seguenti operazioni fondamentali:

- scrittura del programma nella memoria del PLC
- lettura del programma contenuto nel controllore
- modifica (editing) e controllo (test) del programma
- ricerca di istruzioni del programma
- compilazione del programma

Poiché il programma viene scritto dall’utente in un linguaggio comprensibile ad esso ma non direttamente alla CPU, la console è fornita di un «compilatore» ovvero di un software residente su ROM che traduce i codici utilizzati dall’operatore in «codici macchina» comprensibili al processore del PLC.

Le Funzioni del PLC

I controllori programmabili simulano al loro interno un certo numero di operazioni ed inviano alle uscite segnali che sono il risultato di tali simulazioni.
Ogni PLC possiede un certo “magazzino funzioni”.

Ingressi esterni (Contatti)	Uscite esterne (Bobine)	Uscite interne ritentive (relè)	Uscite esterne non ritentive (relè)
Registro a scorrimento	Temporizzatori	Contatori	Uscite di controllo speciali (relè)
	Altre varie	Sequenziatori logici	Operatori Matematico/logici

Tipi di Uscite

Gli ingressi e le uscite esterne sono gli unici dispositivi effettivamente presenti e pertanto devono essere cablati, ovvero collegati elettricamente alla morsettiera del PLC.

Le uscite di controllo interne possono essere di due tipi:

- ritentive
- non ritentive

Relè Ritentivi e non Ritentivi

I relè non ritentivi non mantengono lo stato in caso di caduta di alimentazione; questo significa che se un relè di questo tipo si trova in posizione On al momento dell'interruzione dell'alimentazione della CPU, esso si troverà in posizione 0ff quando questa sarà riattivata.

Le uscite interne ritentive sono anche dette di mantenimento, o relè Latch, perché mantengono il loro stato anche durante un'interruzione dell'alimentazione.

Blocchi Funzionali

Contatori, temporizzatori, sequenziatori e registri a scorrimento sono funzioni, dette di box o di blocco funzionale (o non a relè), che emulano i rispettivi dispositivi fisici

Elementi Funzionali Matematici

Gli operatori matematici consentono al PLC di effettuare operazioni aritmetico-logiche quali:

- Comparazione (>, <, = , < =, > =);
- operazioni aritmetiche (+, -, x, /, $\sqrt{}$);
- conversioni di codice numerico (ad esempio Decimale Binario e viceversa)

In genere anche queste funzioni sono dette di box.

Contatti e Bobine

I contatti e le bobine sono gli elementi fondamentali per l'elaborazione della logica a relè.

I contatti simulati dal controllore possono essere NA (Normalmente Aperto) e NC (Normalmente Chiuso).

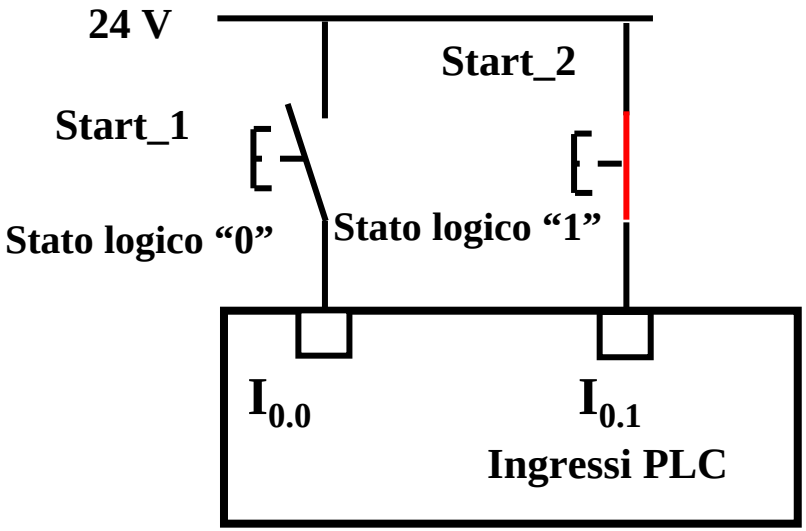
Si assegna:

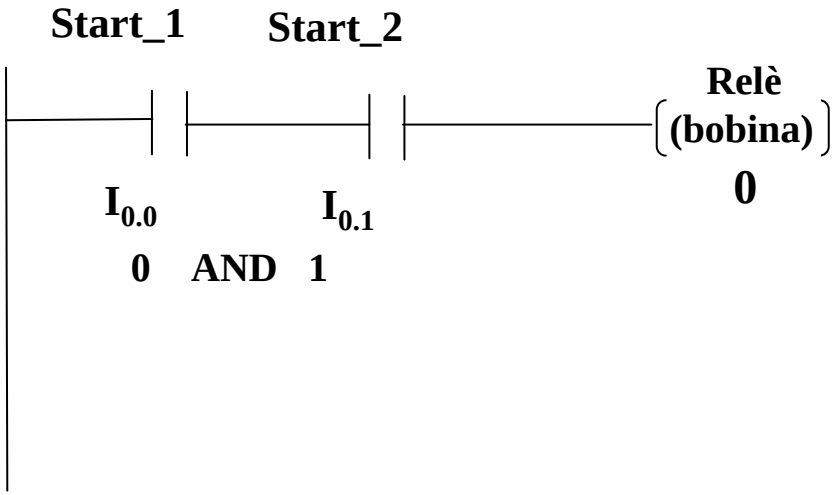
- ad un contatto aperto (che non si trova in tensione 0 V) lo stato logico "0"
- ad un contatto chiuso (che si trova in tensione 24 V) lo stato logico "1"

Ingresso NA

Un ingresso NA si comporta esattamente come l'unità collegata.

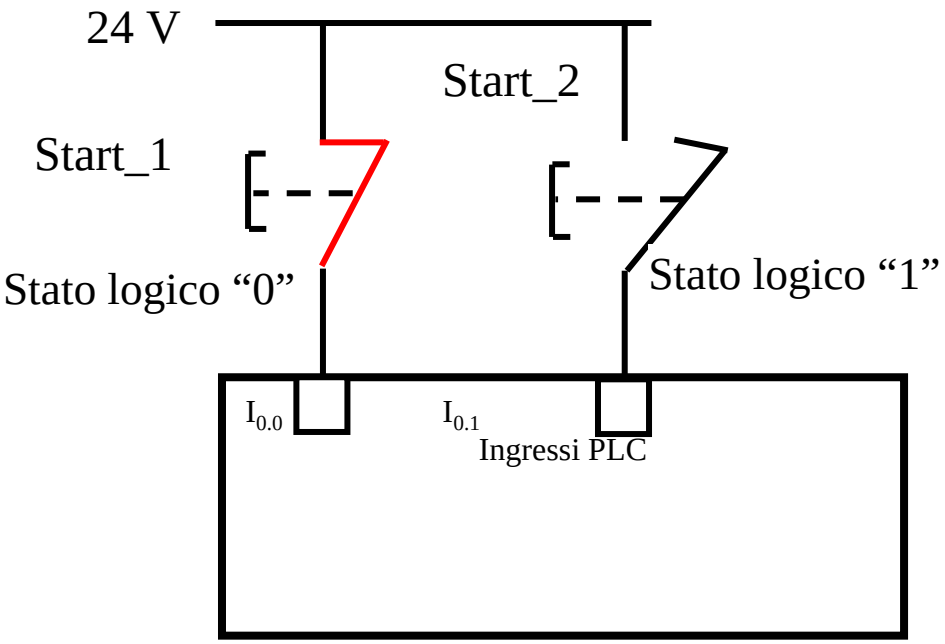
Se l'unità collegata è un circuito aperto (quindi con stato logico "0"), il contatto NA del programma resta un circuito aperto (con stato logico "0").

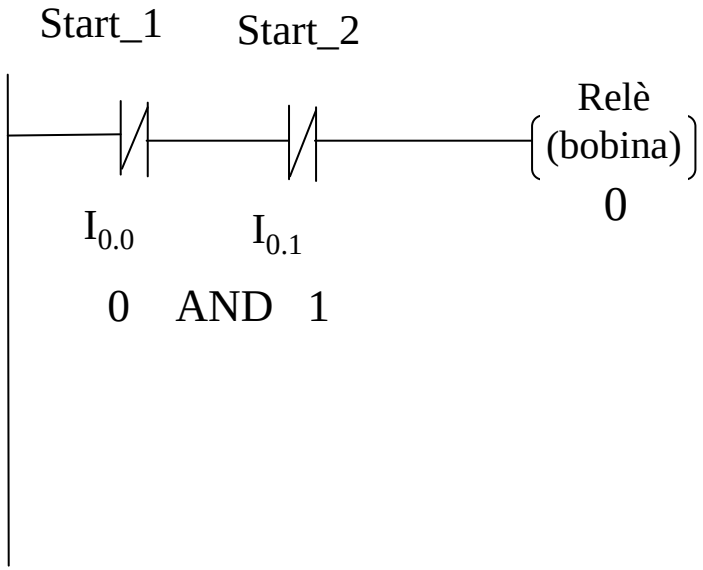




Ingresso NC

Un ingresso NC si comporta in modo opposto all'unità collegata; se questa è un circuito aperto il contatto è un circuito chiuso e viceversa.





Bobine

I contatti possono essere usati singolarmente, in serie, in parallelo, in serie-parallelo ed in parallelo-serie.

Una bobina si eccita (Valore logico “1”) solamente quando viene preceduta da un percorso di contatti che si trovano allo stato logico “1”.

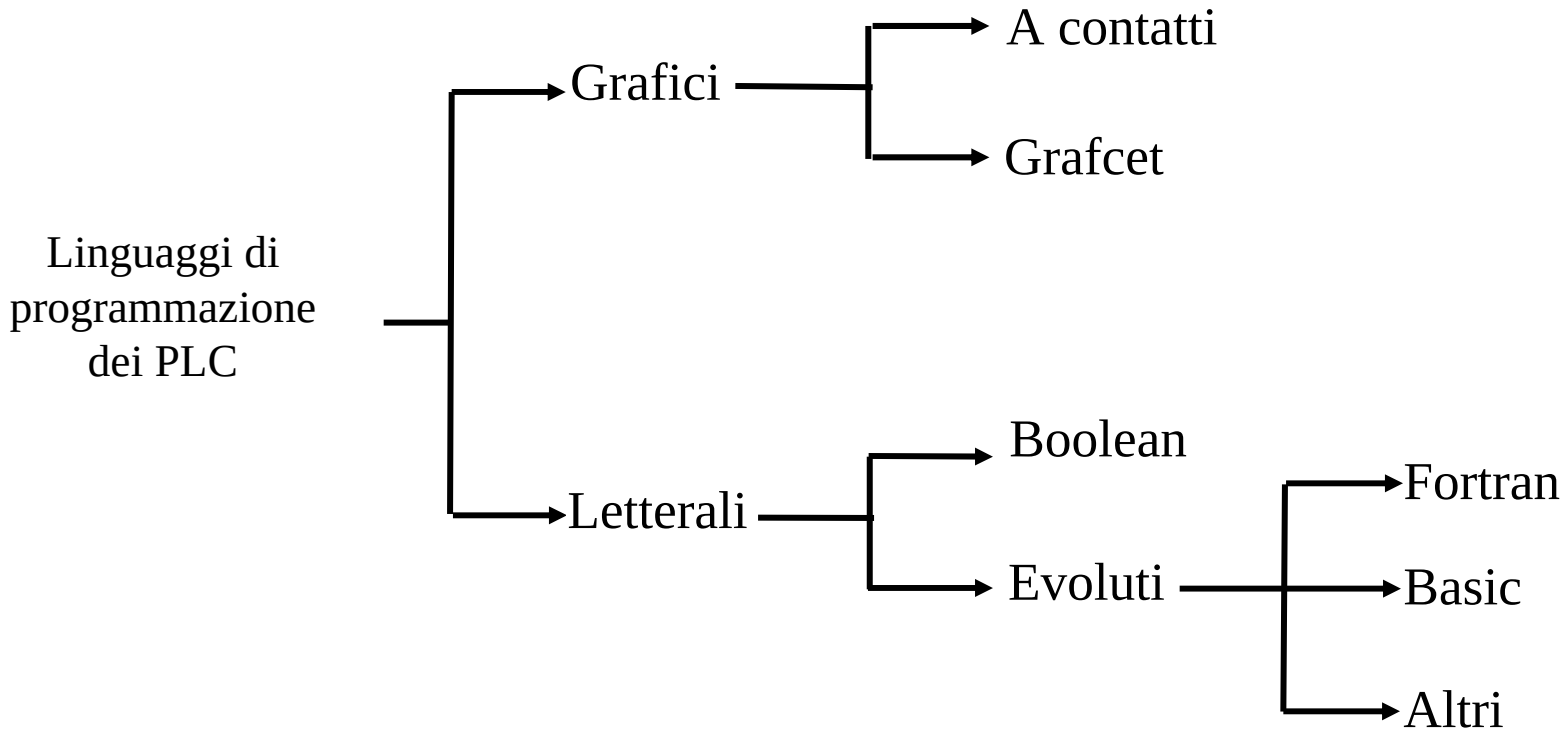
La bobina, rappresenta l’operatore di scrittura, infatti consente di scrivere un bit in memoria.

Le bobine possono essere usate singolarmente o in parallelo.

I Linguaggi di Programmazione

Un linguaggio dì programmazione consiste nell'insieme delle regole che stabiliscono come scrivere le istruzioni necessarie a comunicare al PLC che cosa deve fare.

Essi appartengono essenzialmente a due tipologie: linguaggi grafici e linguaggi letterali.



Linguaggi Grafici

I linguaggi grafici sono basati sull'impostazione del diagramma a contatti, o del grafcet, che risolve l'applicazione dal punto di vista logico.

Il linguaggio a contatti (KOP) è molto diffuso ed è il primo linguaggio studiato per i PLC.

Il linguaggio grafcet si è diffuso, in tempi abbastanza recenti, parallelamente all'affermazione conseguita dal metodo grafcet per lo studio dei sistemi automatici, il cui più grande pregio consiste nell'estrema semplicità con la quale consente di analizzare processi sequenziali.

Linguaggi Letterali

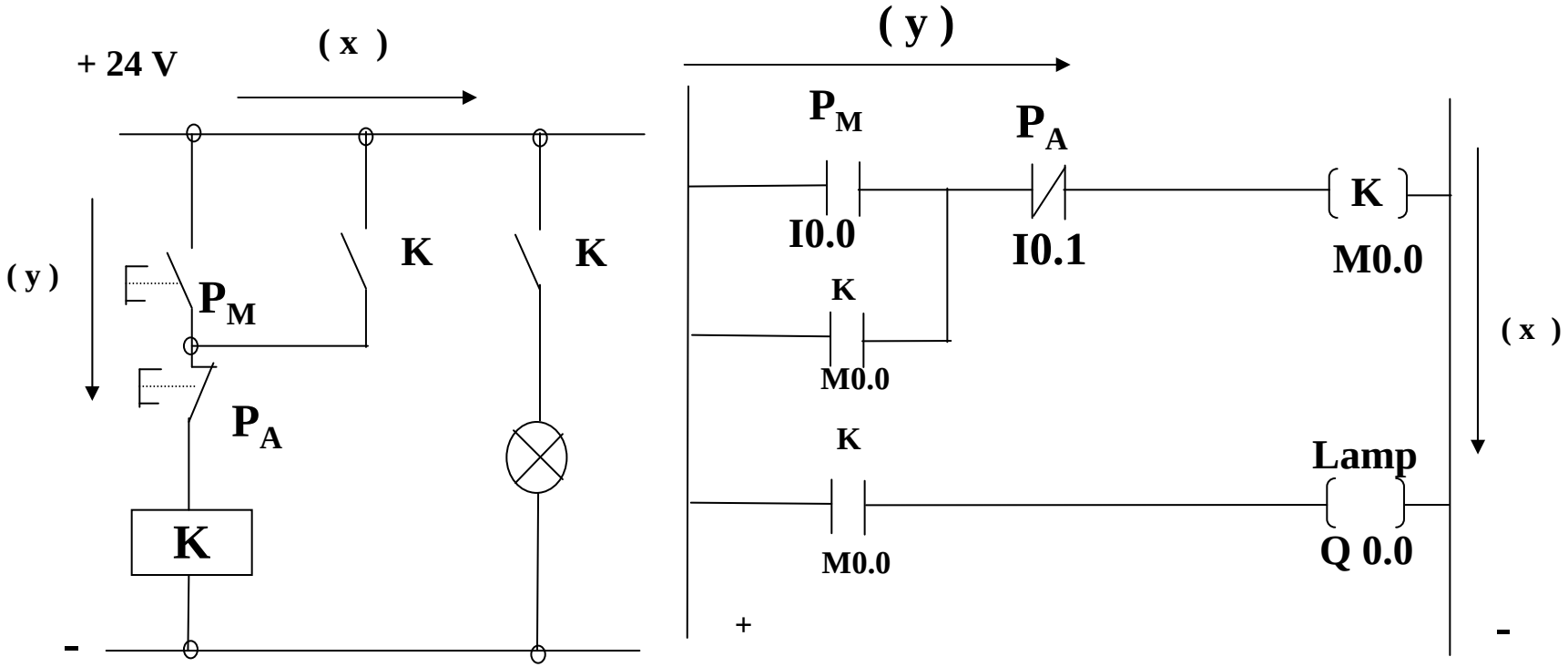
Ognuno di tali codici si esprime con parole che traducono o ricordano l'operazione cui si riferiscono.

Il booleano è noto anche sotto altri nomi: mnemonico, lista istruzioni, AWL sono i più frequentemente utilizzati dai costruttori che, generalmente, adottano proprie sigle per ogni linguaggio, con il risultato di dare un messaggio commerciale meno chiaro di quanto essi credono.

I linguaggi evoluti hanno avuto un notevole sviluppo soprattutto in ambienti dove, per certe mansioni, viene utilizzato personale con una cultura informatica di base.

Conversione degli schemi elettrici funzionali in diagrammi a contatti

Consideriamo lo schema elettrico funzionale del circuito di autoritenuta; in esso compaiono gli elementi principali della logica a relè ovvero: contatti NA, contatti NC, bobine e collegamenti.

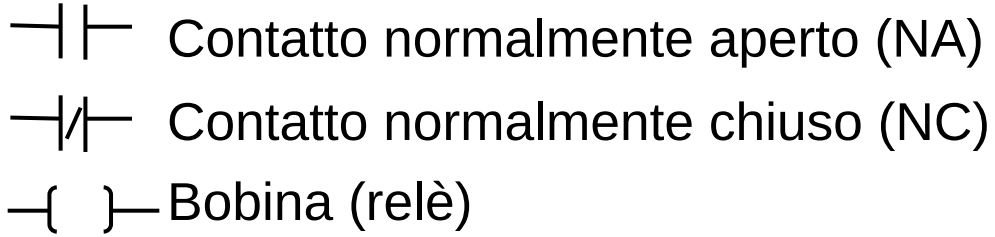


La freccia y indica il **flusso di potenza,** la freccia x indica il flusso logico.

Passaggio dallo schema elettrico al diagramma a contatti

Gli schemi elettrici funzionali sono trasformabili in diagrammi a contatti mediante alcune semplici modifiche:

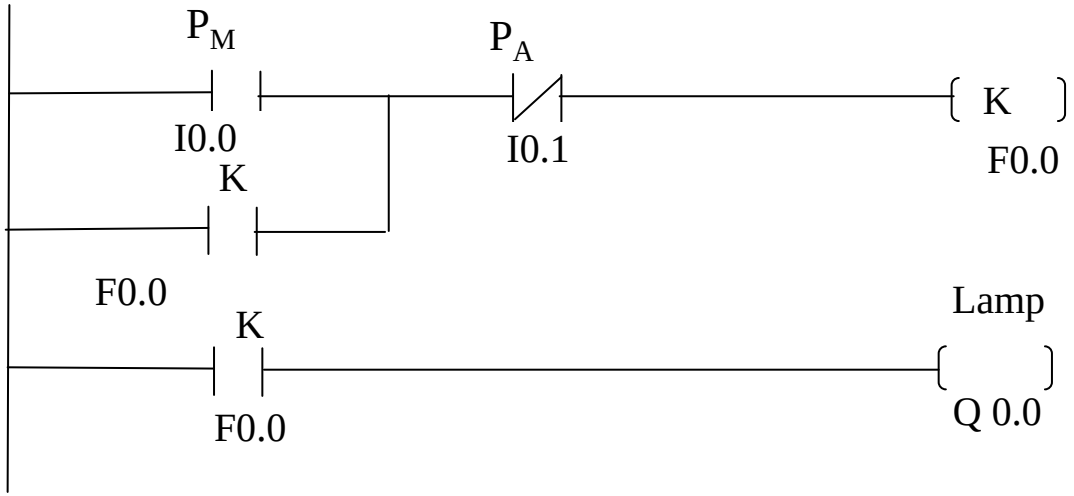
- le linee che rappresentano l'alimentazione vengono disegnate verticalmente;
- i simboli dei contatti NA, di quelli NC e delle bobine vengono semplificati nel modo seguente:



- Il flusso di potenza va da sinistra verso destra;
- il flusso logico va dall'alto verso il basso;
- ogni circuito viene disegnato in posizione orizzontale invece che verticale.

Diagramma a Contatti

Con tali semplicissime regole lo schema elettrico si trasforma nel seguente diagramma a contatti:



La linea verticale sinistra può essere pensata come la linea in cui è presente la tensione, quella destra come la linea posta a massa.
Con tale convenzione il flusso di energia può andare solo da sinistra verso destra ed è pertanto qui che deve essere posta la bobina da azionare.
La linea a sinistra prende il nome di barra, quella destra può addirittura essere omessa.

Programma in KOP

Il diagramma considerato è costituito da un solo circuito o, come spesso si dice, da una sola linea logica; quando nello schema ci sono più circuiti, le linee che li rappresentano vengono poste in parallelo alla prima, dando luogo ad un'immagine grafica molto vicina a quella di una scala a pioli; da questo fatto discendono i nomi ladder o diagramma a scala.

Il tratto di ciascuna linea logica a sinistra della bobina viene a volte chiamato zona di test.

I contatti NA ed NC, se uniti da linee orizzontali realizzano la funzione AND, se uniti in verticale realizzano la funzione OR.

GRAFCET

Diagramma funzionale sequenziale

GRAFCET
Diagramma funzionale sequenziale

È una tecnica di rappresentazione grafica messa a punto dall'ente francese ADEPA(*).

Il grafcet parte dalla considerazione che ogni comando automatico comprende una parte di potenza e una parte di comando e che il legame tra le due è dato da segnali di potenza e da segnali di pilotaggio.

Nel grafcet vengono rappresentate le diverse fasi di un ciclo, numerate secondo l'ordine di esecuzione a partire dalla prima, è preceduta da uno stato 0

Lo stato 0 rappresenta lo stato iniziale di riposo del sistema, prima di iniziare il ciclo di lavoro, e si evidenzia oltre che con il numero 0 anche con una doppia riquadratura.

Accanto al numero d'ordine della fase viene riportata, entro un apposito rettangolo, la descrizione dell'operazione corrispondente.

Tra un fase e la successiva si interpone un trattino orizzontale con l'indicazione dell'evento che permette il passaggio alla fase successiva (es. finecorsa).

(*) Agenzia nazionale per lo sviluppo della produzione automatizzata.

Introduzione

Utilità del formalismo:

- Rappresentazione schematica uniforme del comportamento funzionale di un automatismo (comunicazione priva di ambiguità)
- Ausilio alla progettazione
- Strumento per la realizzazione dei programmi per PLC
- Strumento operativo per la manutenzione

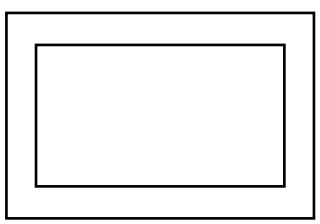
Elementi base del Grafcet

Passo (step)

Individua lo stato dell'automatismo
Ad esso sono associate le azioni

Transizione (transiction)

Individua le condizioni per il
cambiamento di stato dell'automatismo



Passo iniziale

Individua lo stato iniziale

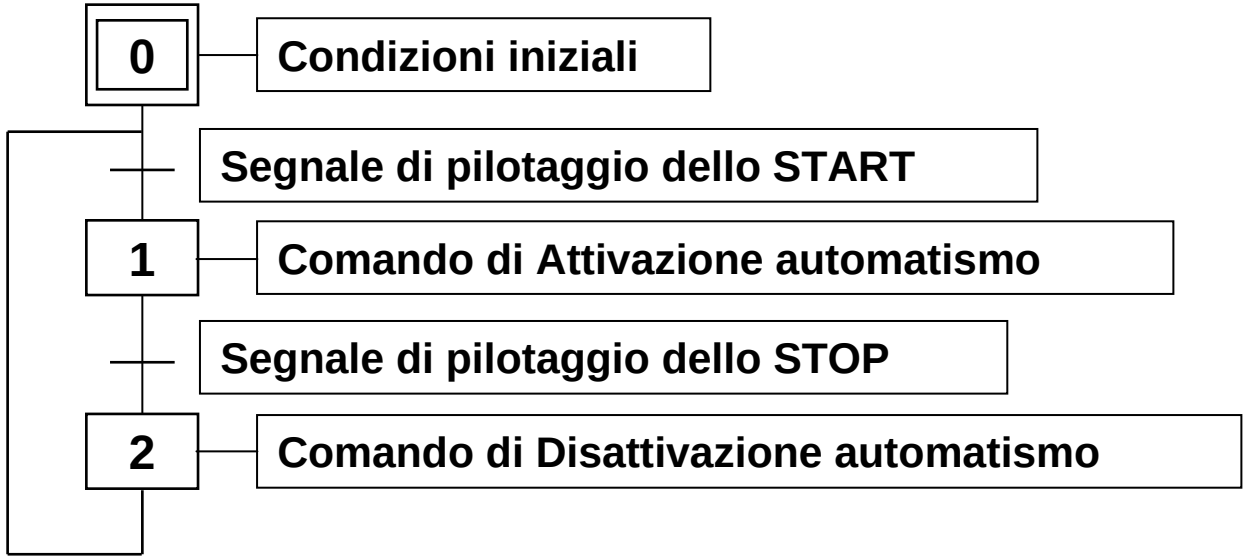
Grafcet descrittivo

Dato un automatismo così descritto:

- premendo lo *start* *l'automatismo* *parte*
- premendo lo *stop* *l'automatismo* si *arresta*

La schematizzazione Grafcet descrittiva è la seguente

Grafcet operativo



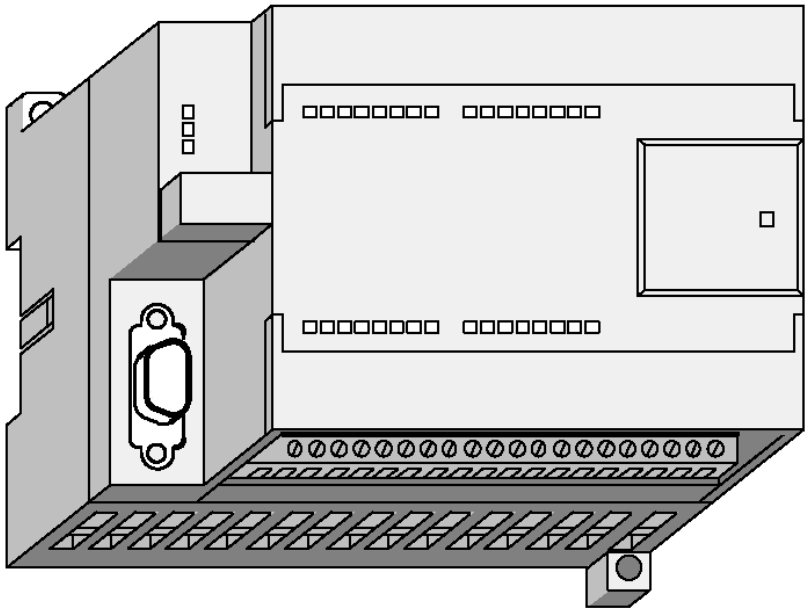
Micro PLC S7-200
SIEMENS

ISTRUZIONI ED ESERCITAZIONI

Micro PLC S7-200
SIEMENS

La serie S7-200 è una linea di controllori programmabili di dimensioni ridotte (Micro PLC).

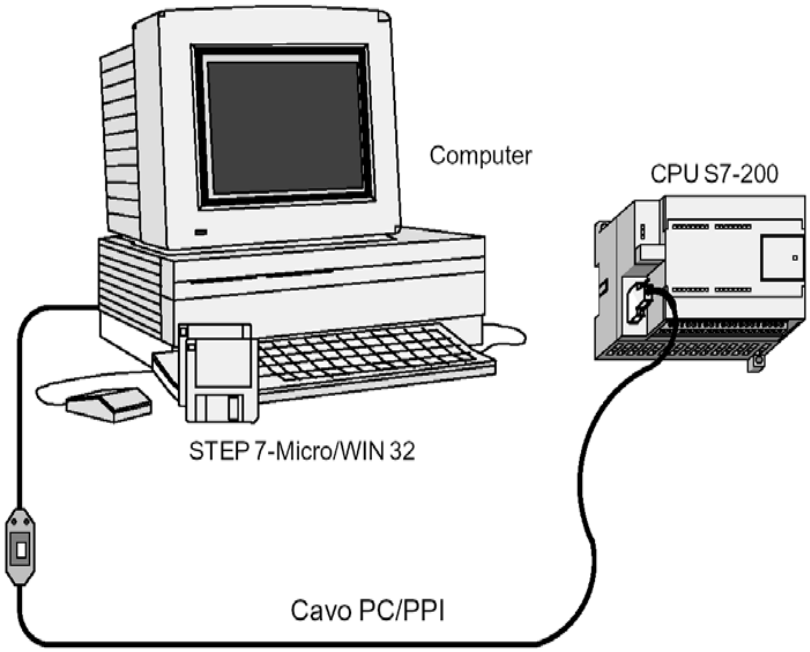
Un’ ampia gamma di modelli di CPU con diverse tensioni di alimentazione permette di disporre della flessibilità necessaria per affrontare e risolvere i problemi di automazione.



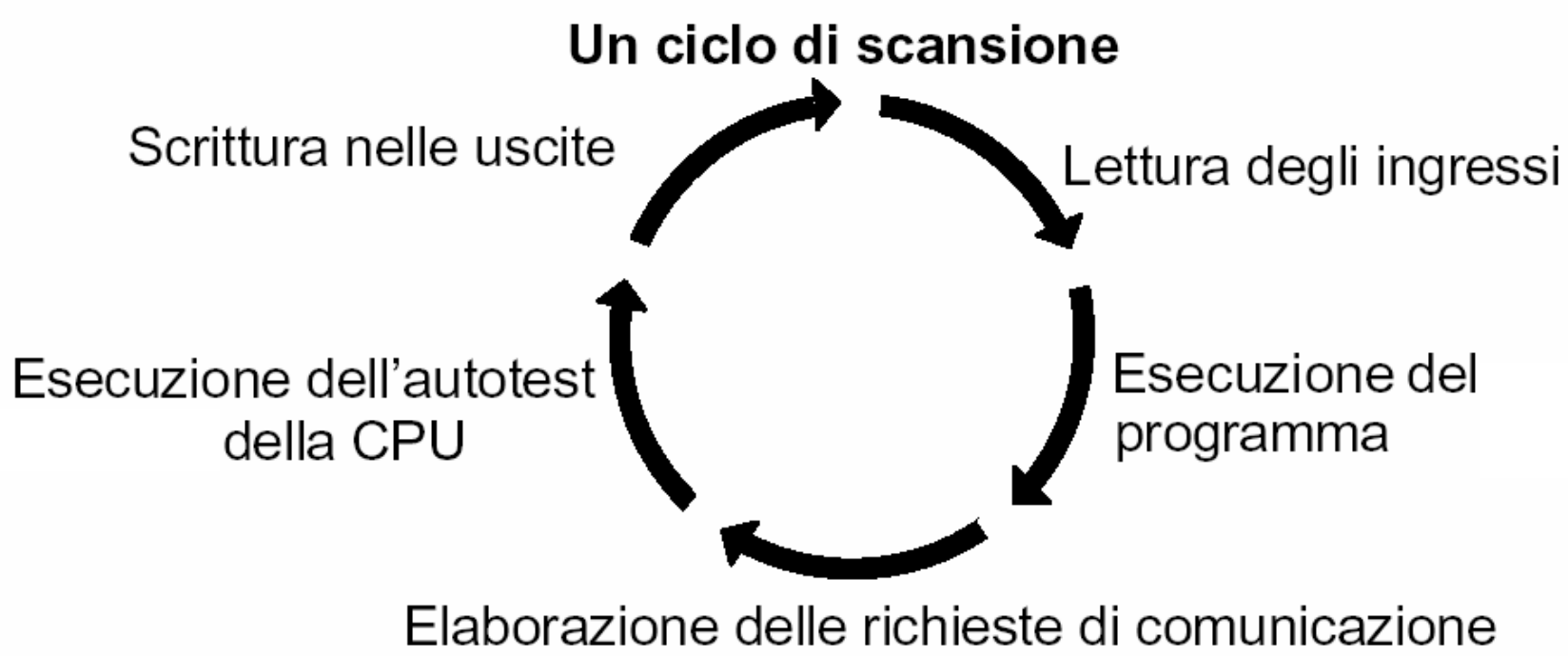
Requisiti hardware

Per poter usare un personal computer (PC) si deve disporre di uno dei seguenti set:

- cavo PC/PPI
- unità per processore di comunicazione (CP) e cavo per interfaccia multipoint (MPI)
- scheda per interfaccia multipoint (MPI). La scheda MPI viene fornita con un cavo MPI



Ciclo di scansione



Elenco delle CPU S7-200

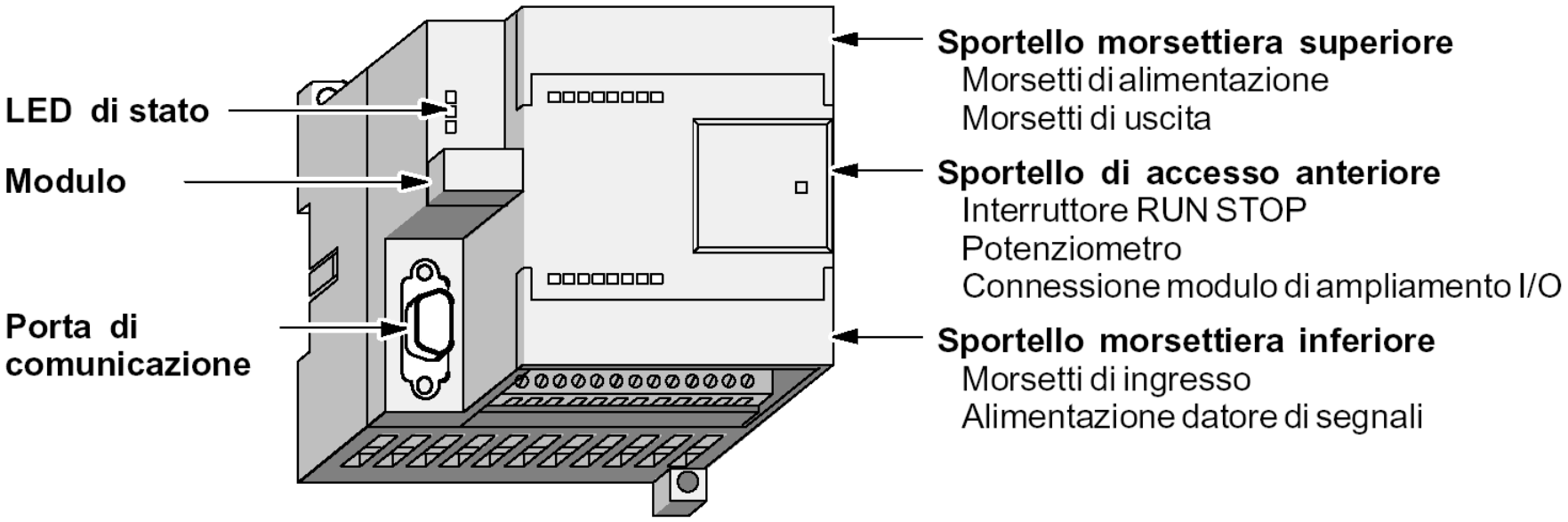
Funzioni	CPU 221	CPU 222	CPU 224
Dimensioni fisiche dell'unità	90 mm x 80 mm x 62 mm	90 mm x 80 mm x 62 mm	120,5 mm x 80 mm x 62 mm
Memoria			
Programma	2048 parole	2048 parole	4096 parole
Dati utente	1024 parole	1024 parole	2560 parole
Tipo di memoria	EEPROM	EEPROM	EEPROM
Moduli di memoria	EEPROM	EEPROM	EEPROM
Backup dei dati (condensatore ad elevata capacità)	Di reg. 50 ore	Di reg. 50 ore	Di reg. 190 ore
Ingressi/uscite integrati			
Ingressi/uscite integrati	6 ingressi / 4 uscite	8 ingressi / 6 uscite	14 ingressi / 10 uscite
Numero unità di ampliamento	nessuna	2 unità	7 unità
I/O totali			
Dimensione dell'immagine degli I/O digitali	256 (128 ingressi e 128 uscite)	256 (128 ingressi e 128 uscite)	256 (128 ingressi e 128 uscite)
Dimensioni fisiche degli I/O digitali	10	62	128
Dimensione dell'immagine degli I/O analogici	nessuno	16 ingressi / 16 uscite	16 ingressi / 16 uscite
Dimensioni fisiche dell'immagine degli I/O analogici	nessuno	12 ingressi / 10 uscite	12 ingressi / 10 uscite
Operazioni			
Velocità di esecuzione booleana	0,37 µs/operazione	0,37 µs/operazione	0,37 µs/operazione
Relè interni	256	256	256
Contatori / temporizzatori	256/256	256/256	256/256
Relè di controllo sequenziale	256	256	256
Loop For / Next	Sì	Sì	Sì
Operazioni matematiche con i numeri interi (+ - * /)	Sì	Sì	Sì
Operazioni matematiche con i numeri reali (+ - * /)	Sì	Sì	Sì

CPU S7-200

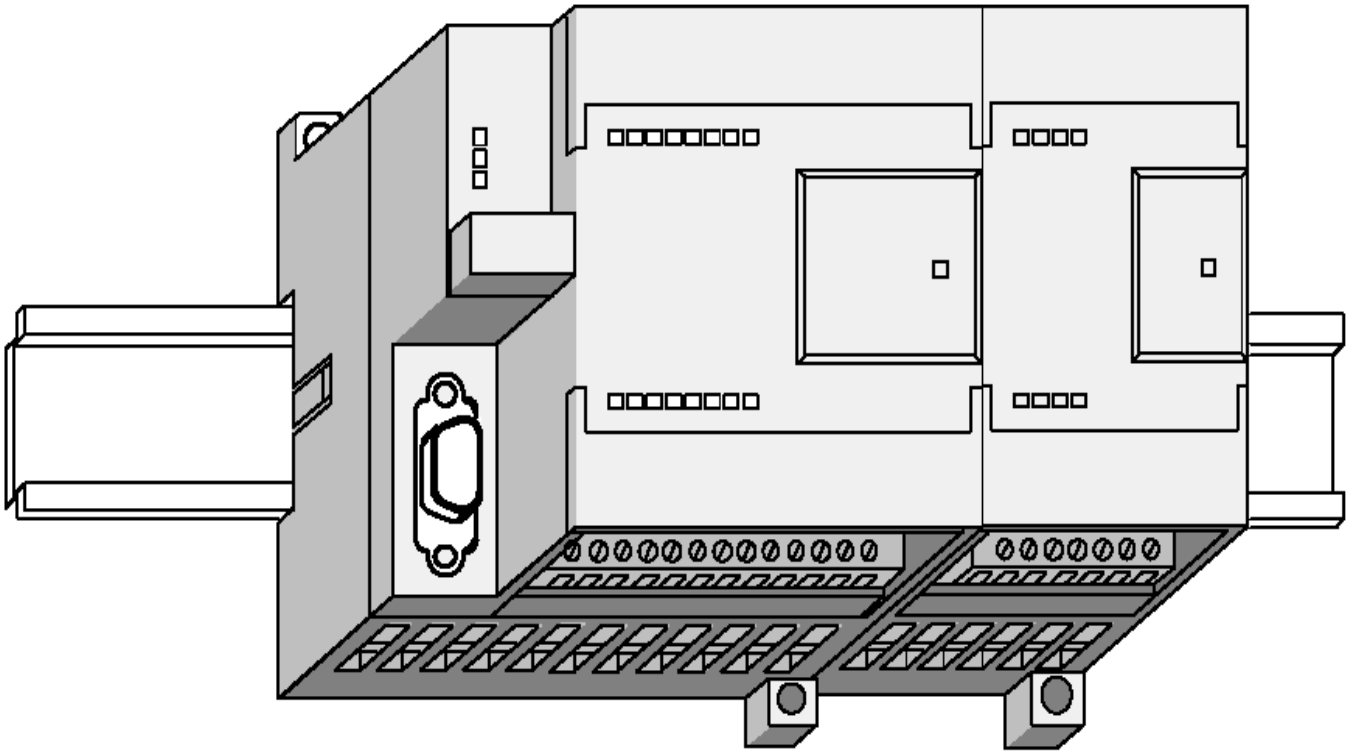
La CPU S7-200 riunisce l'unità centrale (CPU), l'alimentatore e gli ingressi e uscite digitali in un unico dispositivo autonomo e compatto

- La CPU esegue il programma e memorizza i dati per il controllo del compito di automazione e del processo
- L'alimentatore fornisce corrente all'unità di base e alle unità di ampliamento ad essa collegate
- Gli ingressi e le uscite costituiscono i punti di controllo del sistema
  - gli ingressi controllano i segnali provenienti dai dispositivi di campo (quali i sensori e gli interruttori); le uscite controllano pompe, motori o altri dispositivi del processo
- L'interfaccia di comunicazione permette di collegare la CPU ad un dispositivo di programmazione o ad altri dispositivi
- I LED di stato forniscono informazioni sullo stato di funzionamento della CPU (RUN o STOP), lo stato attuale degli I/O locali e gli errori di sistema rilevati
- Un modulo EEPROM seriale plug-in consente di memorizzare i programmi della CPU

CPU S7-200


LED di stato
Modulo
Porta di comunicazione
Sportello morsettiera superiore
Morsetti di alimentazione
Morsetti di uscita
Sportello di accesso anteriore
Interruttore RUN STOP
Potenziometro
Connessione modulo di ampliamento I/O
Sportello morsettiera inferiore
Morsetti di ingresso
Alimentazione datore di segnali

Unità di ampliamento



Configurazione di un sistema di automazione S7-200

PC

- Su cui è stato installato STEP 7-Micro/WIN 32
- In ambiente Windows (95, 98, Windows NT 4.0)
- Almeno 16 MB di RAM e 50 MB di HD

CPU S7-200

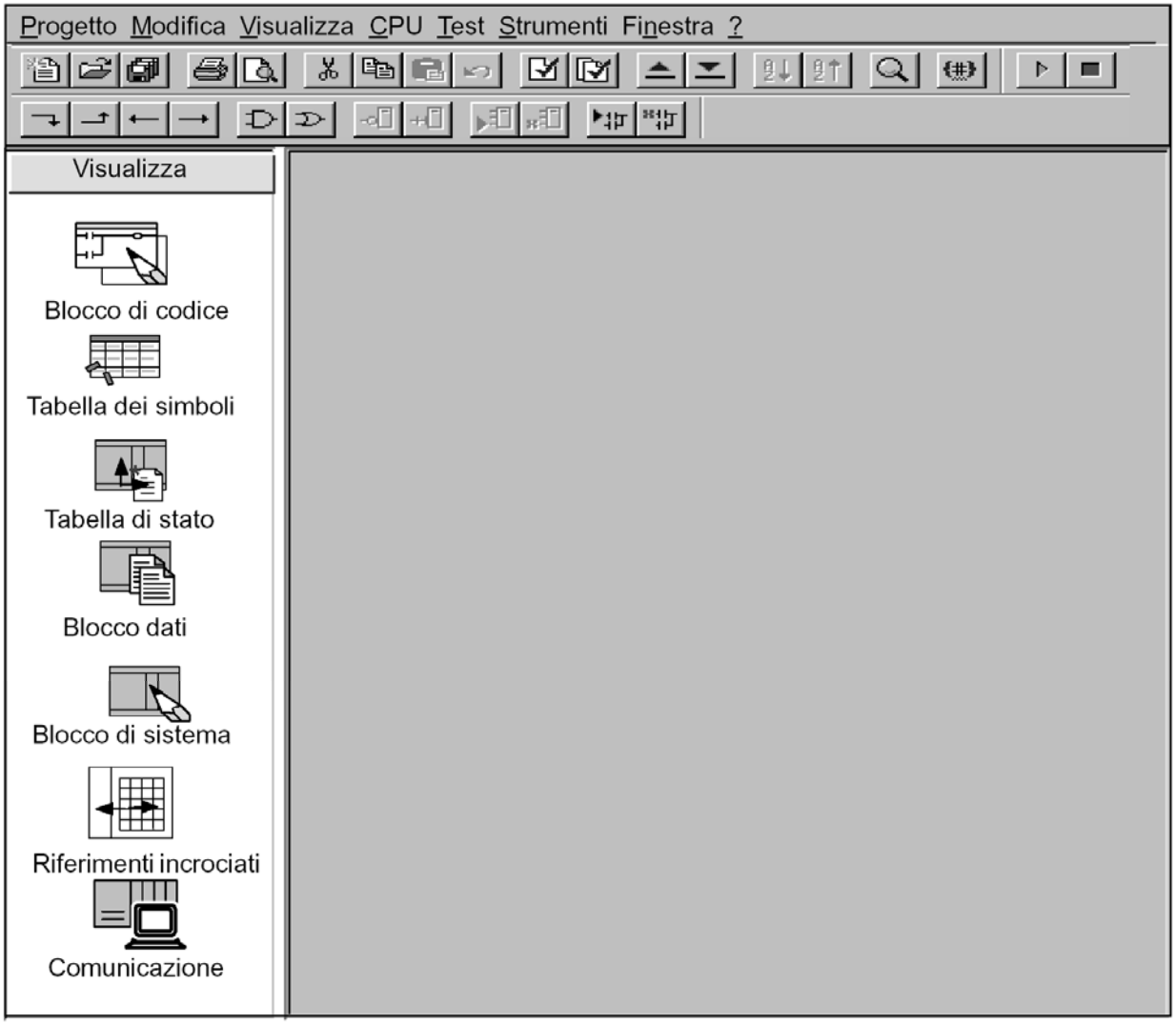
- cavo di connessione PC/PPI
- CPU 221, CPU 222, CPU 224

Errori di installazione

L’installazione potrebbe non riuscire per le seguenti ragioni

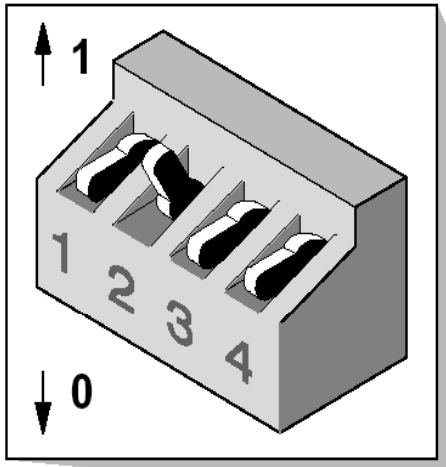
- Memoria non sufficiente: occorrono almeno 50 MB di spazio libero di memoria sul disco fisso.
- Dischetto difettoso: verificare la qualità del dischetto, ed in caso negativo rivolgersi al proprio rappresentante o al distributore
- Errore dell’operatore: ricominciare da capo l’installazione dopo aver letto attentamente le istruzioni
- Qualche applicazione non è stata chiusa: potrebbe trattarsi anche della barra degli strumenti di Microsoft Office

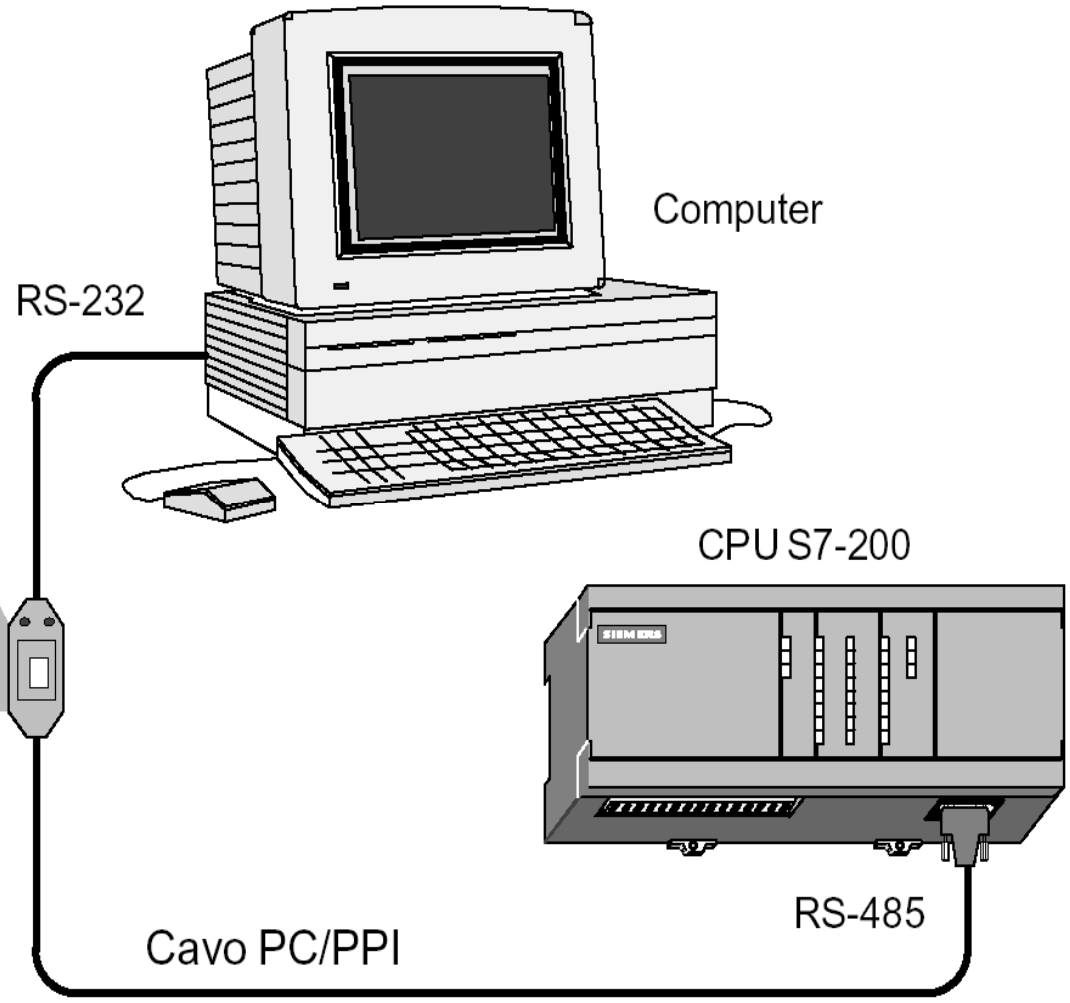
Menu visualizza di STEP 7-Micro/WIN 32



Comunicazione con la CPU in modo PPI

Impostazionidei microinterruttori
(verso il basso = 0, verso l'alto = 1):
0 1 0 0 = 9600 baud (indicati)
0 0 1 0 = 19200 baud





Problemi più frequenti

La comunicazione potrebbe non riuscire per le seguenti ragioni:

- velocità di trasmissione errata: correggere la velocità di trasmissione
- indirizzo di stazione errato: correggere l'indirizzo della stazione
- cavo PC/PPI non impostato correttamente: controllare le impostazioni dei microinterruttori sul cavo PC/PPI
- interfaccia di comunicazione errata nel personal computer: controllare l'interfaccia
- CPU in modo freeport (interfaccia di comunicazione controllata dal programma utente): portare la CPU in STOP
- conflitto con altri master: scollegare la CPU dalla rete

Creazione di un programma in KOP

La finestra dell'editor in schema a contatti consente all'utente di scrivere un programma utilizzando i simboli grafici del linguaggio KOP

La barra degli strumenti comprende alcuni degli elementi KOP maggiormente utilizzati per introdurre i programmi utente

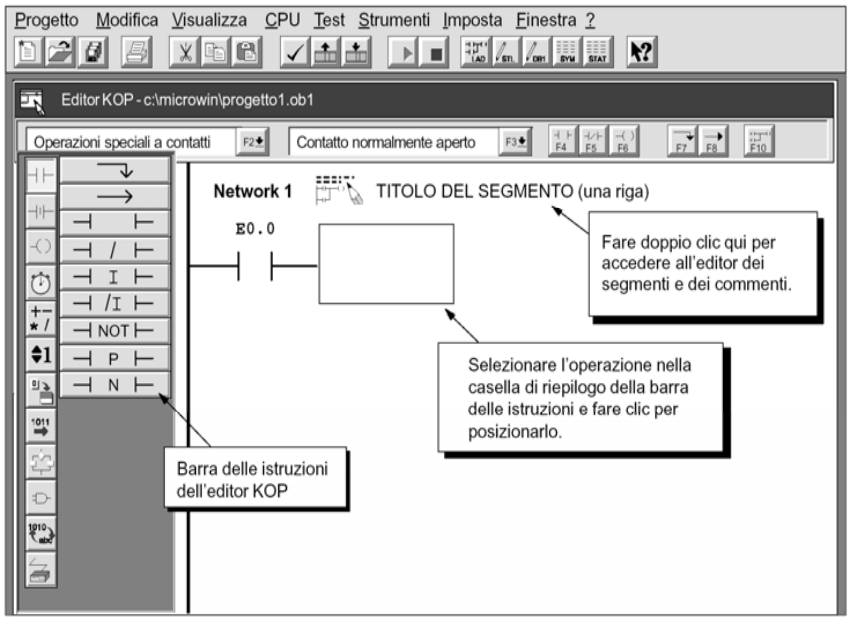
La prima casella di riepilogo a discesa (quella a sinistra) contiene le istruzioni raggruppate per categorie

Premere o fare clic su F2 per accedere a tali categorie

Dopo aver selezionato una categoria, la seconda casella di riepilogo a discesa (quella a destra) visualizzerà tutte le operazioni comprese nella stessa

È anche possibile vedere una lista di tutte le operazioni in ordine alfabetico premendo F9 e selezionando tutte le operazioni

In alternativa, selezionare Visualizza Barra istruzioni per visualizzare la barra delle istruzioni KOP



Immissione del programma

Ad ogni segmento (Network) sono associati i commenti/titolo a una sola riga o i commenti a più righe

Per introdurre il programma utente eseguire i passi seguenti

1. Selezionare il comando del menu **Modifica Titolo del programma** per introdurre un titolo di programma. Immettere il titolo del programma e fare clic sul pulsante “OK”
2. Per introdurre gli elementi KOP, selezionare il tipo di elemento desiderato facendo clic sul corrispondente pulsante ad icona o scegliendo dalla lista di istruzioni
3. Digitare l’indirizzo o il parametro in ogni casella di testo e premere INVIO

Immissione del programma in AWL

L'editor della lista istruzioni (AWL) è un editor di testo in forma libera che consente un discreto grado di flessibilità nel modo in cui si sceglie di introdurre le operazioni del programma.

Editor AWL - progetto1.ob1

```
//Programma per trasportatore

NETWORK 1       //Avvio motore:
LD      "Start1"        //Se I0.0 è attivo (on)
AN      "Stop_emerg1"   //e I0.1 non è attivo,
=       Q0.0            //inserisce motore di trasporta

Network 2       //E-arresta il trasportatore:
LD      I0.1            //Se Stop_emerg1 è attivato
O       I0.3            //o se Stop_emerg2 è attivato,
R       Q0.0, 1         //disinserisce motore di trasportatore.

NETWORK 3       //Fine del programma
MEND
```

Per visualizzare il programma in KOP, occorre dividere i segmenti del codice con la parola chiave NETWORK.

Programma in lista istruzioni

Per poter visualizzare un programma AWL in KOP occorre suddividere il codice in segmenti distinti introducendo la parola chiave NETWORK. (I numeri di segmento vengono generati automaticamente dopo che si è compilato o caricato nella CPU il programma). Le dichiarazioni dei segmenti non devono superare i limiti adatti alla rappresentazione KOP

Iniziare ogni commento con due barrette oblique (//). Ogni riga di commento addizionale deve quindi iniziare con le barrette oblique (double slash)

Terminare ogni riga con un ritorno a capo

Separare ogni operazione dal suo indirizzo o parametro tramite uno spazio o TAB

Non utilizzare spazi tra il tipo di operando e l'indirizzo (ad esempio, scrivere **I0.0** e non **I 0.0**)

Separare ogni operando all'interno di una operazione tramite virgola, spazio o TAB

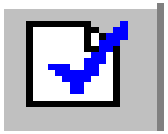
Utilizzare le virgolette per introdurre i nomi dei simboli. Se per esempio la tabella dei simboli contiene il nome simbolico Start_1 per l'indirizzo **I0.0**, introdurre l'operazione nel seguente modo:

LD "Start1"

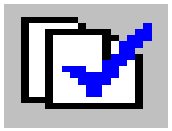
Compilazione del programma

Dopo aver completato un segmento o una serie di segmenti si può verificare la sintassi del codice introdotto selezionando il comando del menu **CPU Compila**, o facendo clic sul pulsante di compilazione:

- "Compila" consente di compilare un singolo elemento del progetto. La finestra (editor di programma o di blocco dati) che viene posta in primo piano quando si sceglie "Compila" viene compilata, mentre le altre due finestre non vengono ancora compilate
- "Compila tutto" consente di compilare l'editor di programma, il blocco di sistema e il blocco dati. Se si sceglie questo comando, non ha più alcuna rilevanza quale finestra sia in primo piano



Compila

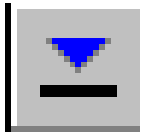


Compila tutto

Carica nella CPU

Per caricare un componente del progetto da STEP 7-Micro/WIN 32 nella CPU:

- Fare clic sul pulsante Carica nella CPU
- Selezionare il comando di menu File > Carica nella CPU
- Premere la combinazione di tasti CTRL+D

Carica nella CPU 

Carica nel PG

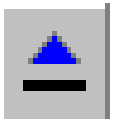
Per caricare i componenti del progetto dalla CPU in un editor di programma di STEP 7-Micro/WIN 32:

- fare clic sul pulsante Carica nel PG
- Selezionare il comando di menu File > Carica nel PG.
- Premere la combinazione di tasti Ctrl+G

Per caricare il programma nel PG/PC (dalla CPU nell'editor) è necessario che la comunicazione con la CPU funzioni correttamente. Accertarsi che il cavo di collegamento tra l'hardware di rete e la CPU sia funzionante

Selezionare i blocchi desiderati (blocco di programma, blocco dati o blocco di sistema)

I componenti del programma selezionati vengono copiati dalla CPU nel progetto attualmente aperto. L'utente potrà quindi salvare il programma caricato nel PG

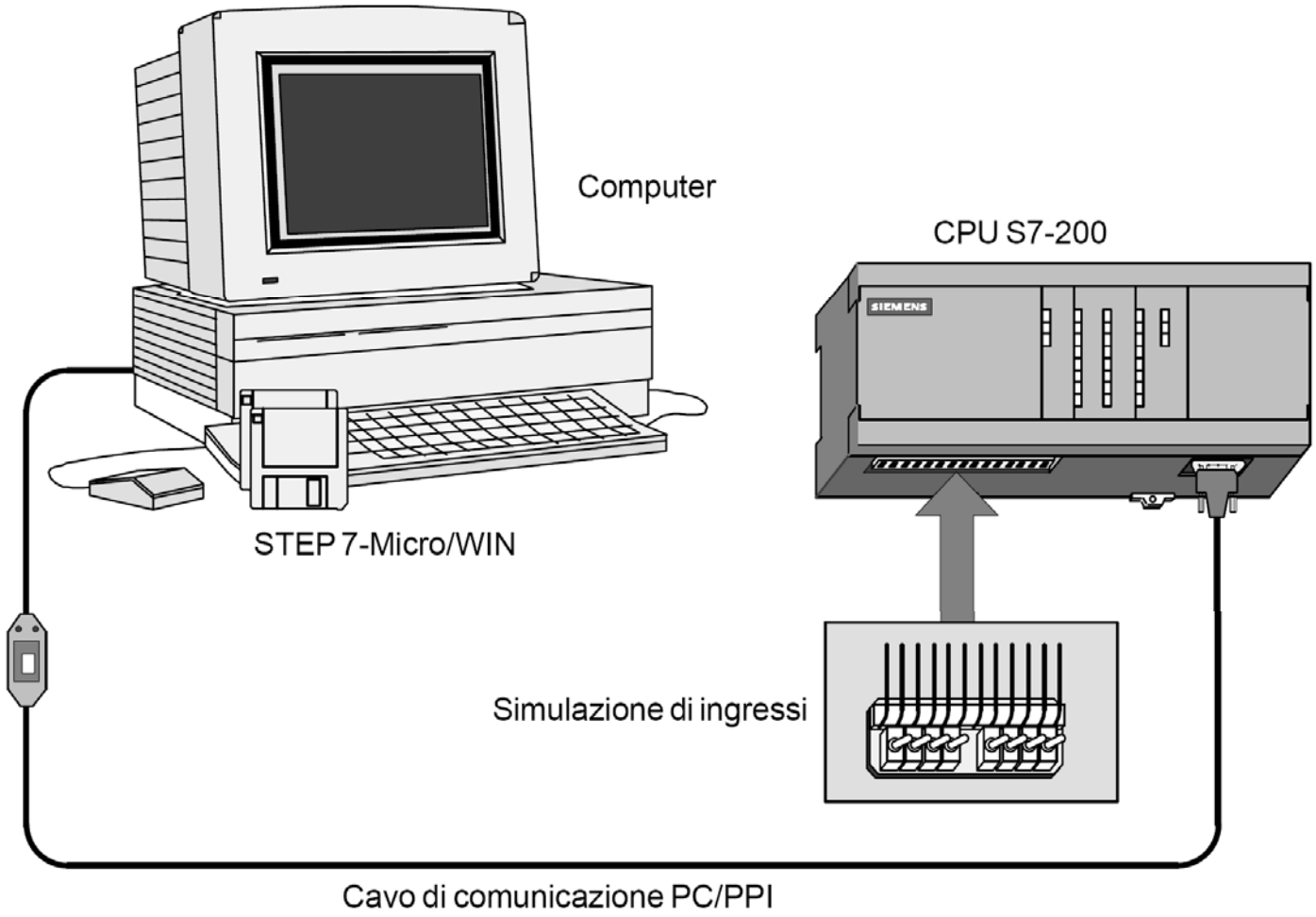
Carica nel PG 

Primo esempio di creazione di un programma

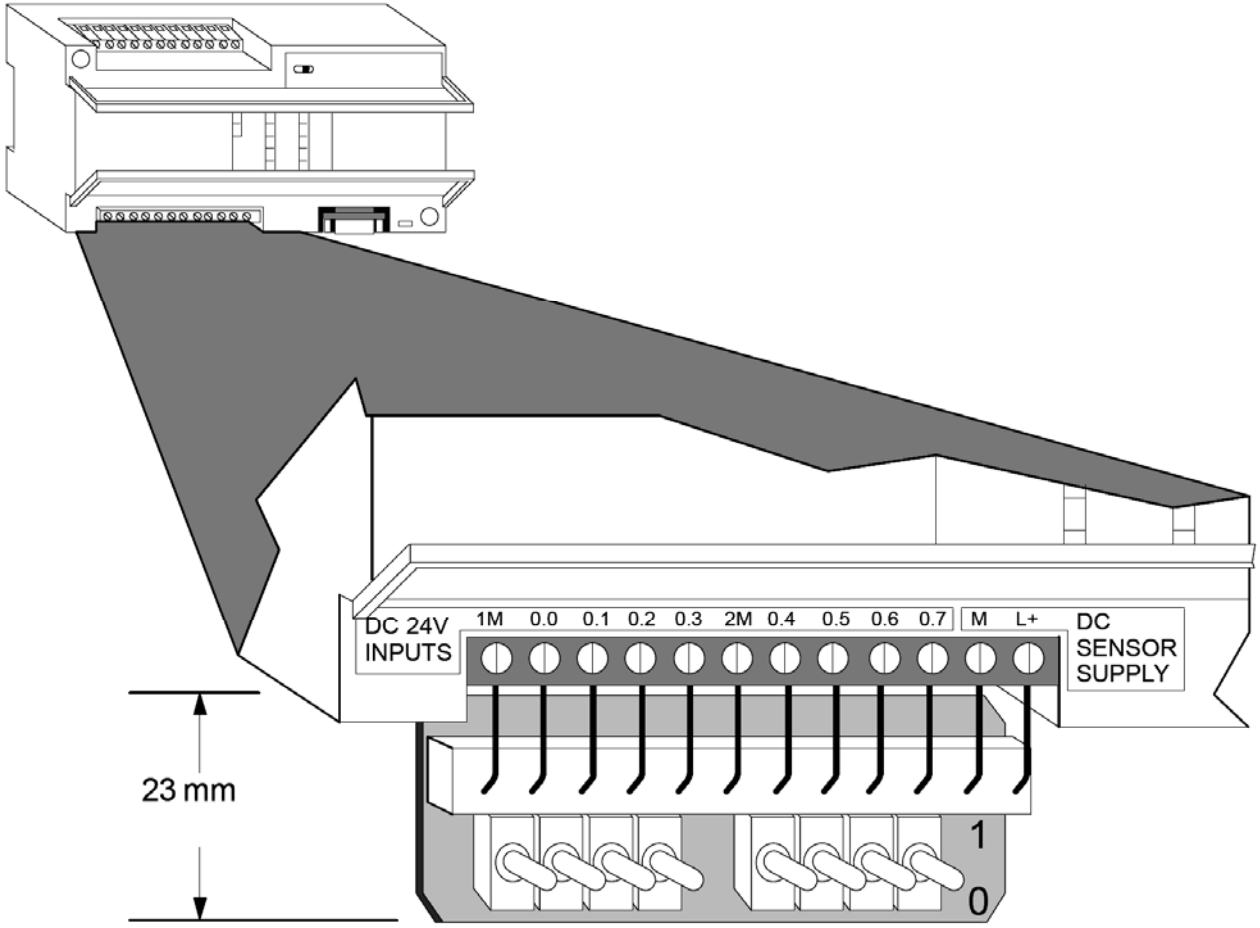
Requisiti di sistema

- Cavo di programmazione PC/PPI o scheda MPI installata nel computer; cavo RS-485 per il collegamento alla CPU S7-200
- CPU S7-200
- Cavo di corrente e alimentazione
- STEP 7-Micro/WIN per Windows

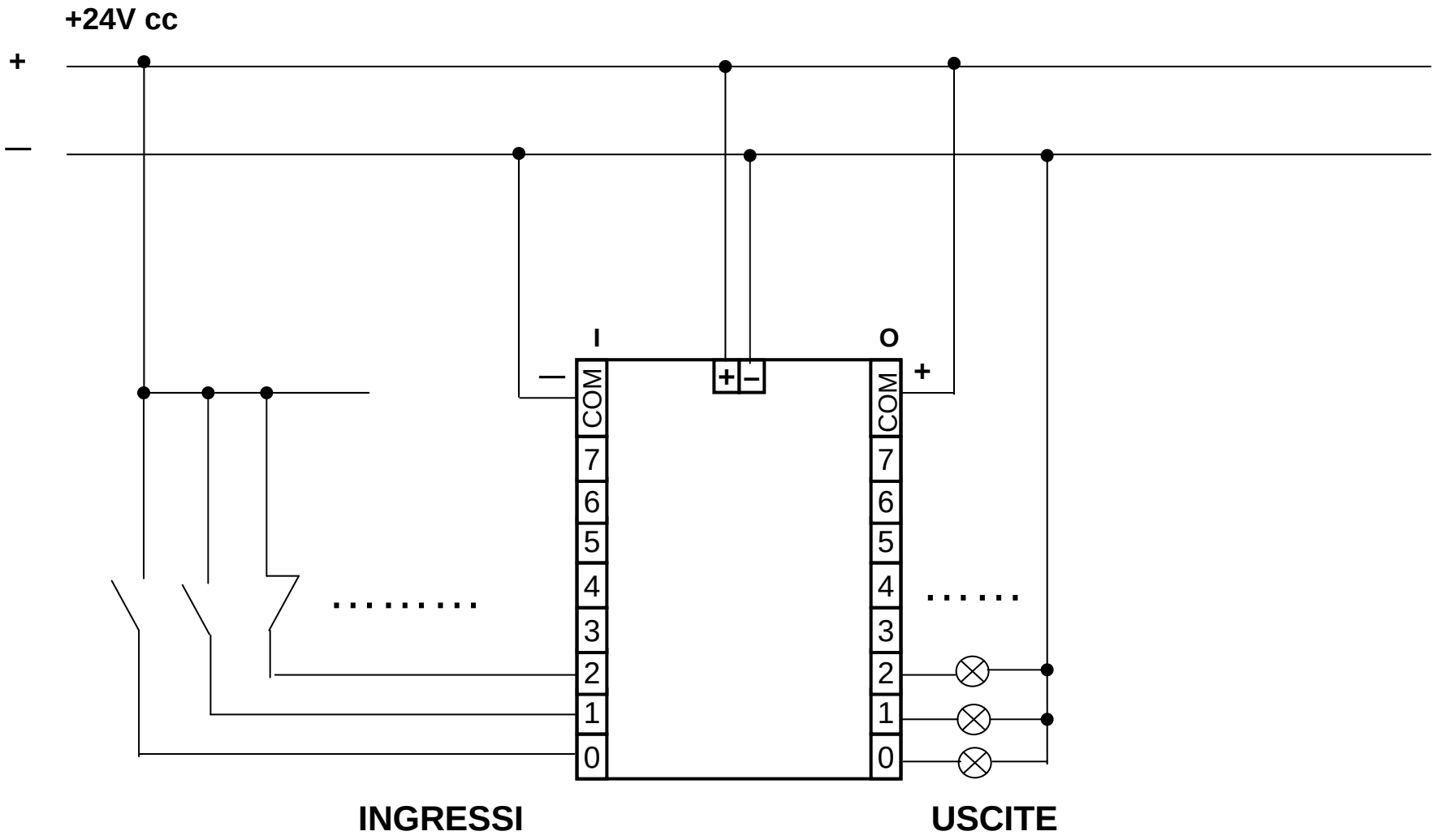
Requisiti per eseguire il programma di esempio



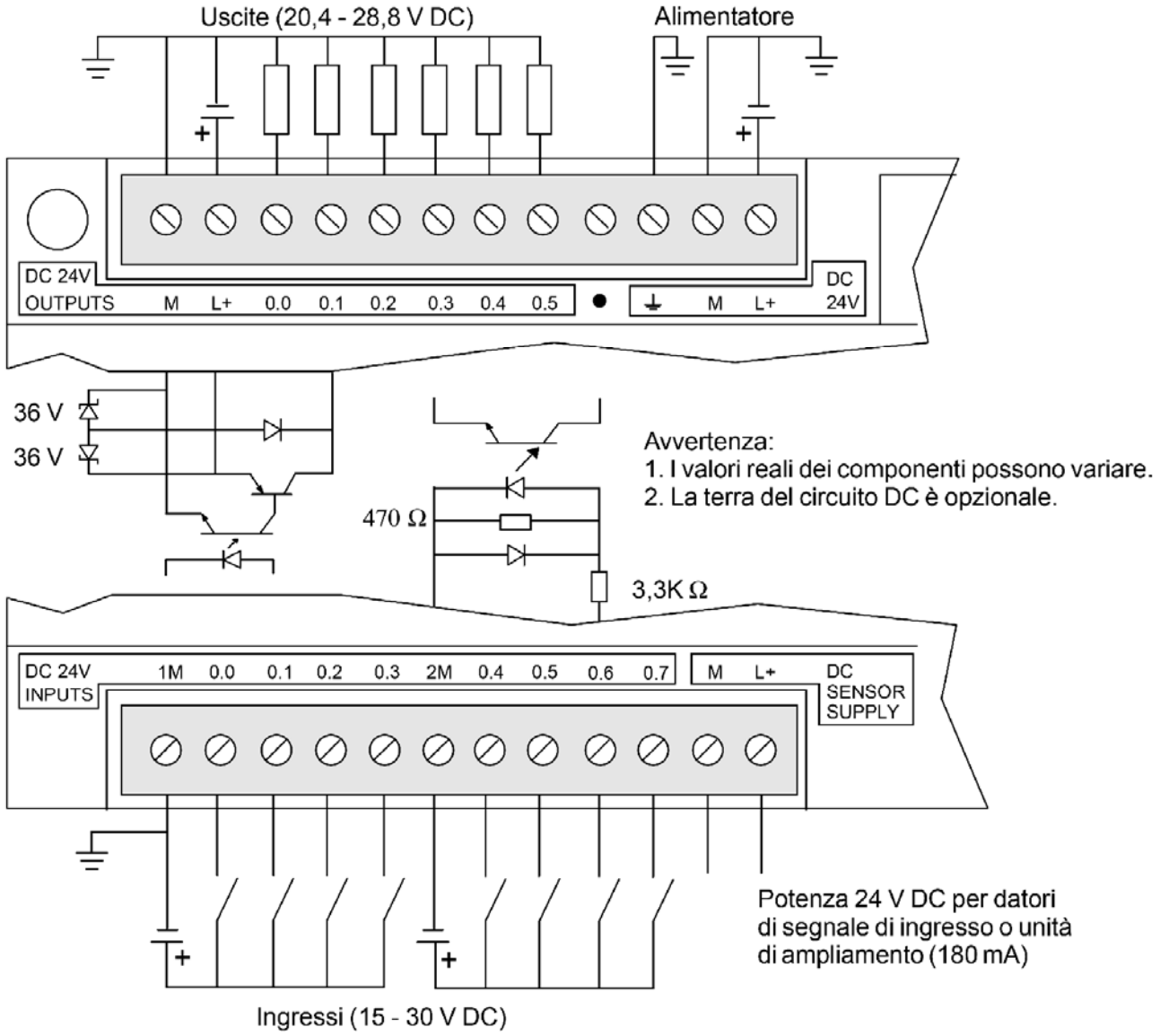
Simulatore di ingressi DC per la CPU 212



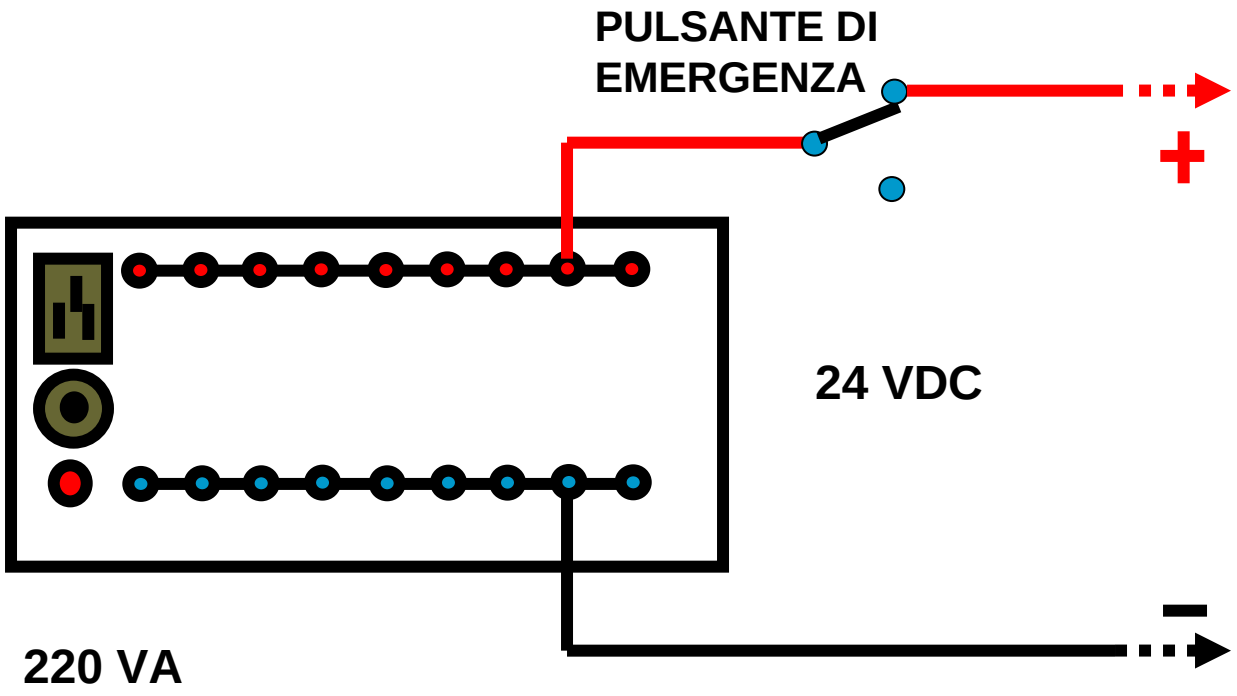
Cablaggio PLC



Collegamenti della CPU 212 DC/DC/DC

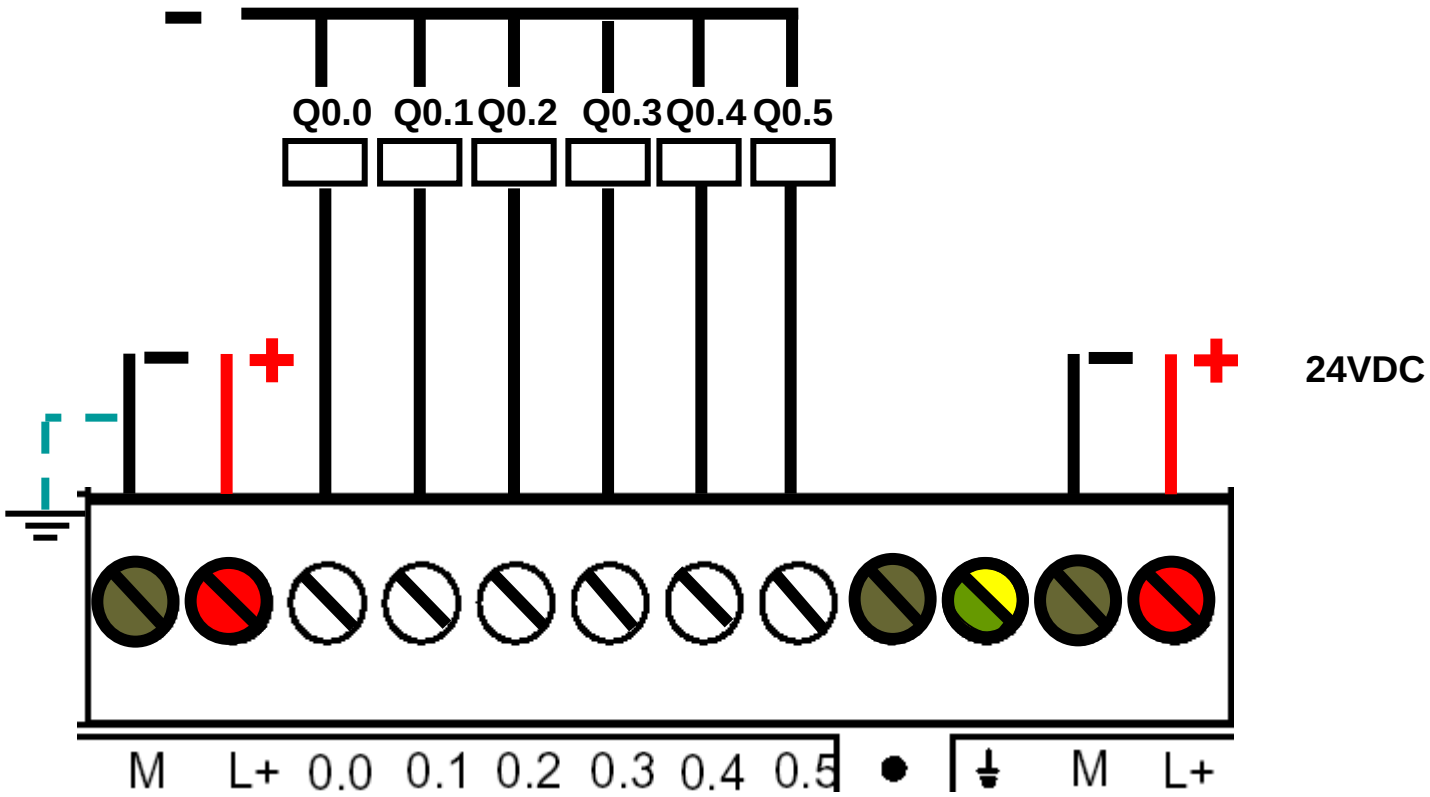


Morsettiera PLC con CPU 212 24 VDC

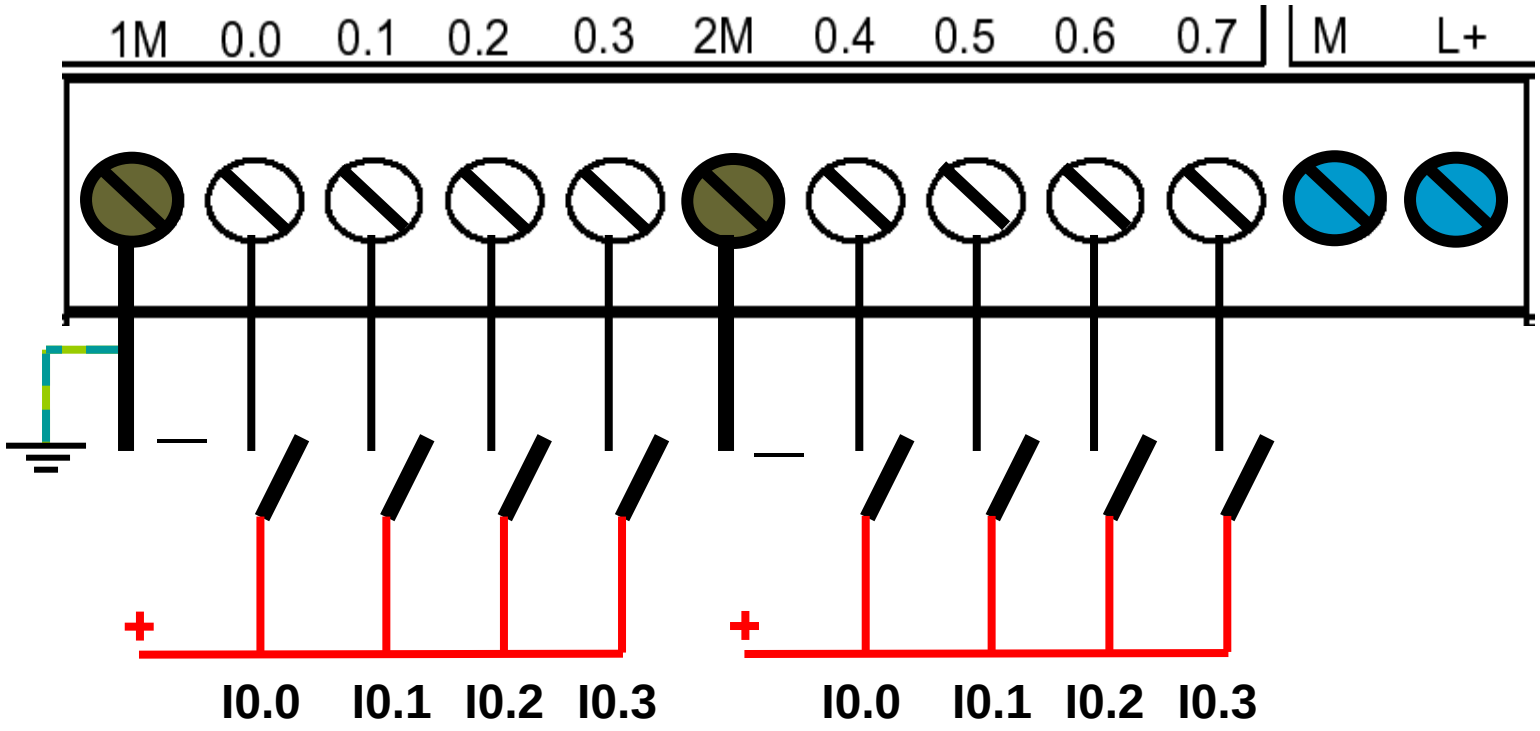


NOTA: **L’alimentazione dei sensori e del PLC viene prelevata all’uscita del pulsante di EMERGENZA**

Uscite



Ingressi



LE ISTRUZIONI

Accesso alla memoria dati

La memoria dati è costituita da cinque aree:

I	Ingresso
Q	Uscita
M	Merker interno
SM	Merker speciale
S	Memoria S
V	Memoria variabile

Per utilizzare un indirizzo di memoria si deve comporre l’indirizzo utilizzando il tipo di memoria.

Accesso a Bit, Byte, Parola e Doppia parola

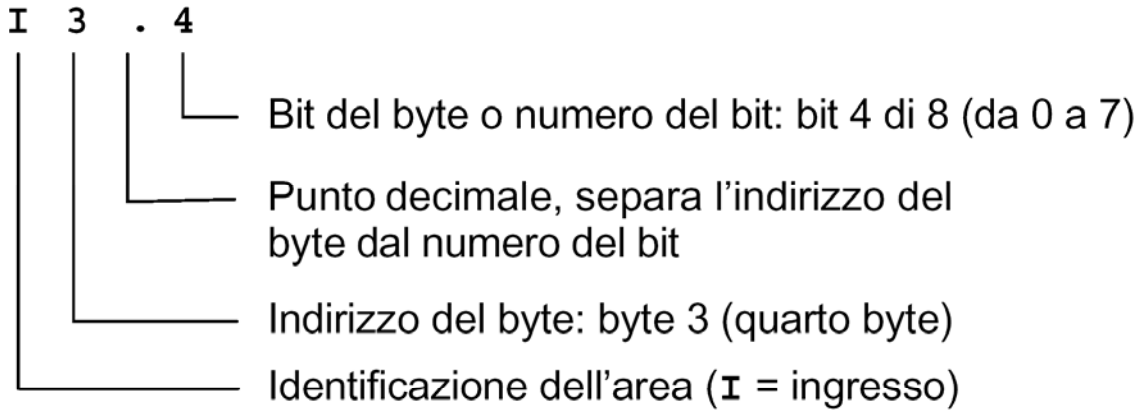
Per accedere a un bit va specificato l'indirizzo del bit con l'indicazione di area e numero di bit separati da un punto.

Es. I0.0 dove zero è il primo indirizzo di tutte le aree e l'indirizzo di bit va da 0 a 7.

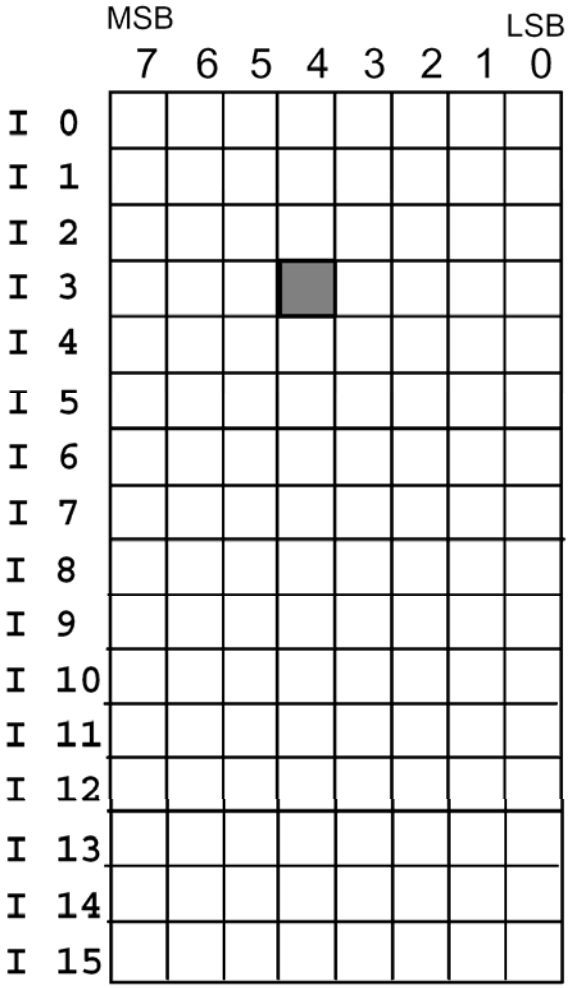
Per accedere a byte, parole e doppie parole, si deve specificare l'area (es. V), la dimensione dei dati (es. B per byte, W per parola, D per doppia parola) e il numero di indirizzo:

- VB 200 accede al byte 200 di un indirizzo della memoria variabile
- VW 200 accede ai byte 200 e 201 di un indirizzo della memoria variabile
- VD 200 accede ai byte 200, 201, 202, 203 di un indirizzo della memoria variabile

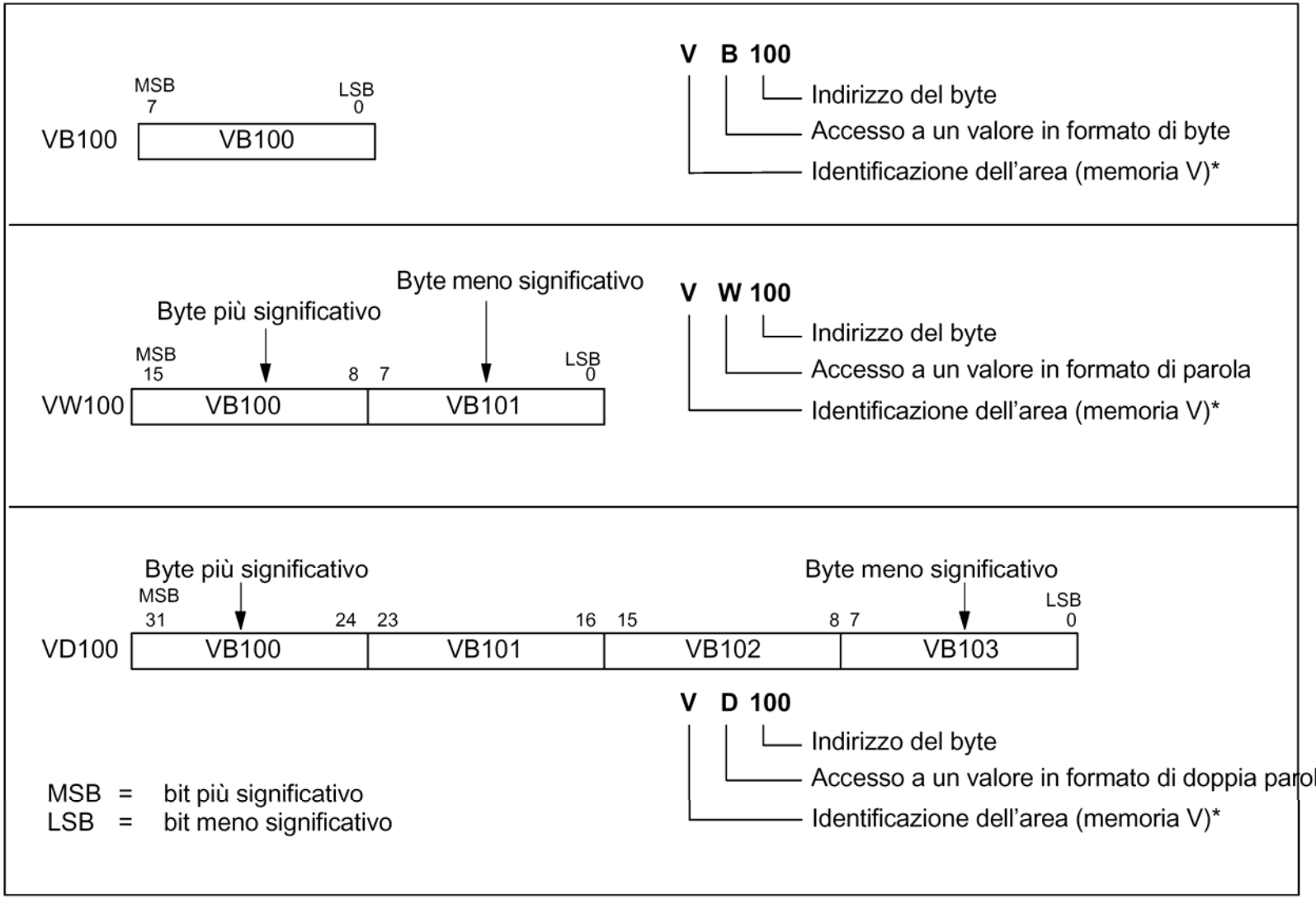
Accesso a un bit di dati nella memoria CPU



MSB = bit più significativo
LSB = bit meno significativo



Indirizzamento diretto nelle aree di memoria della CPU



Rappresentazione dei numeri

Dimensione dati e campi di numeri interi associati

Dimensione dei dati	Campo interi senza segno		Campo interi con segno	
	Decimale	Esadecimale	Decimale	Esadecimale
B (byte): valore a 8 bit	da 0 a 255	da 0 a FF	da -128 a 127	da 80 a 7F
W (parola): valore a 16 bit	da 0 a 65,535	da 0 a FFFF	da -32.768 a 32.767	da 8000 a 7FFF
D (doppia parola, Dparola): valore a 32 bit	da 0 a 4.294.967.295	da 0 a FFFF FFFF	da -2.147.483.648 a 2.147.483.647	da 8000 0000 a 7FFF FFFF

Indirizzamento

Indirizzamento del registro delle immagini di processo degli ingressi (I)

Come già descritto al paragrafo 6.5, la CPU campiona i punti di ingresso fisici all'inizio di ogni ciclo di scansione, e scrive questi valori nel registro delle immagini di processo degli ingressi. Si potrà accedere a tale registro in bit, byte, parola e doppia parola.

Formato:

Bit	I [indirizzo byte].[indirizzo bit]	I0.1
Byte, parola, doppia parola	I [dimensione][indirizzo byte iniziale]	IB4

Indirizzamento del registro delle immagini di processo delle uscite (Q)

Alla fine del ciclo di scansione la CPU copia nelle uscite fisiche i valori memorizzati nell'immagine di processo delle uscite. Si potrà accedere a tale registro in bit, byte, parola e doppia parola.

Formato:

Bit	Q [indirizzo byte].[indirizzo bit]	Q1.1
Byte, parola, doppia parola	Q [dimensione][indirizzo byte iniziale]	QB5

Indirizzamento

Indirizzamento dell’area di memoria variabile (V)

La memoria V può essere utilizzata per memorizzare i risultati intermedi di operazioni che vengono eseguite dalla logica di controllo del programma utente. Si potrà inoltre usare la memoria V per memorizzare altri dati pertinenti al processo o al compito di interesse dell’utente. Si potrà accedere all’area di memoria V in bit, byte, parola e doppia parola.

Formato:

Bit	V [indirizzo byte].[indirizzo bit]	V10.2
Byte, parola, doppia parola	V [dimensione][indirizzo byte iniziale]	VW100

Indirizzamento dell’area di merker (M)

I bit di merker interni (memoria M) possono essere utilizzati come relé di controllo per memorizzare lo stato intermedio di una operazione o altre informazioni di controllo. Ai merker interni si può accedere non solo in bit, ma anche in byte, parola e doppia parola.

Formato:

Bit	M [indirizzo byte].[indirizzo bit]	M26.7
Byte, parola, doppia parola	M [dimensione][indirizzo byte iniziale]	MD20

Indirizzamento

Indirizzamento dell’area di memoria del relè di controllo sequenziale (S)

I bit di relé di controllo sequenziale (S) sono utilizzati per organizzare il funzionamento di un impianto in sequenze o in segmenti di programma equivalenti. Gli SCR permettono la segmentazione logica del programma di controllo. Si può accedere ai bit S in formato bit, byte, parola e doppia parola.

Formato:

Bit	S [indirizzo byte].[indirizzo bit]	S3.1
Byte, parola, doppia parola	S [dimensione][indirizzo byte iniziale]	SB4

Indirizzamento dei merker speciali (SM)

I bit SM forniscono una possibilità di comunicare informazioni tra la CPU e il programma utente. Si potrà utilizzare tali bit per selezionare e controllare alcune delle funzioni speciali della CPU S7-200, come ad esempio le seguenti:

- un bit che si attiva solo per il primo ciclo
- bit che si attivano e disattivano a frequenze stabili
- bit che visualizzano lo stato delle funzioni matematiche o di altre operazioni

Formato:

Bit	SM[indirizzo byte].[indirizzo bit]	SM0.1
Byte, parola, doppia parola	SM[dimensione][indirizzo byte iniziale]	SMB86

Indirizzamento

Indirizzamento dell'area di memoria dei temporizzatori (T)

Nella CPU S7-200 i contatori sono elementi che contano gli incrementi di tempo. I temporizzatori S7-200 hanno risoluzioni (incrementi su base tempo) di 1 ms, 10 ms e 100 ms. Le seguenti due variabili sono associate al temporizzatore.

Valore corrente:	numero intero con segno a 16 bit che memorizza l'ammontare di tempo conteggiato dal temporizzatore.
Bit di temporizzazione:	questo bit si attiva (è impostato su 1) se il valore corrente del temporizzatore è maggiore o uguale al valore di default. (Il valore di default viene immesso come parte integrante dell'operazione di temporizzazione).

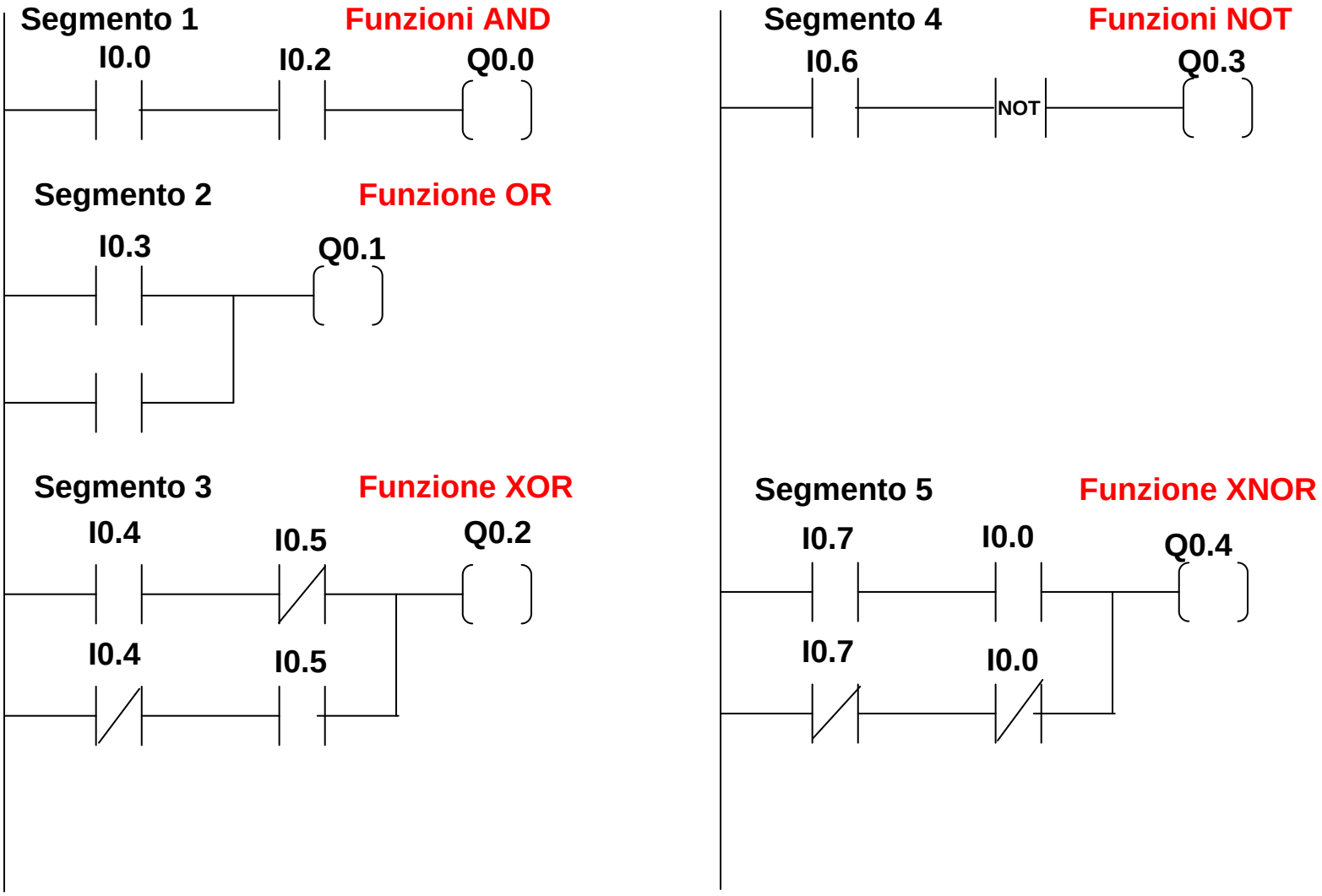
Si può accedere a entrambe queste variabili utilizzando l'indirizzo del temporizzatore (T + numero temporizzatore). L'accesso al bit di temporizzazione o al valore corrente dipende dall'istruzione adoperata: quelle con operandi in bit accedono al bit di temporizzazione; quelle con gli operandi in parola accedono al valore corrente.

Formato:	T [Numero temporizzatore]	T24

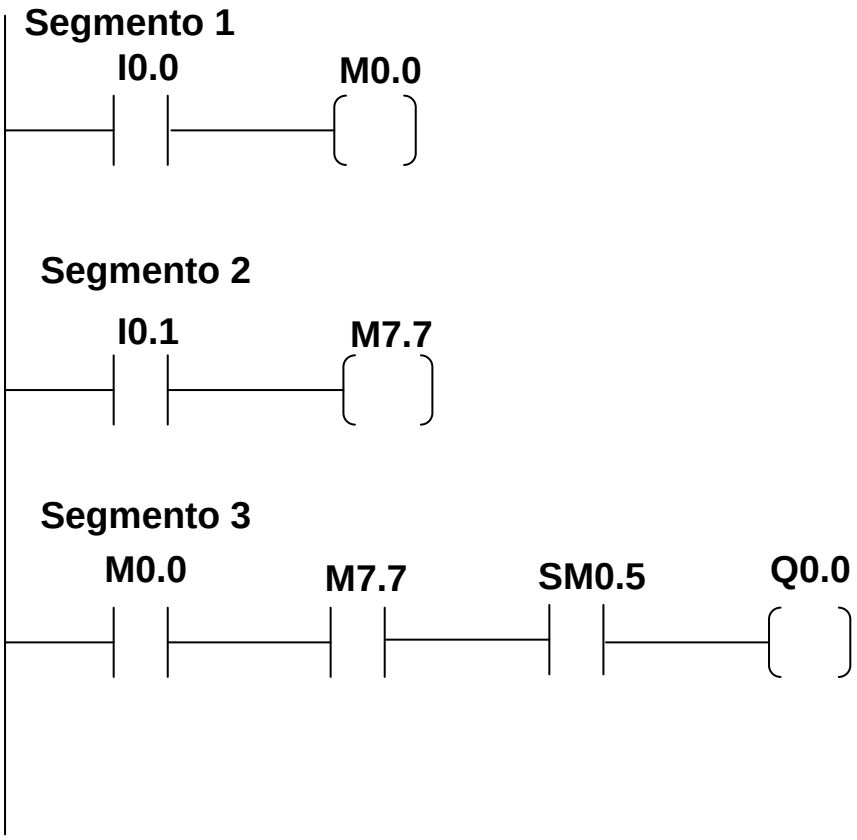
Bit di Stato aggiornati alla fine di ogni ciclo di scansione

SM0.0	Questo bit è sempre ON.
SM0.1	Questo bit è sempre ON per il primo ciclo di scansione. Viene utilizzato, ad esempio, per richiamare un sottoprogramma di inizializzazione.
SM0.2	Questo bit è on per 1 ciclo di scansione in caso di perdita dei dati a ritenzione. Può essere utilizzato come merker di errore o come meccanismo per richiamare una speciale sequenza di avvio.
SM0.3	Questo bit viene attivato per un ciclo se si passa allo stato RUN da una condizione di avvio. Può essere utilizzato per fornire un tempo di riscaldamento (warm-up) del sistema prima di avviare delle operazioni.
SM0.4	Questo bit mette a disposizione un impulso di clock di 60 secondi (on per 30 secondi, off per altri 30). Viene così fornito un ritardo facile da programmare o un impulso di clock di un minuto.
SM0.5	Questo bit mette a disposizione un impulso di clock di 1 secondo (on per 0,5 secondi, off per altri 0,5 secondi). Viene così fornito un tempo di ritardo facile da programmare o un impulso di clock di un secondo.
SM0.6	Questo bit è un clock di ciclo di scansione che è attivo per un ciclo e disattivato per il ciclo successivo. Può essere utilizzato come ingresso di conteggio del ciclo di scansione.
SM0.7	Questo bit rispecchia la posizione dell'interruttore degli stati di funzionamento (off=TERM; on=RUN). Se viene utilizzato per attivare il modo freeport quando l'interruttore è in RUN, esso consente di abilitare la comunicazione con il PG commutando l'interruttore su TERM.

Funzioni logiche di base

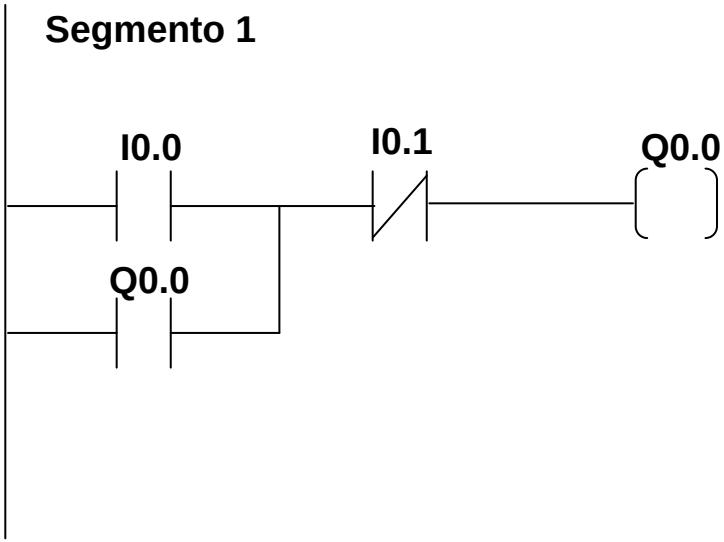


Merker (Relè interni)

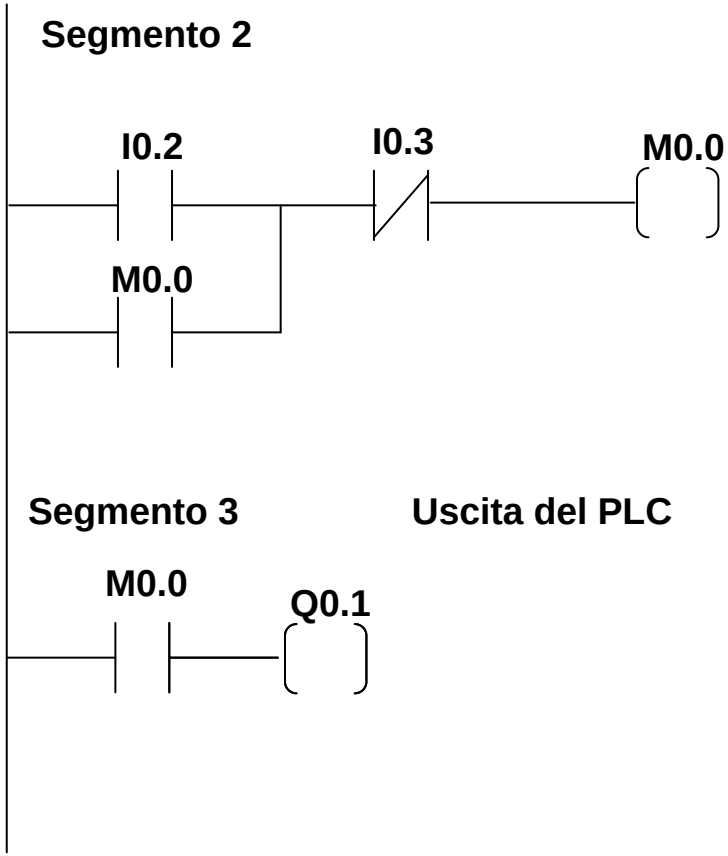


Uscita fisica NON ritentiva (Q0.0)

Al ripristino della tensione oppure nel passaggio da RUN a STOP e poi ancora in RUN, questa uscita sarà sempre diseccitata, qualunque sia stata la condizione di stato che aveva prima di tale eventi.

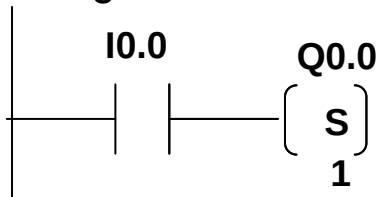


Uscita fisica ritentiva con Merker



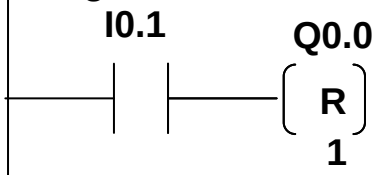
Bobine di SET / RESET

Segmento 1 set di una uscita



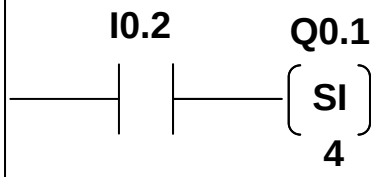
Chiudendo il contatto I0.0 si ottiene il set dell'uscita Q0.0

Segmento 2 reset di una uscita



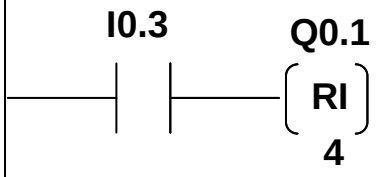
Chiudendo il contatto I0.1 si ottiene il set dell'uscita Q0.0

Segmento 3 set di più uscite



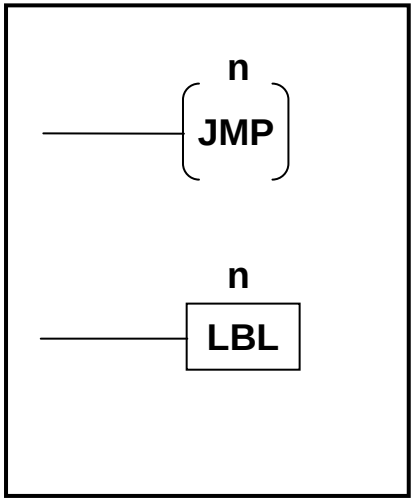
Chiudendo il contatto I0.2 si ottiene il set delle uscite Q0.1, Q0.2, Q0.3, Q0.4, essendo Q0.1 l'inizio del settaggio e avendo posto il numero 4 a indicare il numero di uscite da settare (DAL BIT 1 AL BIT 4).

Segmento 4 reset di più uscite



Con questa operazione si resettano le stesse uscite

Salto di programma



L'operazione Salta all'etichetta esegue una diramazione verso l'etichetta specificata (n) all'interno del programma. Quando viene effettuato il salto, il valore della sommità dello stack è sempre un 1 logico.

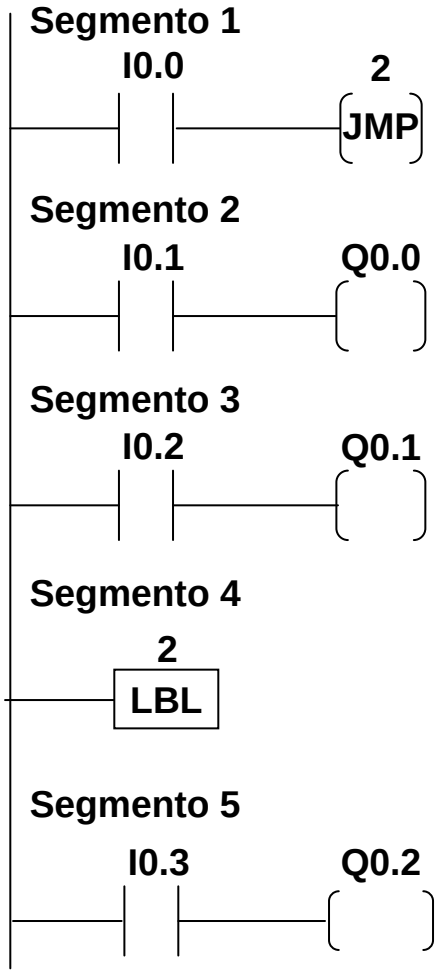
L'operazione Definisci l'etichetta (LBL) contrassegna l'indirizzo della destinazione del salto (n).

Operandi n: parola costante (da 0 a 255)

Il salto e la relativa etichetta devono trovarsi entrambi nel programma principale, in un sottoprogramma o in una routine di interrupt. Non si può saltare dal programma principale ad una etichetta che si trova in un sottoprogramma o in una routine di interrupt. Analogamente, non sarà possibile saltare da un sottoprogramma o routine di interrupt ad una etichetta collocata in un punto diverso del programma.

Esempio JMP e LBL

Quando la bobina di JMP è attivata, il programma esegue un salto fino alla etichetta LBL che ha lo stesso numero definito nel JMP.



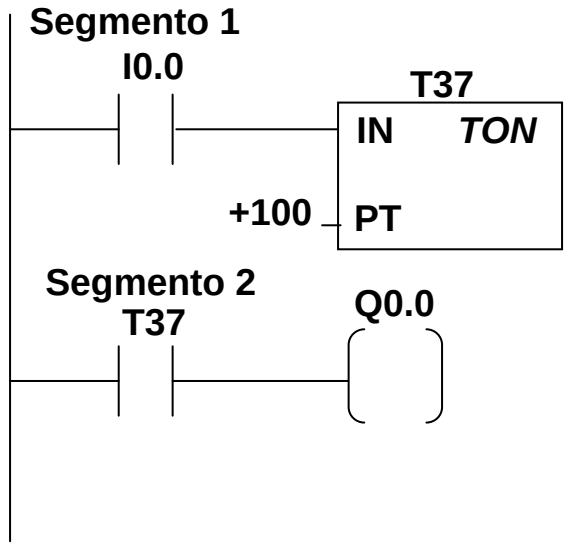
ESECUZIONE DEL SALTO DI PROGRAMMA (JMP)

Quando la bobina di JMP è eccitata, il programma esegue un salto fino al numero del segmento definito sia nell'etichetta di JMP che in quella di LBL. Si ricorda che i numeri associati a tali funzioni, devono obbligatoriamente essere uguali

FINE DEL SALTO DI PROGRAMMA (LBL)

L'istruzione LBL va programmata dopo il segmento di fine salto, associandogli lo stesso numero scritto nell'istruzione di JMP

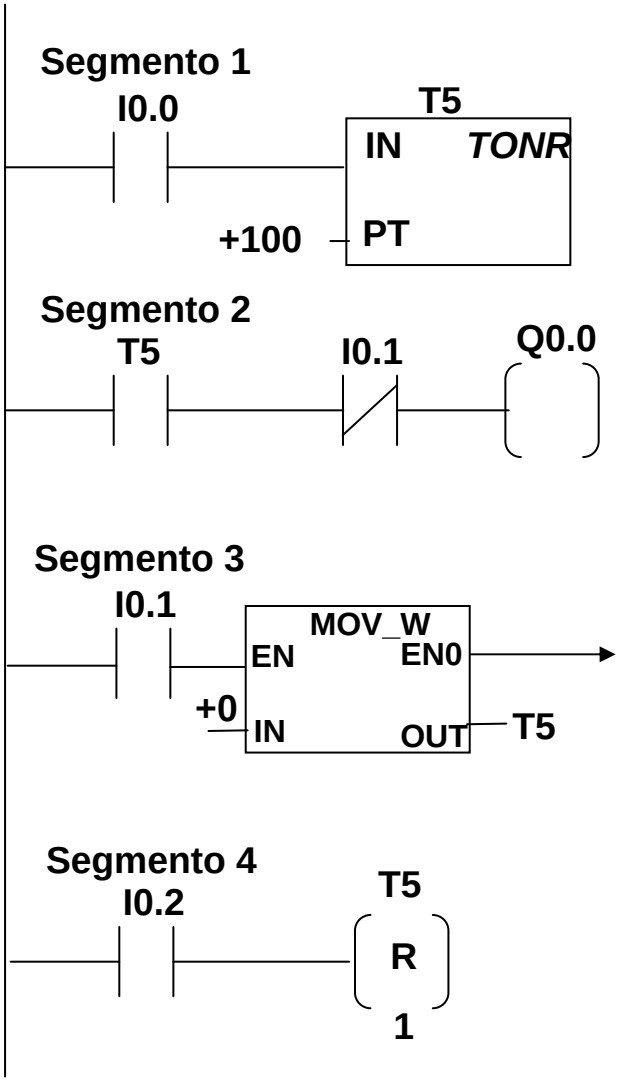
Temporizzatore ritardato all’eccitazione



Conteggia il tempo quando l'ingresso di abilitazione è attivo (ON). Il bit di temporizzazione viene attivato quando il valore corrente diventa maggiore o uguale al tempo preimpostato (PT). Quando l'ingresso di abilitazione è disattivato (OFF), il valore corrente del temporizzatore di ritardo all'inserzione viene resettato.
Ogni conteggio del valore corrente è un multiplo della base di tempo: 1ms, 10ms, 100ms.

L'operazione Resetta (R) consente di resettare tutti i tipi di temporizzatori ed esegue le seguenti operazioni:
Bit di temporizzazione = OFF e valore corrente di temporizzazione = 0

Temporizzatore ritardato all'eccitazione con memoria



Bit di temporizzazione ON - Il valore corrente avanza fino a 32.767

Ingresso ON =>Il valore corrente conta il tempo

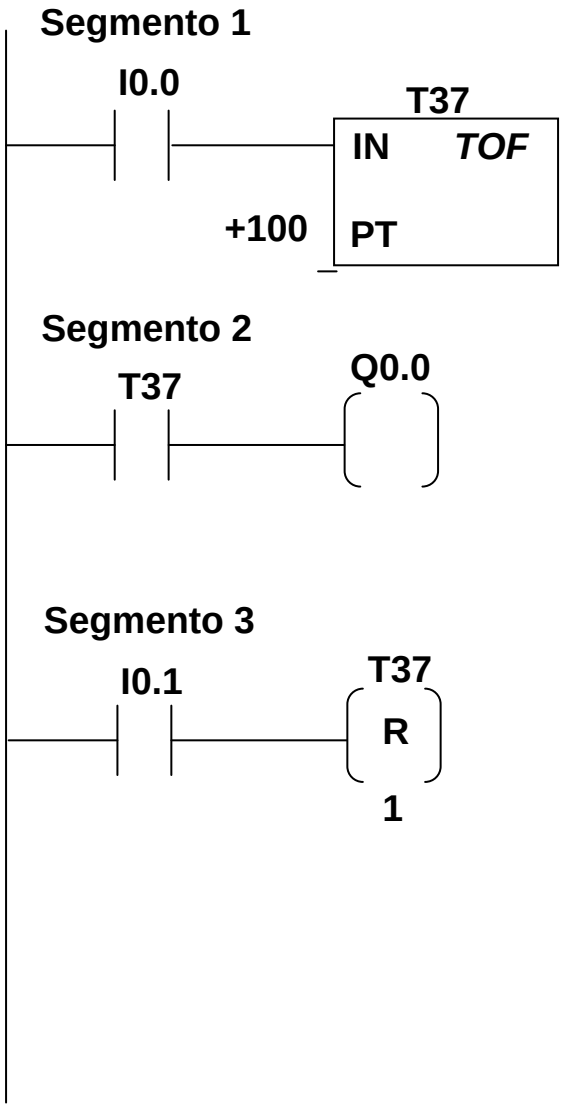
Ingresso OFF =>Il bit di temporizzazione e il valore corrente mantengono l'ultimo stato

Il bit di temporizzazione è OFF - Il valore corrente può essere mantenuto

Azzeramento del valore corrente del temporizzatore, ma il bit di temporizzazione rimane in ON

L'operazione Reset è l'unica che consente di resettare il temporizzatore TONR.

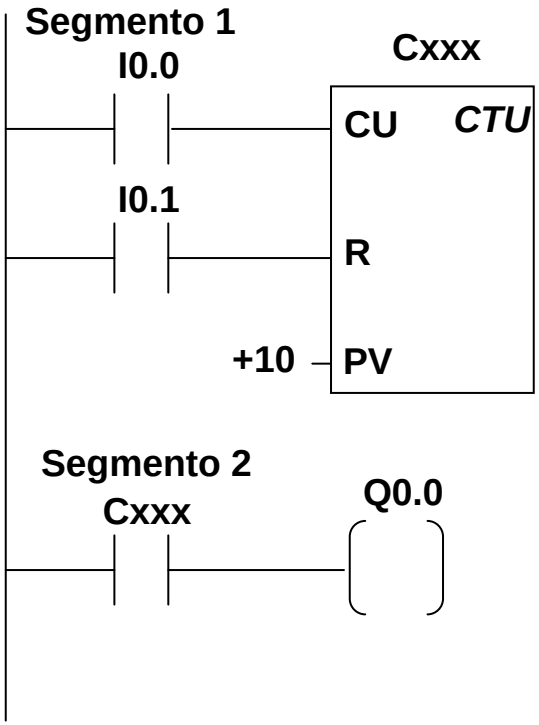
Temporizzatore ritardato alla diseccitazione



Il temporizzatore conta dopo una transizione ON – OFF
Ingresso ON => valore corrente conta il tempo
Bit di temporizzazione OFF - Valore corrente = preimpostato, smette di contare
Ingresso OFF => Bit di temporizzazione ON - Valore corrente = 0

Se si esegue un reset, i temporizzatori TOF potranno essere riavviati solo dopo una transizione ON - OFF dell'ingresso di abilitazione.

Contatore in avanti

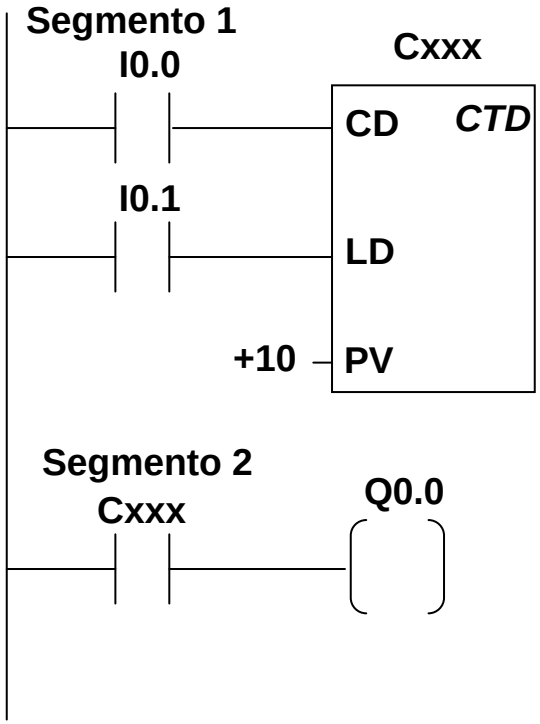


CTU conta in avanti

L'operazione Conta in avanti conta in avanti fino al valore massimo sui fronti di salita dell'ingresso di conteggio in avanti (CU). Quando il valore corrente (Cxxx) è >= al valore preimpostato (PV), il bit di conteggio (Cxxx) viene attivato. Il contatore viene resettato quando si attiva l'ingresso di reset (R). Il contatore smette di contare quando raggiunge il PV.

Aree dei contatori: Cxxx=da C0 a C255

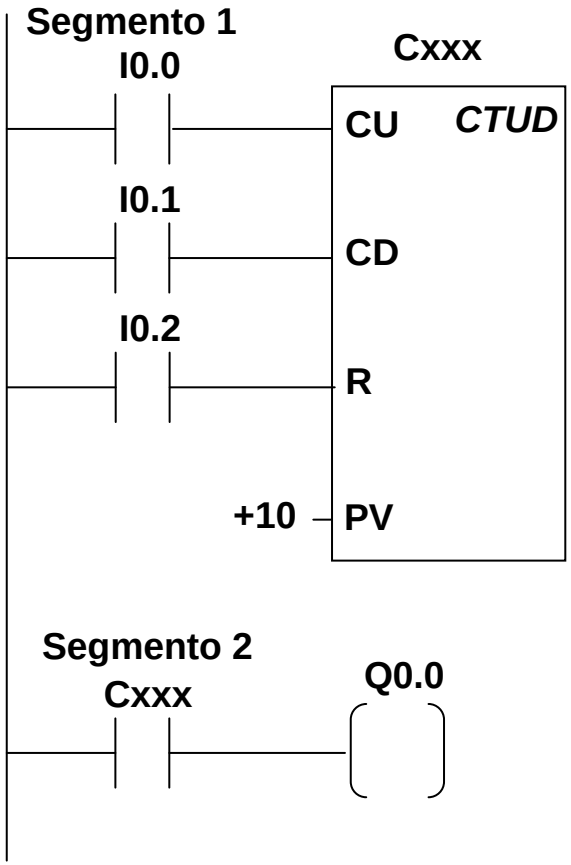
Contatore di Deconteggio (all’indietro)



Il contatore Conta indietro conta all'indietro da un valore predefinito sui fronti di salita dell'ingresso di conteggio all'indietro (CD). Quando il valore corrente diventa uguale a zero, il bit di conteggio (Cxxx) viene attivato. Il contatore resetta il bit di conteggio (Cxxx) e carica il valore corrente con il valore preimpostato (PV) quando l'ingresso di caricamento (LD) diventa attivo. Il contatore di deconteggio smette di contare quando raggiunge lo zero.
Pertanto la prima operazione da compiere è l’attivazione di LD affinché venga caricato il valore preimpostato PV.

Aree dei contatori: Cxxx=da C0 a C255

Contatore in avanti / indietro

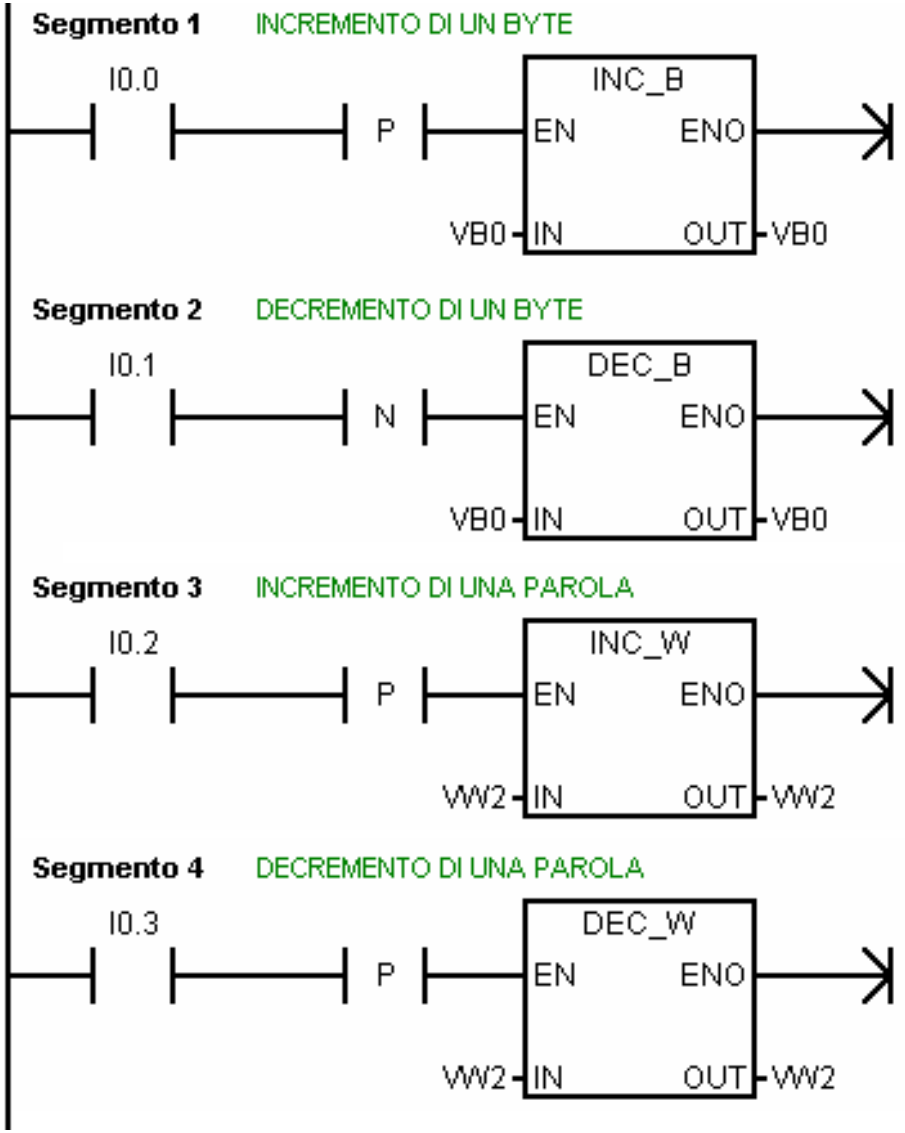


CTUD conta in avanti/indietro

L'operazione Conta in avanti/indietro conta in avanti sui fronti di salita dell'ingresso di conteggio in avanti (CU) e conta all'indietro sui fronti di salita dell'ingresso di conteggio all'indietro (CD). Quando il valore corrente (Cxxx) è >= al valore preimpostato (PV), il bit di conteggio (Cxxx) viene attivato. Il contatore viene resettato quando si attiva l'ingresso di reset (R).

Aree dei contatori: Cxxx=da C0 a C255

Incremento e Decremento

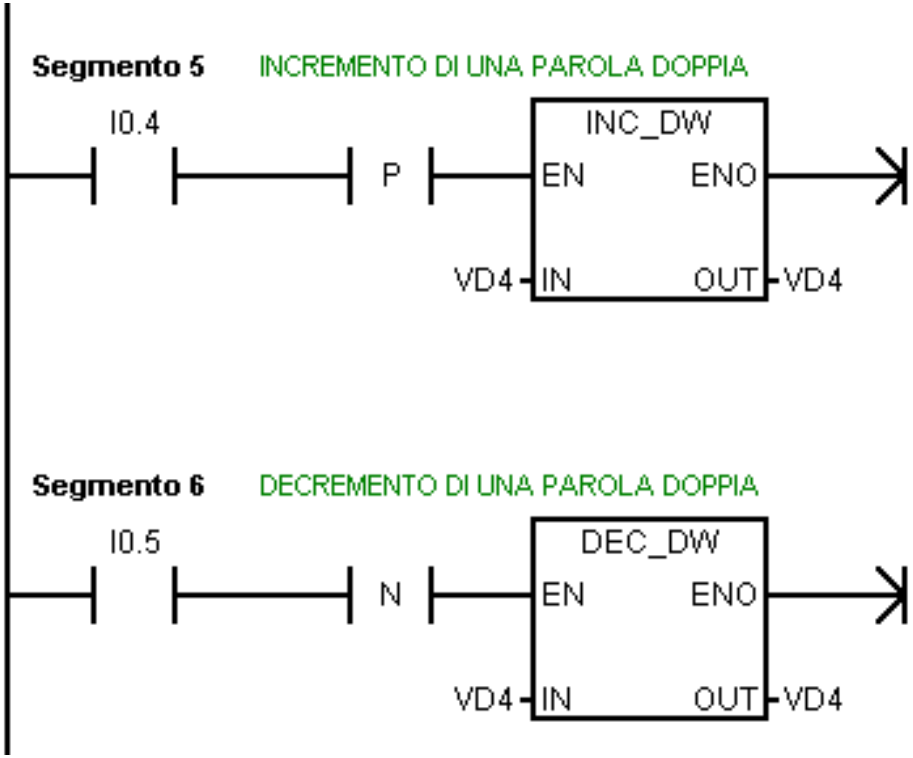


Il contatto Transizione Positiva attiva il flusso di corrente per un ciclo di scansione ad ogni transizione da off a on, generando un fronte di salita dell'ingresso I0.0; quindi ogni qualvolta che si chiude tale ingresso si ottiene l'incremento di una unità del contenuto del byte 0, visualizzato in esadecimale

Il contatto Transizione negativa attiva il flusso di corrente per un ciclo di scansione ad ogni transizione da on a off, generando un fronte di discesa dell'ingresso I0.1; quindi ogni qualvolta che si apre tale ingresso si ottiene il decremento di una unità del contenuto del byte 0, visualizzato in esadecimale .

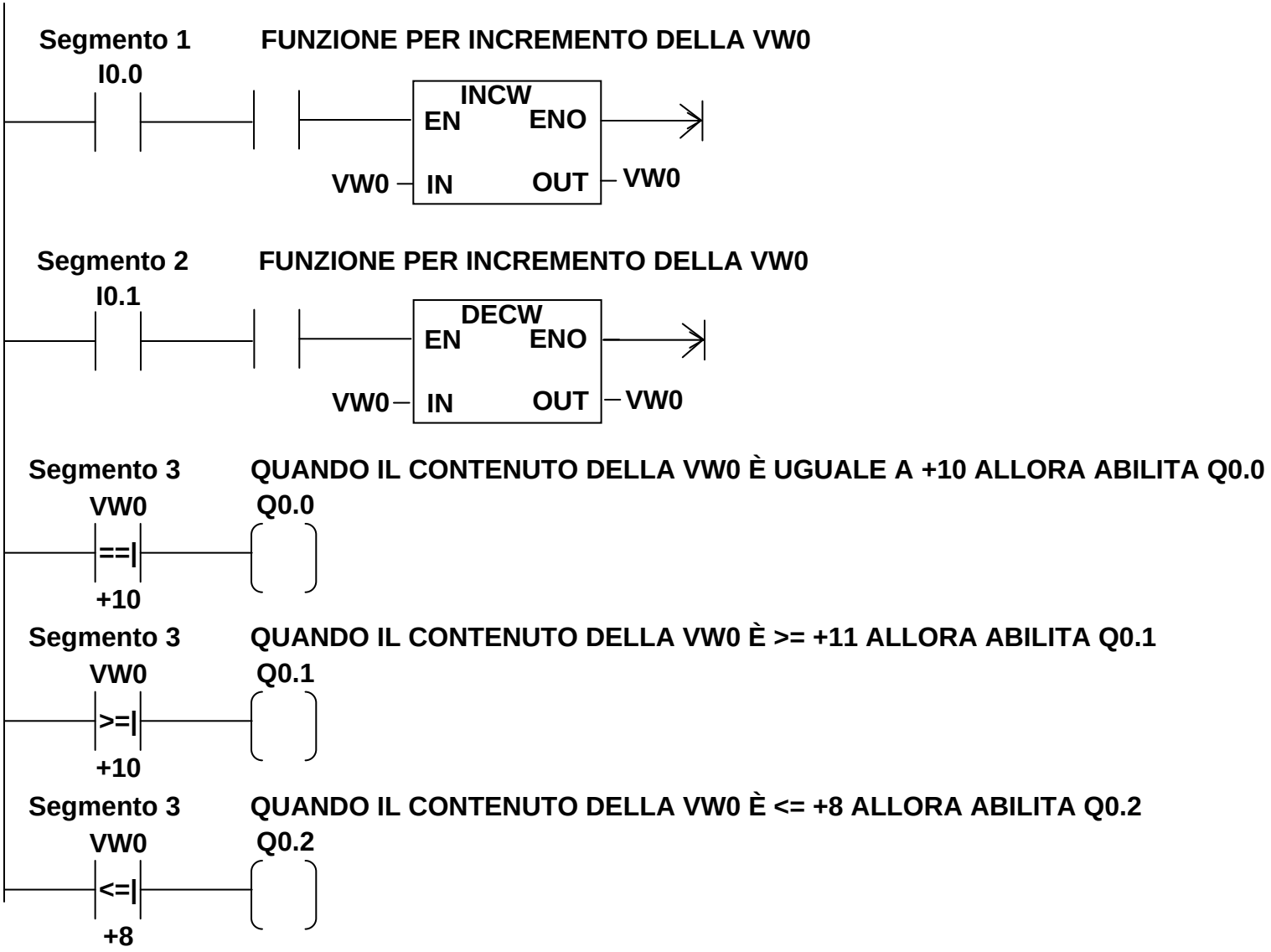
Il valore letto nella parola viene espresso in decimale

Incremento e Decremento

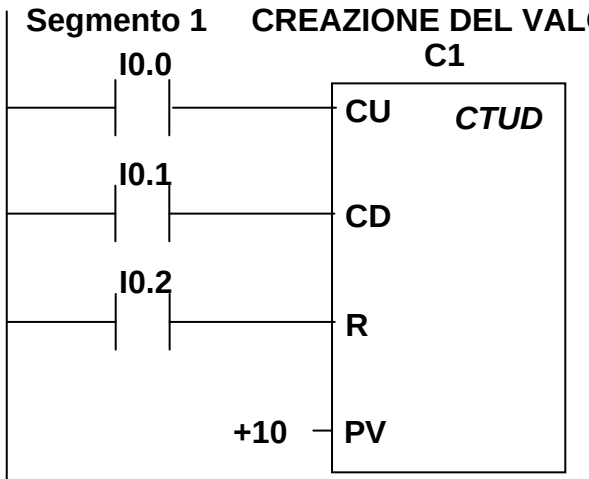


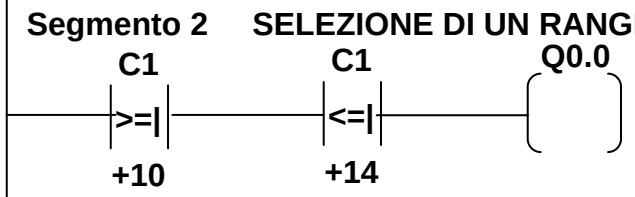
Il valore letto nella doppia parola viene espresso in DECIMALE

Confronti

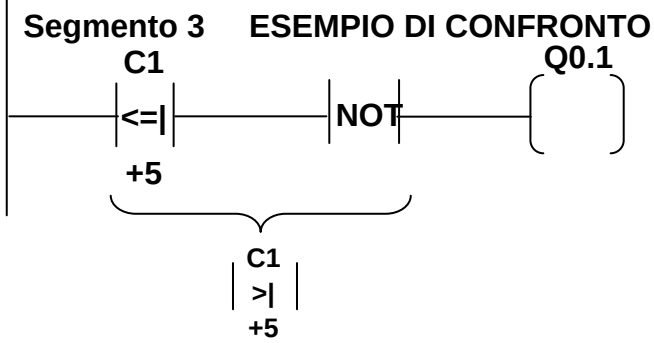

Segmento 1
FUNZIONE PER INCREMENTO DELLA VW0
I0.0
INCW
EN
ENO
VW0
IN
OUT
VW0
Segmento 2
FUNZIONE PER INCREMENTO DELLA VW0
I0.1
DECW
EN
ENO
VW0
IN
OUT
VW0
Segmento 3
QUANDO IL CONTENUTO DELLA VW0 È UGUALE A +10 ALLORA ABILITA Q0.0
VW0
==I
+10
Q0.0
Segmento 3
QUANDO IL CONTENUTO DELLA VW0 È >= +11 ALLORA ABILITA Q0.1
VW0
>=I
+10
Q0.1
Segmento 3
QUANDO IL CONTENUTO DELLA VW0 È <= +8 ALLORA ABILITA Q0.2
VW0
<=I
+8
Q0.2

Confronti





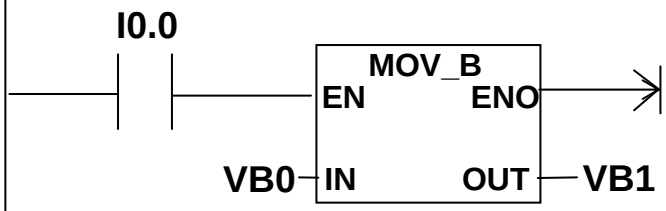
Queste operazioni di confronto, generano l'attivazione dell'uscita quando il valore corrente del contatore è compreso tra 10 e 15



I tipi di confronto (>) (<>) (<) possono essere realizzati mediante (<=) (=) (>=) seguiti da una operazione NOT

Trasferimento

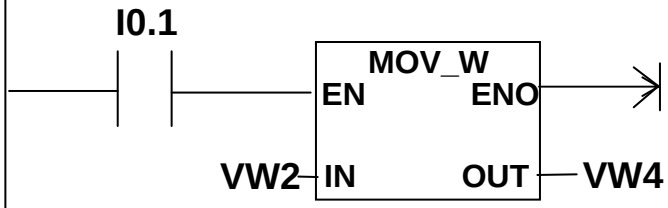
Segmento 1 **TRASFERIMENTO DI UN BYTE IN UN ALTRO BYTE**



Il Byte è una locazione di memoria di 8 bit

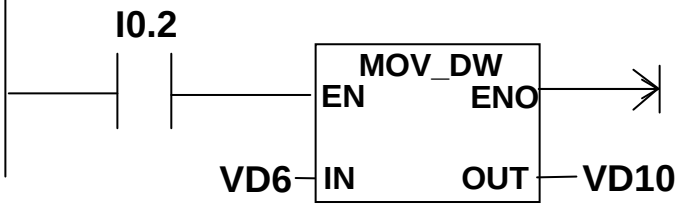
Verificare il trasferimento nella tabella dei dati

Segmento 1 **TRASFERIMENTO DI UNA WORD IN UN'ALTRA WORD**



La parola VW2 è composta dai Bytes VB2 e VB3 mentre VW4 è composta dai Bytes VB4 e VB5

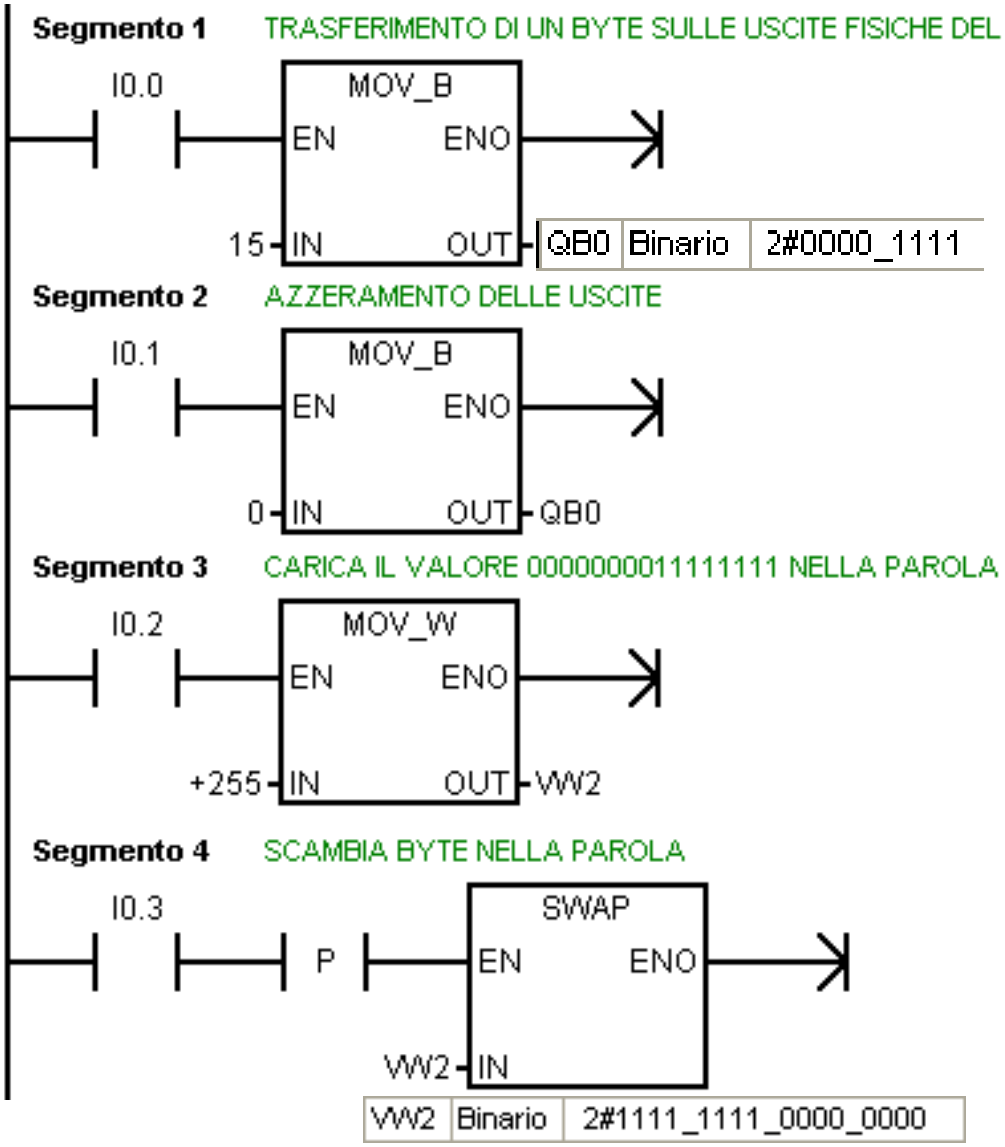
Segmento 1 **TRASFERIMENTO DI UNA DOPPIA WORD IN UN'ALTRA DOPPIA WORD**



La doppia parola VD6 è composta da VW6 e VW7, e rispettivamente dai Bytes VB6, VB7, VB8, VB9

Per verificare l'avvenuto trasferimento, visualizzare e compilare la Tabella di Stato

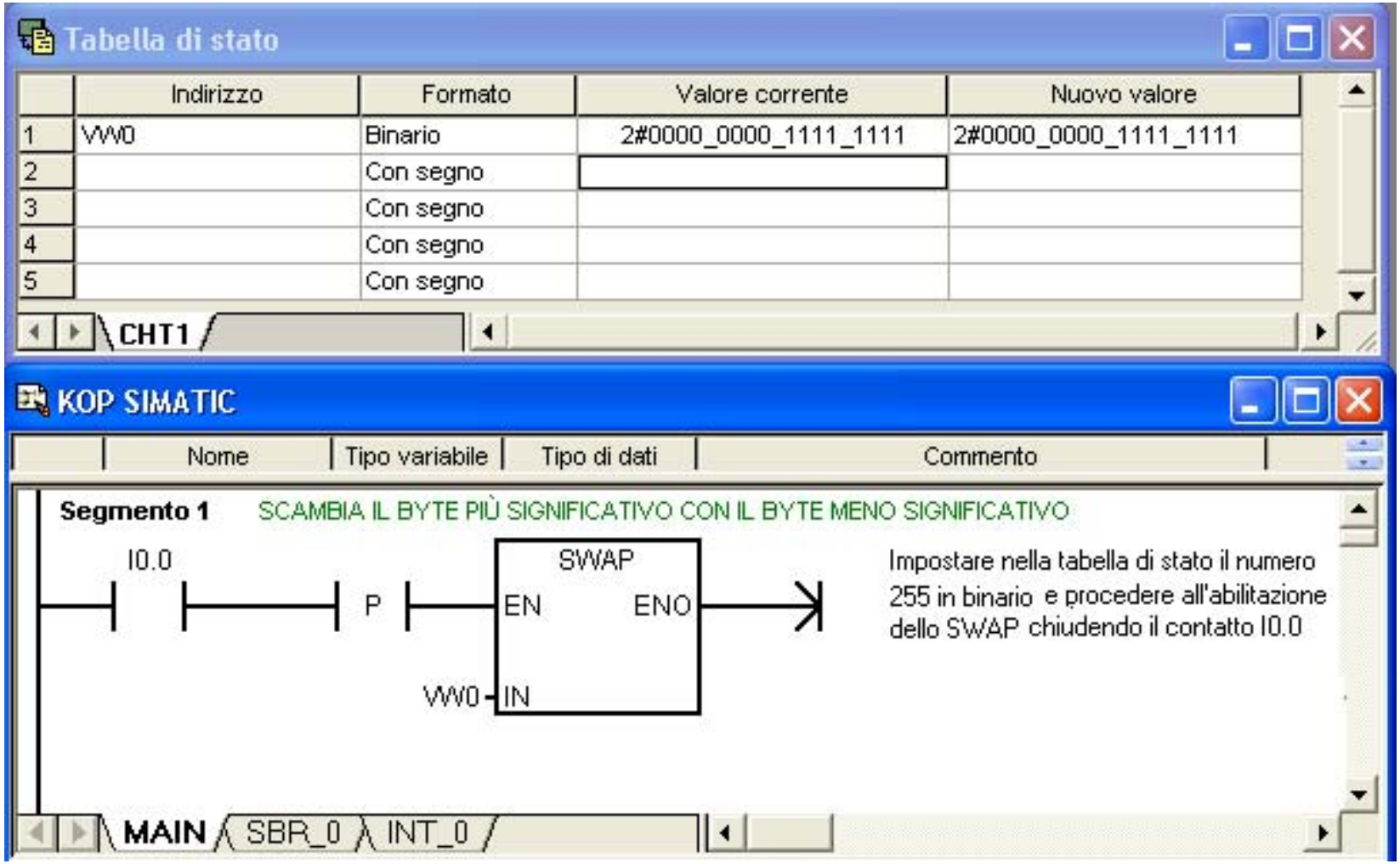
Trasferimenti sulle uscite



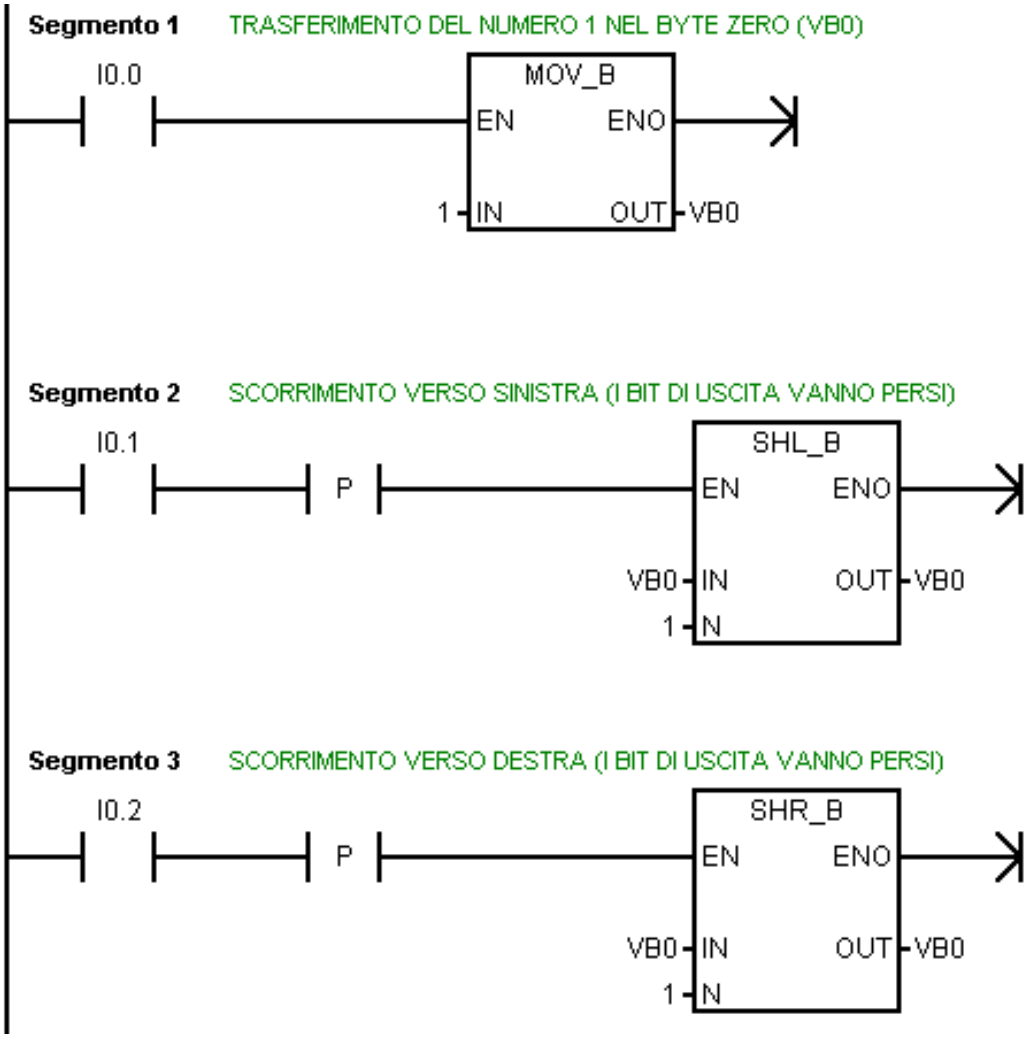
Associando all'ingresso IN della funzione MOV_B il valore 15 decimale, all'attivazione della funzione tramite un impulso all'ingresso I0.0 si ottiene la eccitazione delle uscite Q0.0 Q0.1 Q0.2 Q0.3

Per la funzione SWAP occorre un fronte di salita o un fronte di discesa

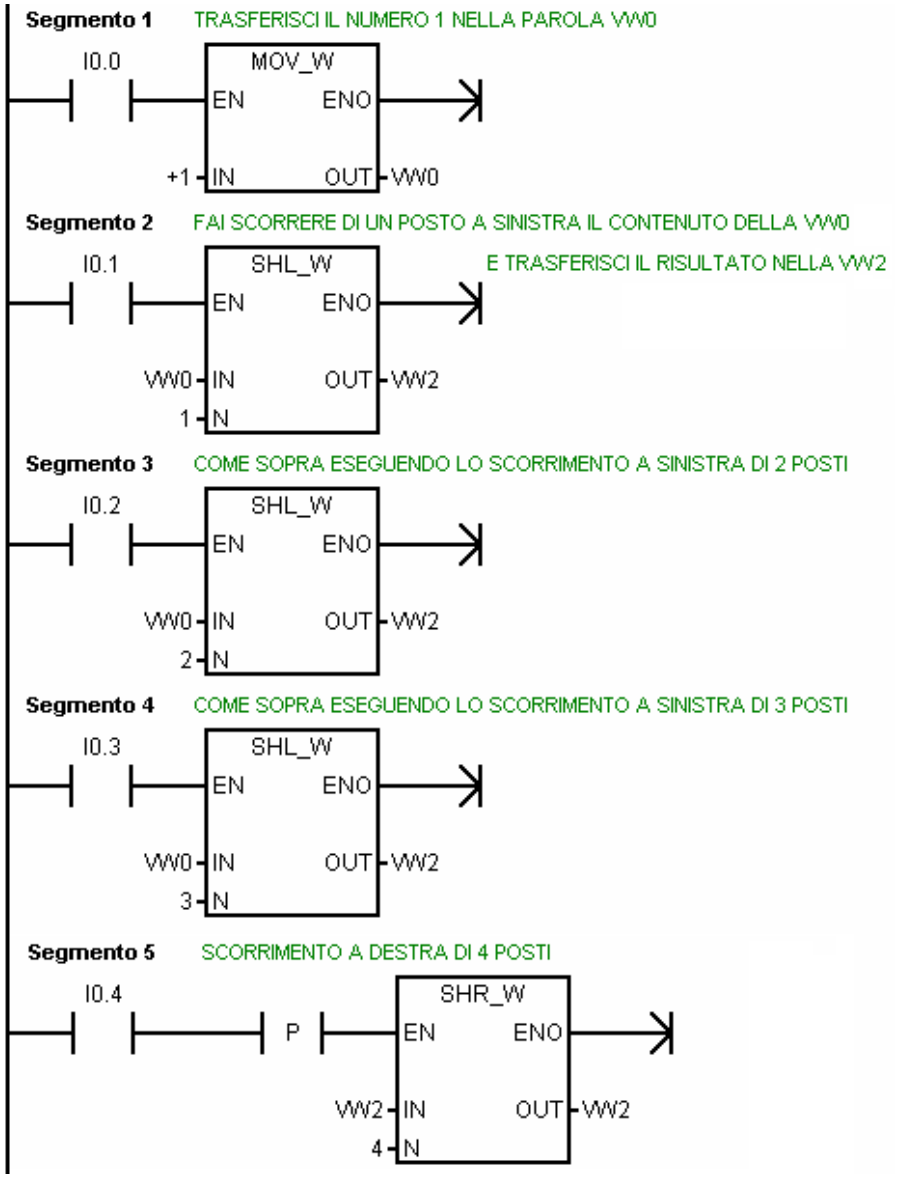
Operazione scambia byte nella parola (SWAP)



Scorrimento di un byte



Scorrimento di una parola

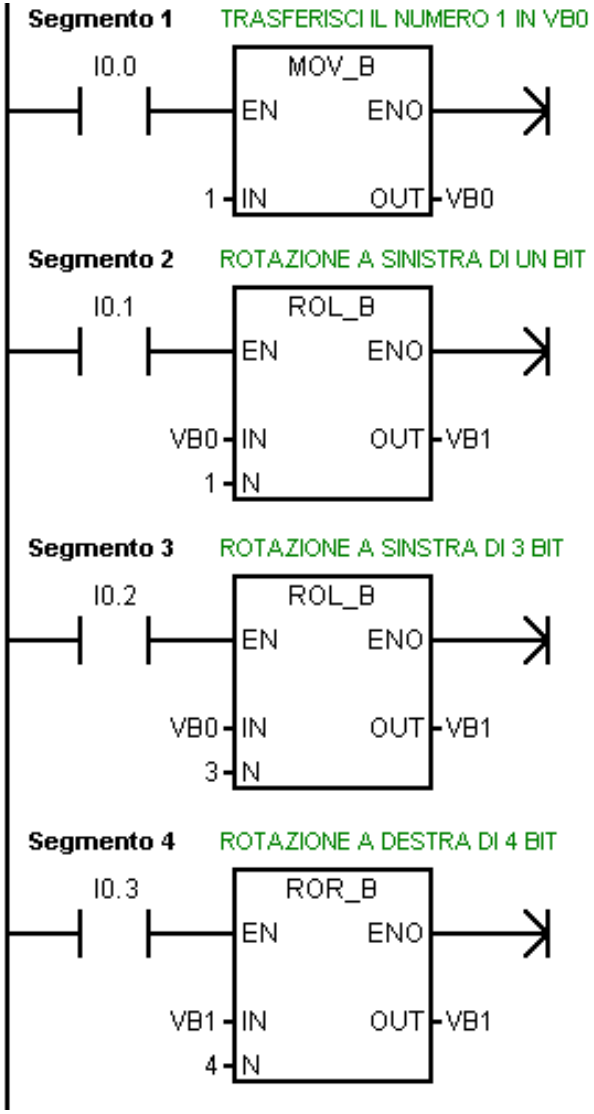


Nota: il dato scritto nella VW0 non cambia

Quindi sia ha lo scorrimento del contenuto di una parola e trasferimento del risultato in una seconda parola lasciando la prima inalterata

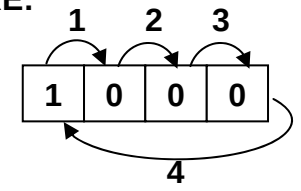
Si ottiene così l'azzeramento

Rotazione di un byte

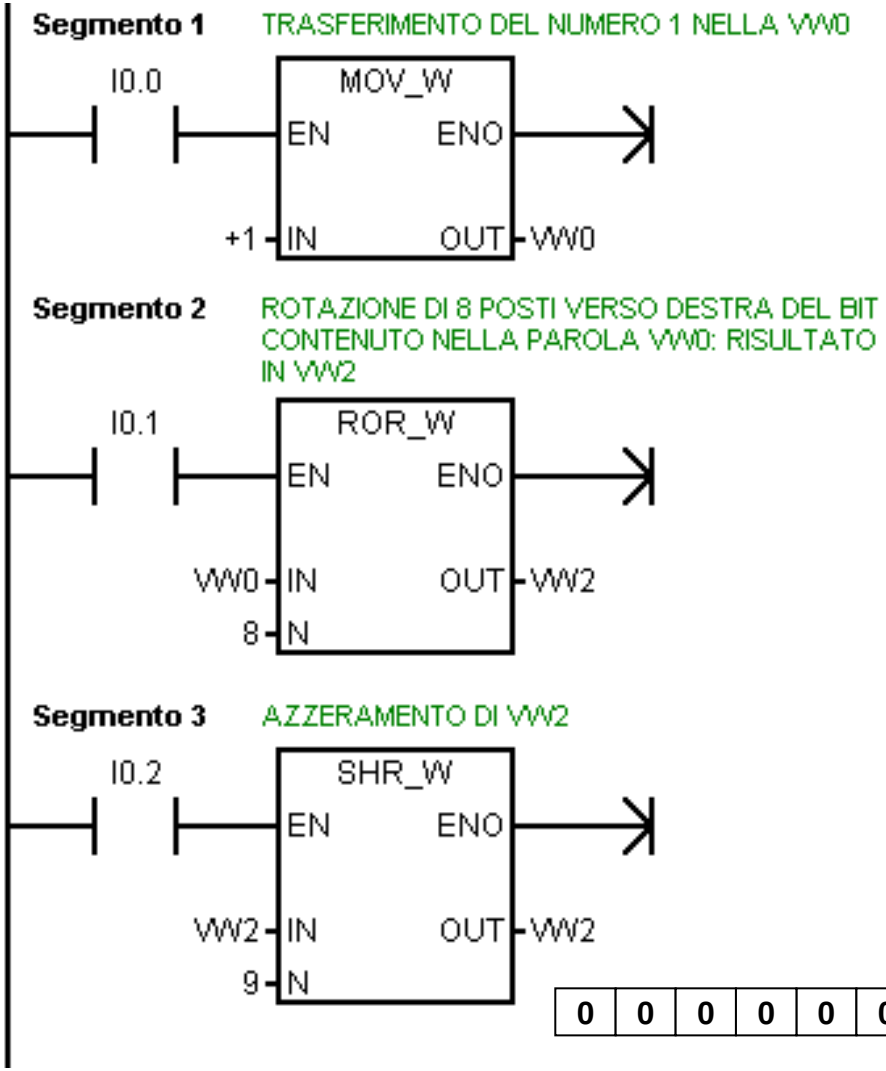


Nota: **con la funzione rotazione, il bit di uscita non va perso, ma rientra da destra verso sinistra o da sinistra verso destra, e pertanto il bit di uscita non va perso (al contrario della funzione SCORRIMENTO.**

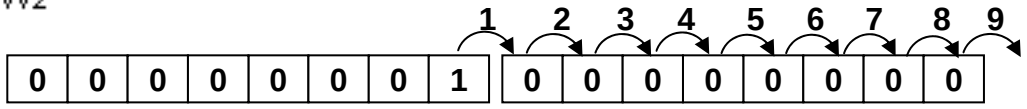
IN QUESTO CASO IL BYTE VB1 NON SI AZZERA MA MANTIENE LO STESSO VALORE:



Rotazione di una parola

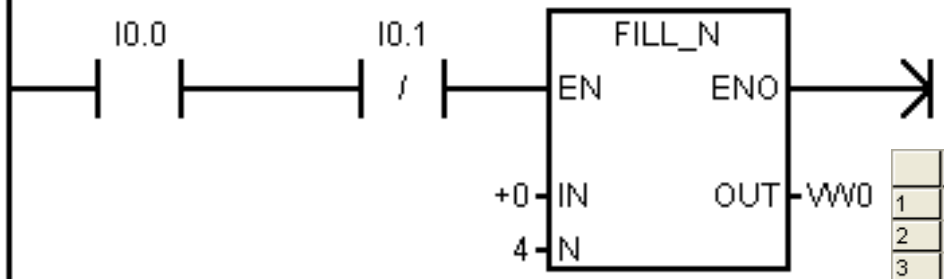


IL CONTENUTO DI VW0 RIMANE INALTERATO



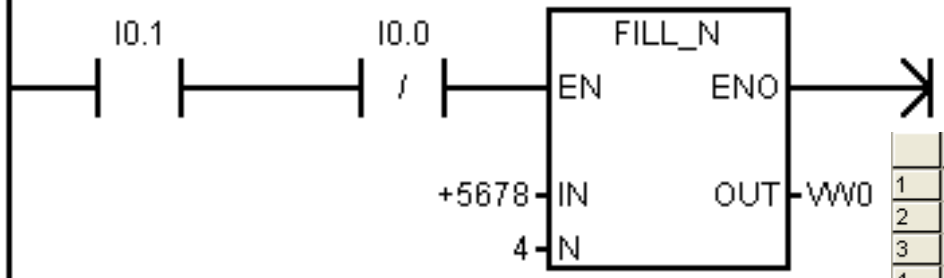
Trasferimento consecutivo (FILL-N)

Segmento 1 AZZERAMENTO DI 4 PAROLE CONSECUTIVE PARTENDO DALLA VW0



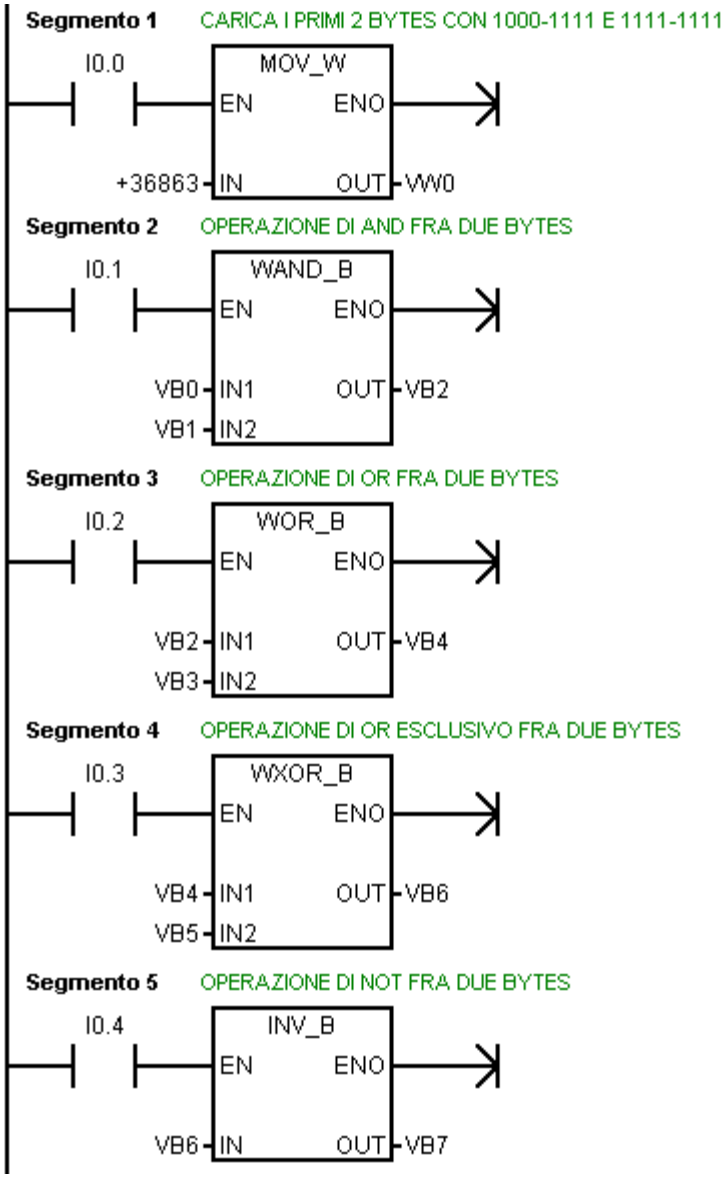
	Indirizzo	Formato	Valore corrente
1	VW0	Con segno	+0
2	VW2	Con segno	+0
3	VW4	Con segno	+0
4	VW6	Con segno	+0

Segmento 2 INTRODUZIONE DEL NUMERO 5678 IN 4 PAROLE CONSECUTIVE



	Indirizzo	Formato	Valore corrente
1	VW0	Con segno	+5678
2	VW2	Con segno	+5678
3	VW4	Con segno	+5678
4	VW6	Con segno	+5678

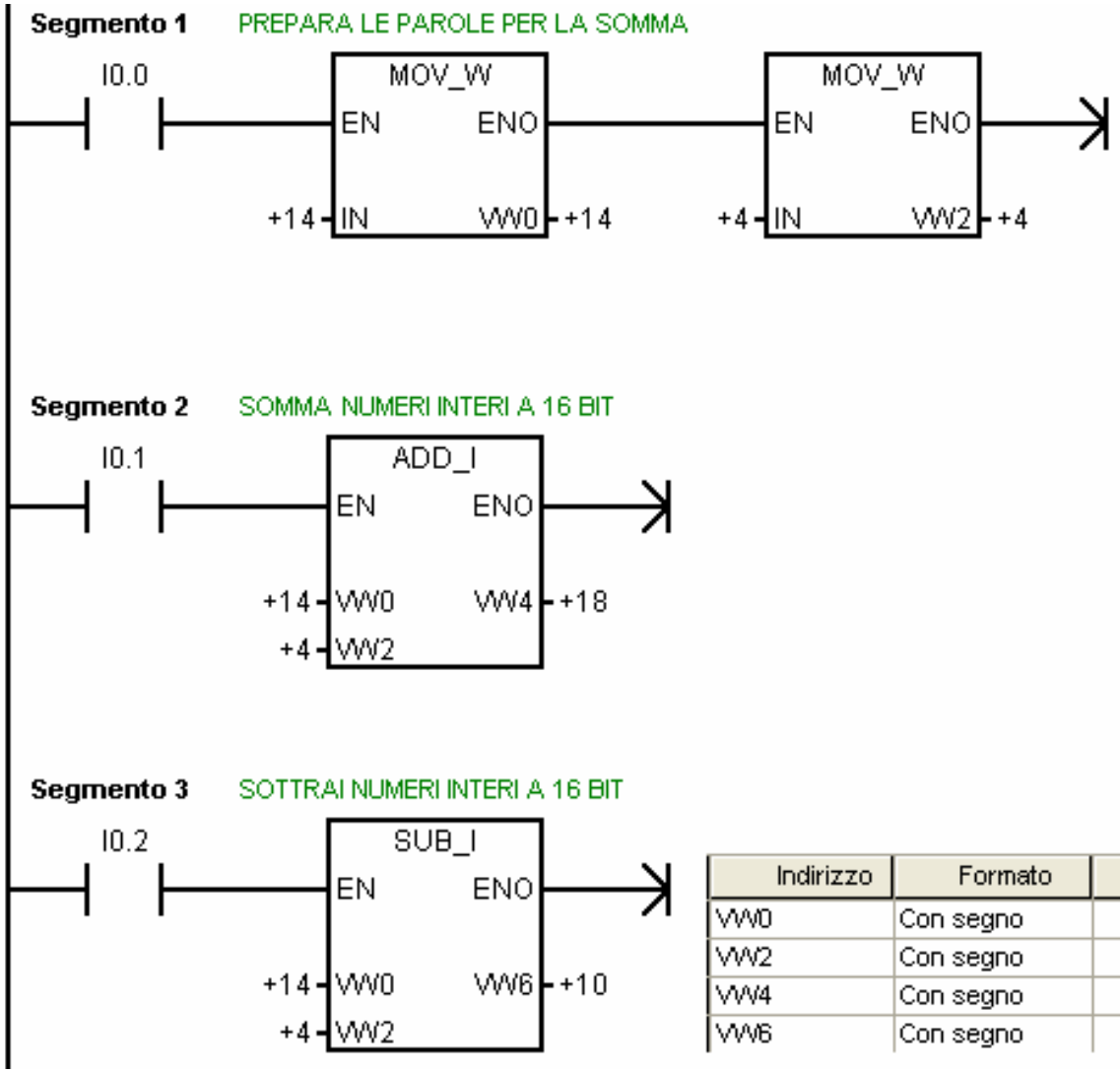
Funzioni logiche fra bytes



AND = Basta uno zero per avere una uscita zero
OR = Basta un uno per avere una uscita uno
XOR = Due segnali uguali in ingresso danno un uno in uscita
NOT = Negazione del segnale in ingresso

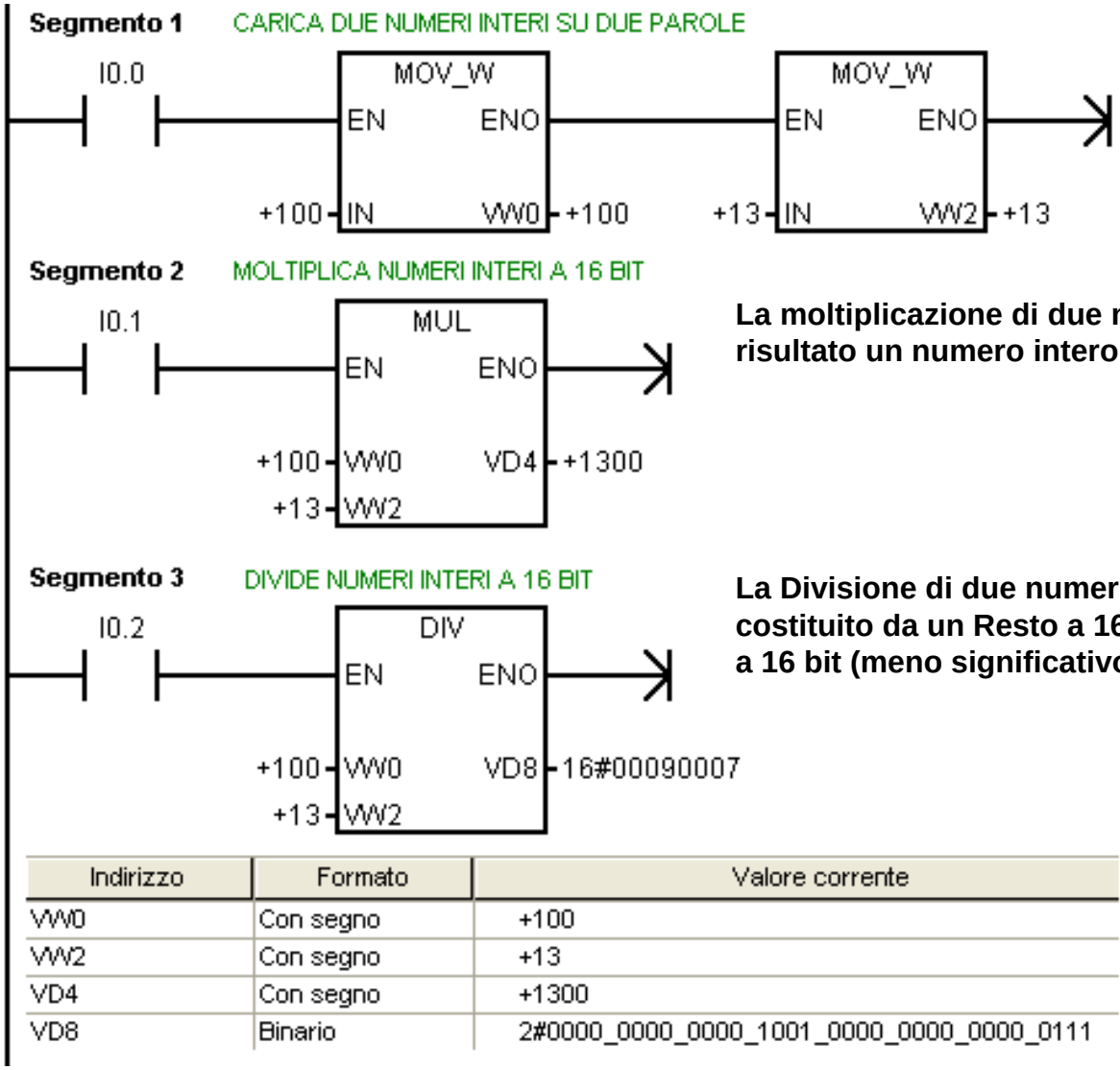
Indirizzo	Formato	Valore corrente
VB0	Binario	2#1000_1111
VB1	Binario	2#1111_1111
VB2	Binario	2#1000_1111
VB3	Binario	2#0000_0000
VB4	Binario	2#1000_1111
VB5	Binario	2#0000_0000
VB6	Binario	2#1000_1111
VB7	Binario	2#0111_0000

Somma e Sottrazione



Indirizzo	Formato	Valore corrente
VW0	Con segno	+14
VW2	Con segno	+4
VW4	Con segno	+18
VW6	Con segno	+10

Moltiplicazione e Divisione

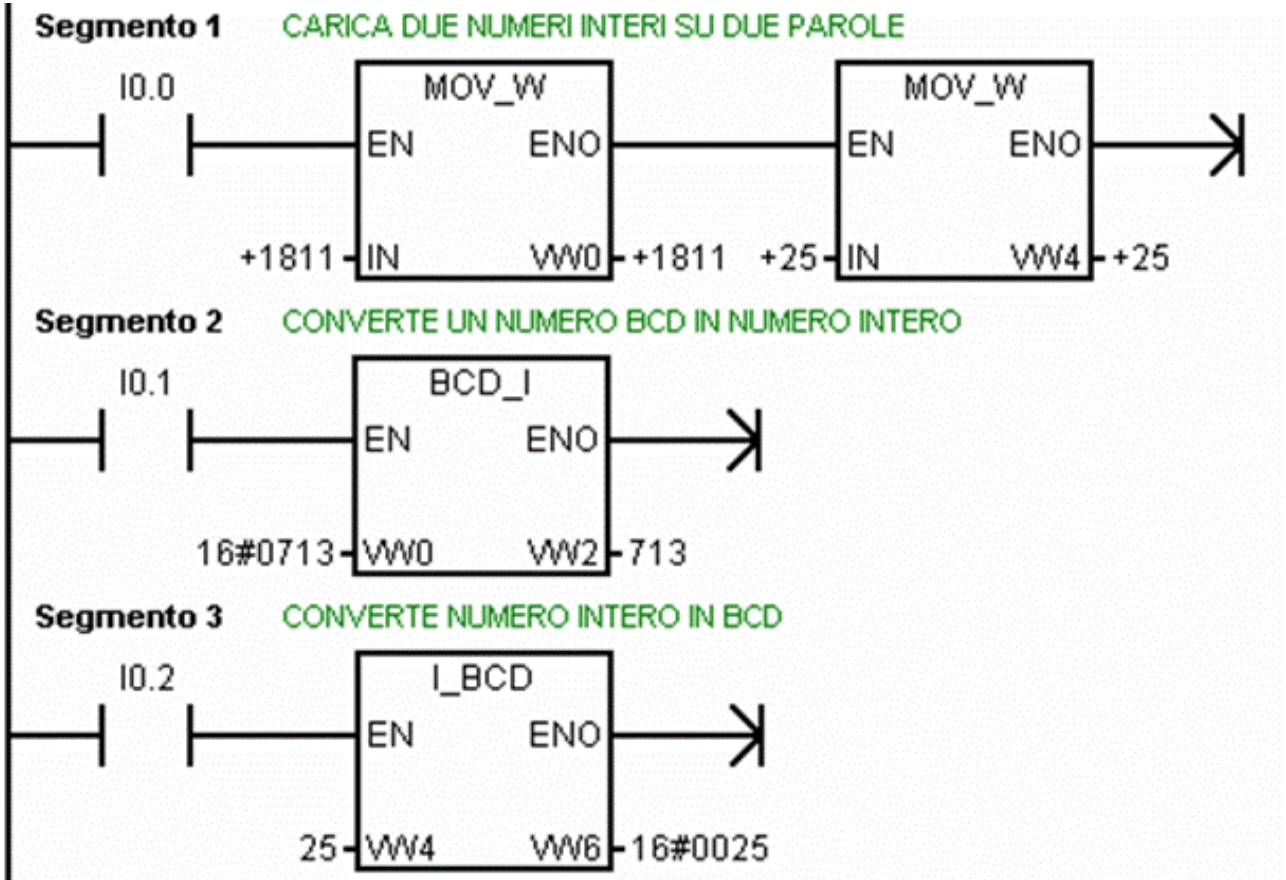


La moltiplicazione di due numeri interi a 16 bit produce come risultato un numero intero a 32 bit

La Divisione di due numeri ineri a 16 bit dà un risultato a 32 bit, costituito da un Resto a 16 bit (più significativo) e un Quoziente a 16 bit (meno significativo): infatti 100 : 13 = 7 con il resto di 9

Indirizzo	Formato	Valore corrente
VW0	Con segno	+100
VW2	Con segno	+13
VD4	Con segno	+1300
VD8	Binario	2#0000_0000_0000_1001_0000_0000_0000_0111

Conversioni



Indirizzo	Formato	Valore corrente
VW0	Binario	2#0000_0111_0001_0011
VW2	Con segno	+713
VW4	Con segno	+25
VW6	Binario	2#0000_0000_0010_0101

PLC SIEMENS

ESERCITAZIONI

Esempio di applicazione per miscelatore di vernici

Un serbatoio di miscelazione viene utilizzato per produrre diversi colori di vernice:

Due vernici vengono immesse alla sommità del serbatoio da due diverse condutture.

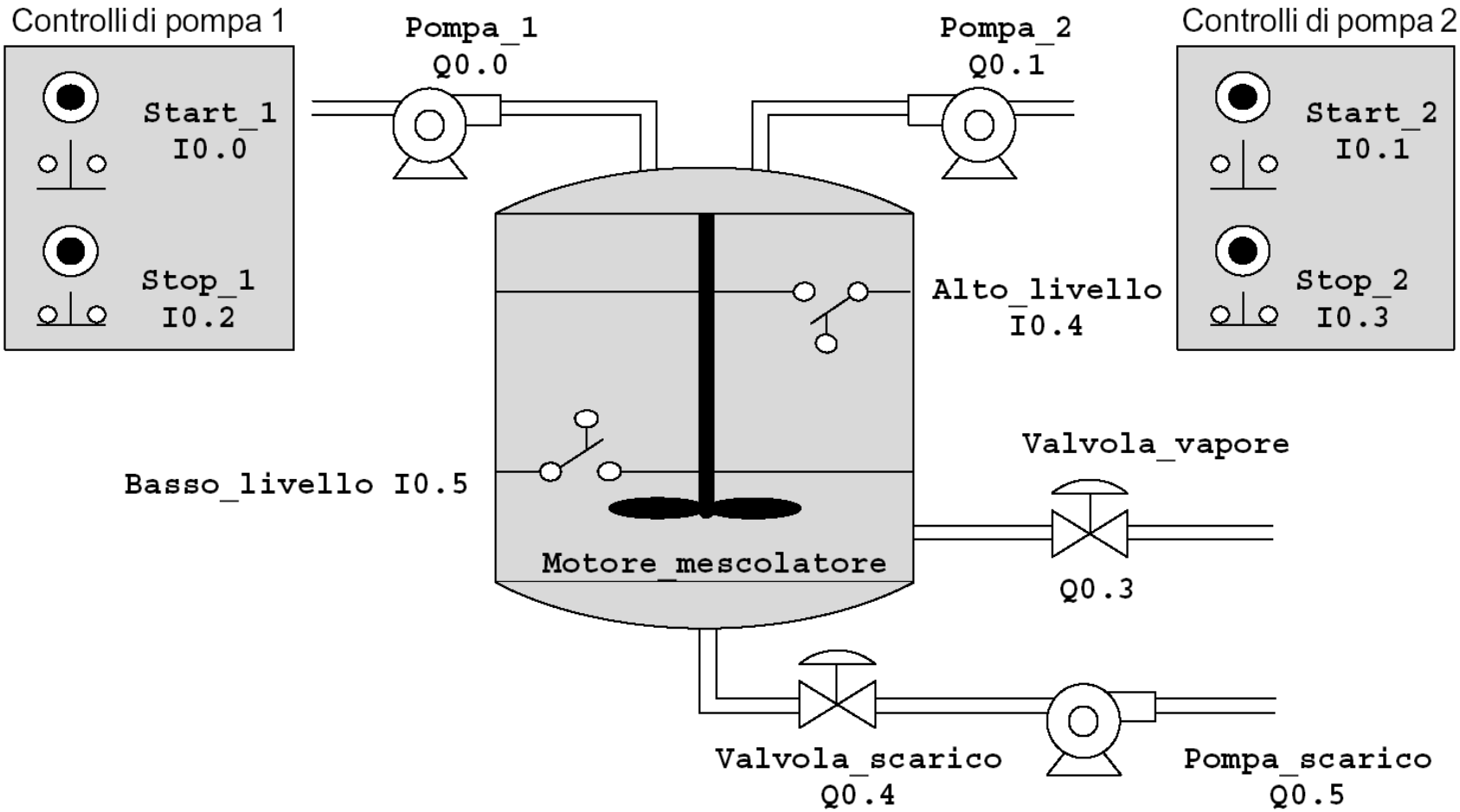
Una conduttura singola alla base del serbatoio trasporta la miscela finita.

Il programma di esempio controlla l'operazione di riempimento, sorveglia il livello del serbatoio, e gestisce il ciclo di miscelazione e riscaldamento, nella sequenza seguente.

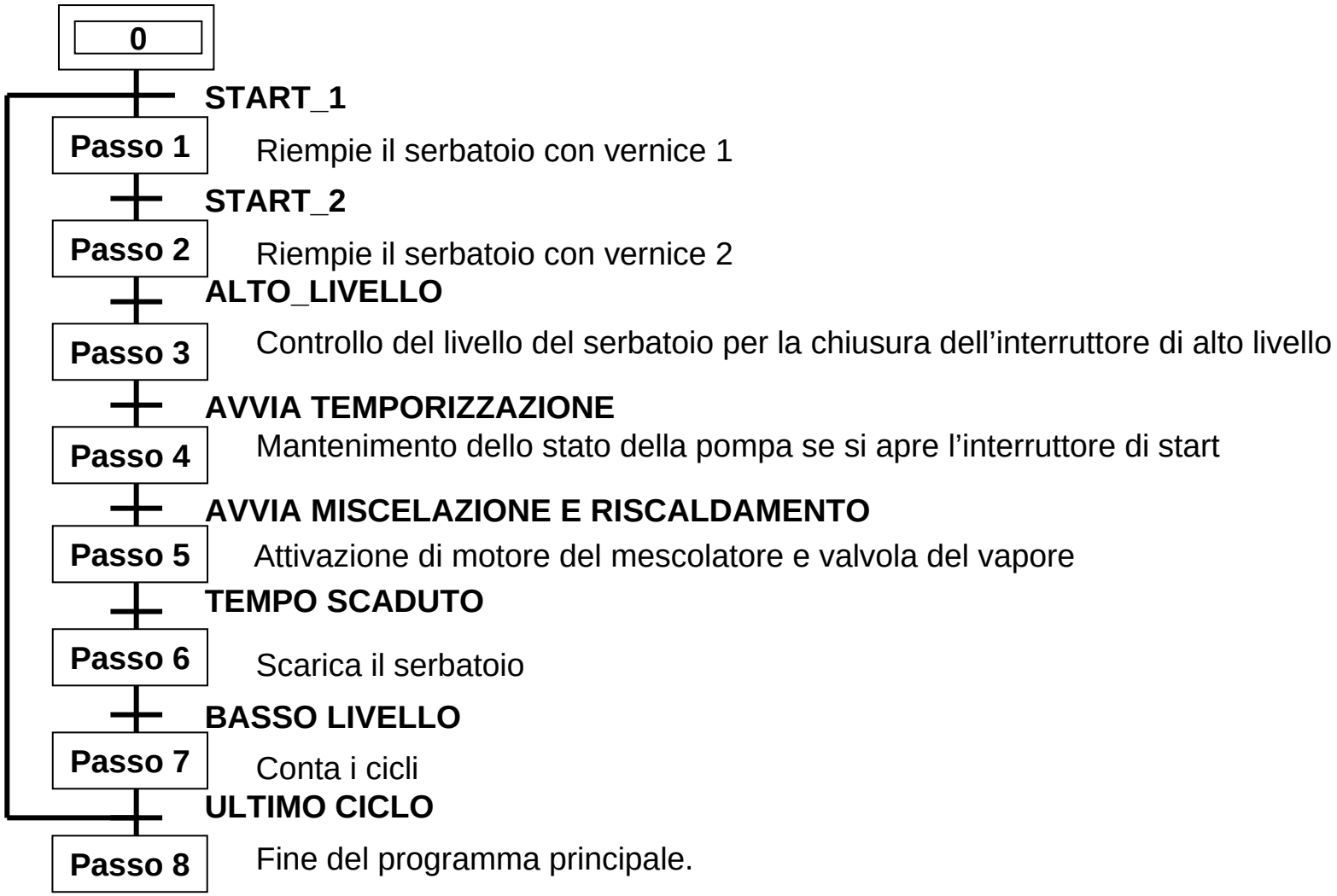
Ciclo di miscelazione e riscaldamento

Passo 1:	Riempie il serbatoio con vernice 1
Passo 2:	Riempie il serbatoio con vernice 2
Passo 3:	Controllo del livello del serbatoio per la chiusura dell’interruttore di alto livello
Passo 4:	Mantenimento dello stato della pompa se si apre l’interruttore di start
Passo 5:	Avviamento del ciclo di miscelazione e riscaldamento
Passo 6:	Attivazione di motore del mescolatore e valvola del vapore
Passo 7:	Scarica il serbatoio
Passo 8:	Conta i cicli

Miscelatore di vernici



Ciclo di miscelazione e riscaldamento

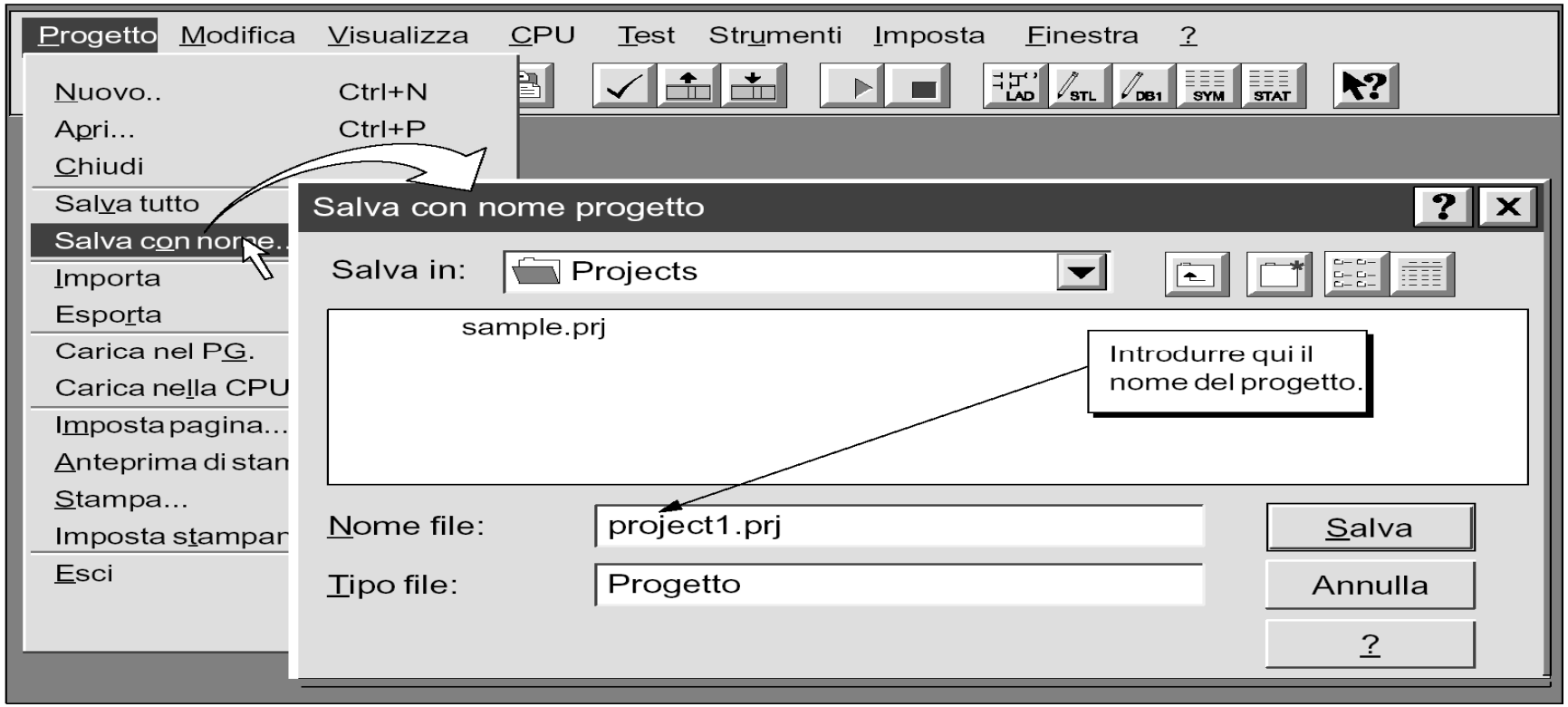


Assegnazione del nome al progetto di esempio

Selezionare il comando del menu **Progetto Salva con nome...**.

Nella casella Nome file digitare: **miscelatore di vernici.prj**

Fare clic sul pulsante “Salva”.



Creazione di una tabella dei simboli

Si apra l'editor della tabella dei simboli per definire il set dei nomi simbolici che rappresentano gli indirizzi assoluti nel programma di esempio

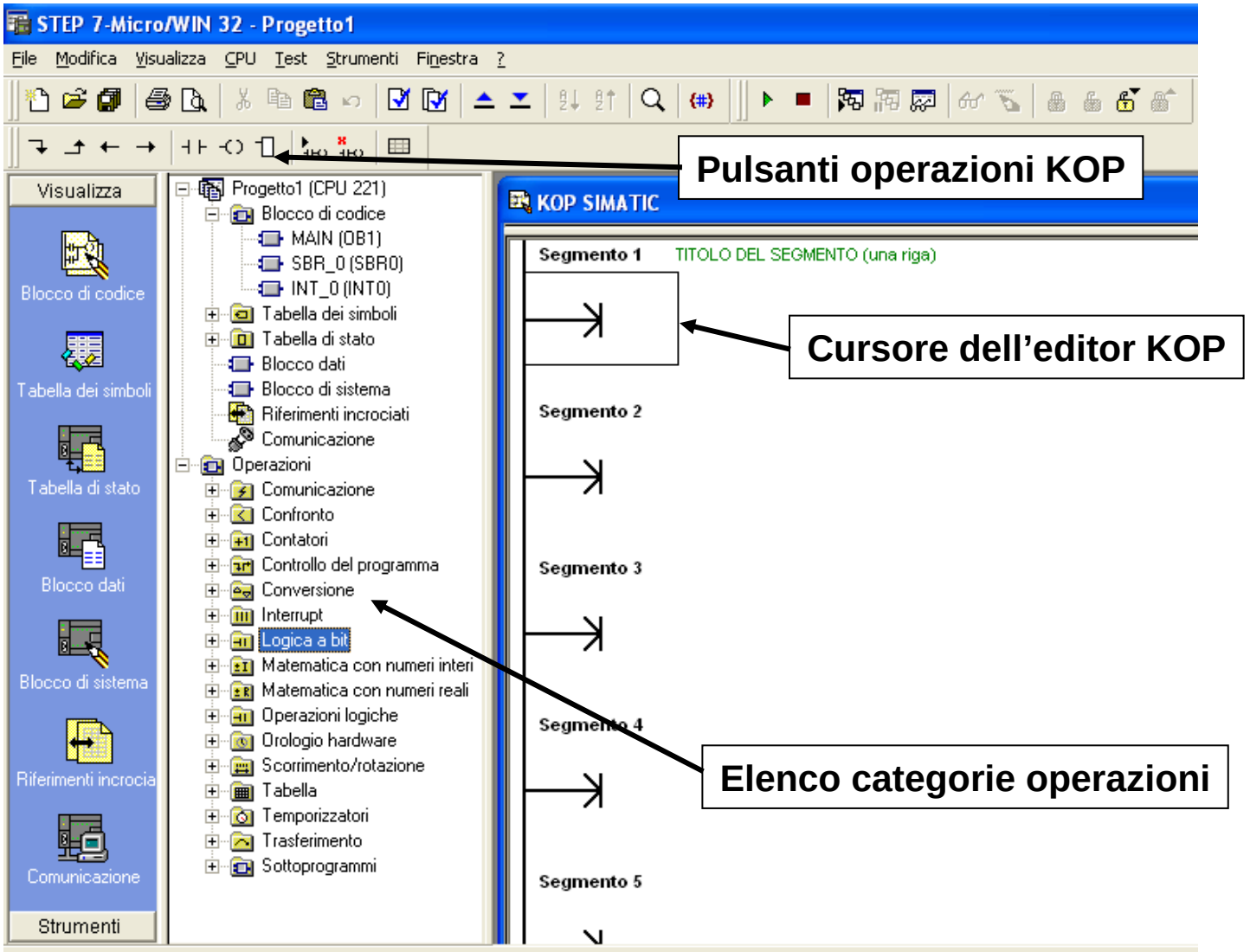
- Selezionare la prima cella nella colonna Nome simbolico e digitare: Start_1
- Premere INVIO per spostare la selezione alla prima cella della colonna Indirizzo. Digitare l'indirizzo **I0.0** e premere INVIO. La selezione viene spostata alla cella nella colonna Commento. (I commenti sono opzionali, ma comunque consigliabili per individuare gli elementi del programma)
- Premere INVIO per avviare la successiva riga della colonna Nome simbolico, e ripetere i passi descritti per ognuno dei rimanenti nomi simbolici e indirizzi
- Utilizzare il comando **Salva** per salvare la tabella dei simboli

Tabella dei simboli per il programma di esempio

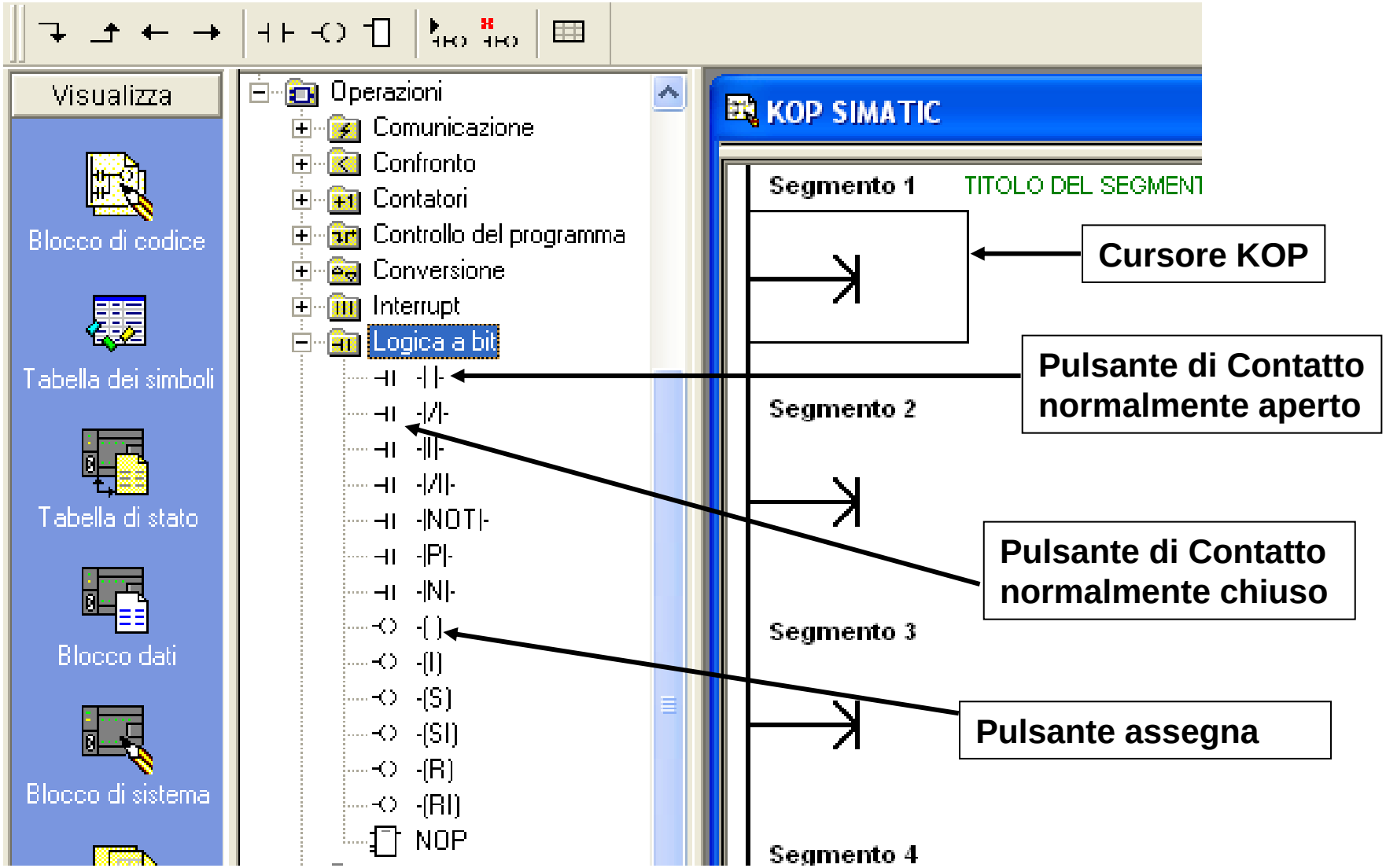
Tabella dei simboli

	Nome	Indirizzo	Commento
1	CONTATORE_CICLI	C30	CONTA SOMMA CICLI DI MESCOLATURA E RISCALDAMENTO COMPLETATI
2	START_1	I0.0	INTERRUTTORE DI AVVIO PER VERNICE 1
3	START_2	I0.1	INTERRUTTORE DI AVVIO PER VERNICE 2
4	STOP_1	I0.2	INTERRUTTORE DI ARRESTO PER VERNICE 1
5	STOP_2	I0.3	INTERRUTTORE DI ARRESTO PER VERNICE 2
6	ALTO_LIVELLO	I0.4	INTERRUTTORE LIMITAZIONE LIVELLO MASSIMO SERBATOIO
7	BASSO_LIVELLO	I0.5	INTERRUTTORE LIMITAZIONE LIVELLO MINIMO SERBATOIO
8	RESET	I0.6	CONTROLLO DI RESET PER CONTATORE
9	RAGG_ALTO_LIVELLO	M0.1	MERKER
10	POMPA_1	Q0.0	POMPA PER VERNICE 1
11	POMPA_2	Q0.1	POMPA PER VERNICE 2
12	MOTORE_MESCOLATORE	Q0.2	MOTORE PER MESCOLATORE
13	VALVOLA_VAPORE	Q0.3	VAPORE PER RISCALDARE LA MISCELA NEL SERBATOIO
14	VALVOLA_SCARICO	Q0.4	VALVOLA CHE PERMETTE LO SCARICO DELLA MISCELA
15	POMPA_SCARICO	Q0.5	POMPA CHE PERMETTE SCARICO DELLA MISCELA DAL SERBATOIO
16	TEMPORIZZATORE	T37	TEMPORIZZATORE CONTROLLO MESCOL. / RISCALDAMENTO MISCELA

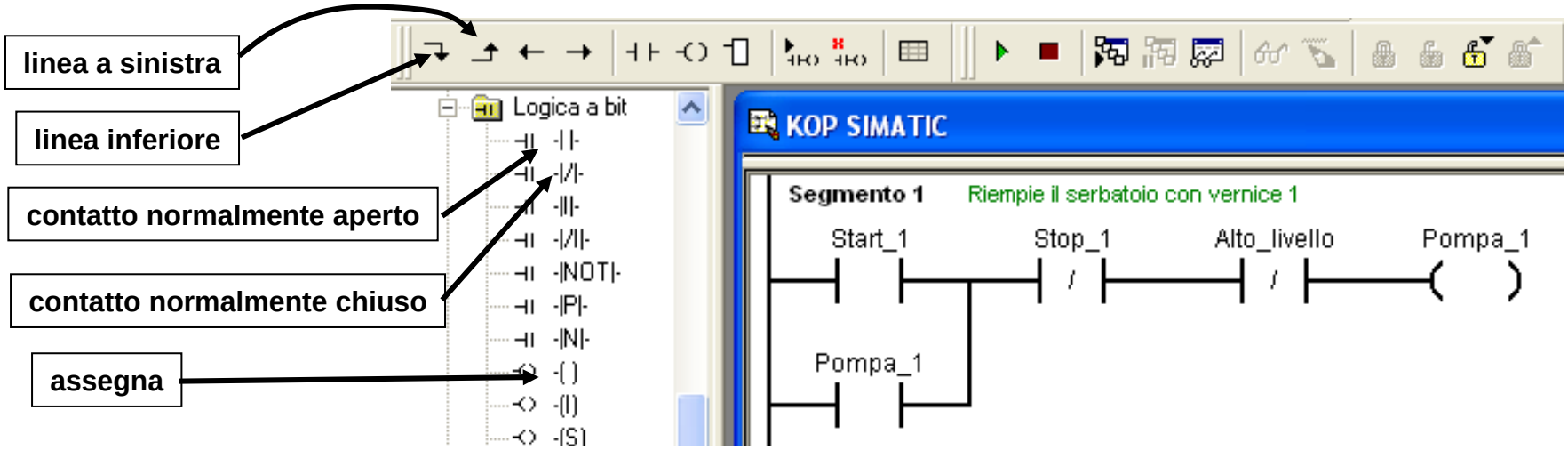
Introduzione del programma in KOP



Operazioni KOP

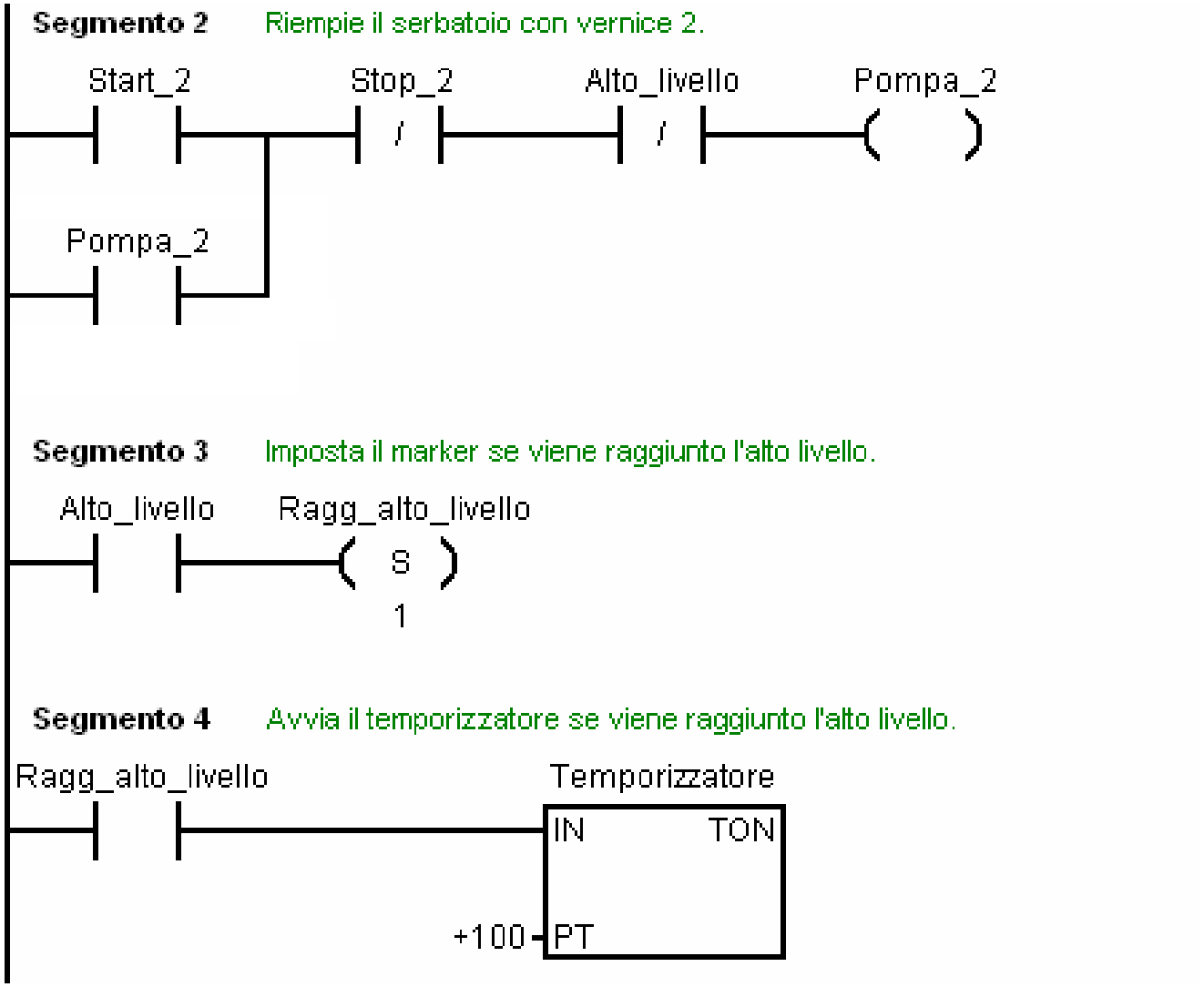


Primo segmento

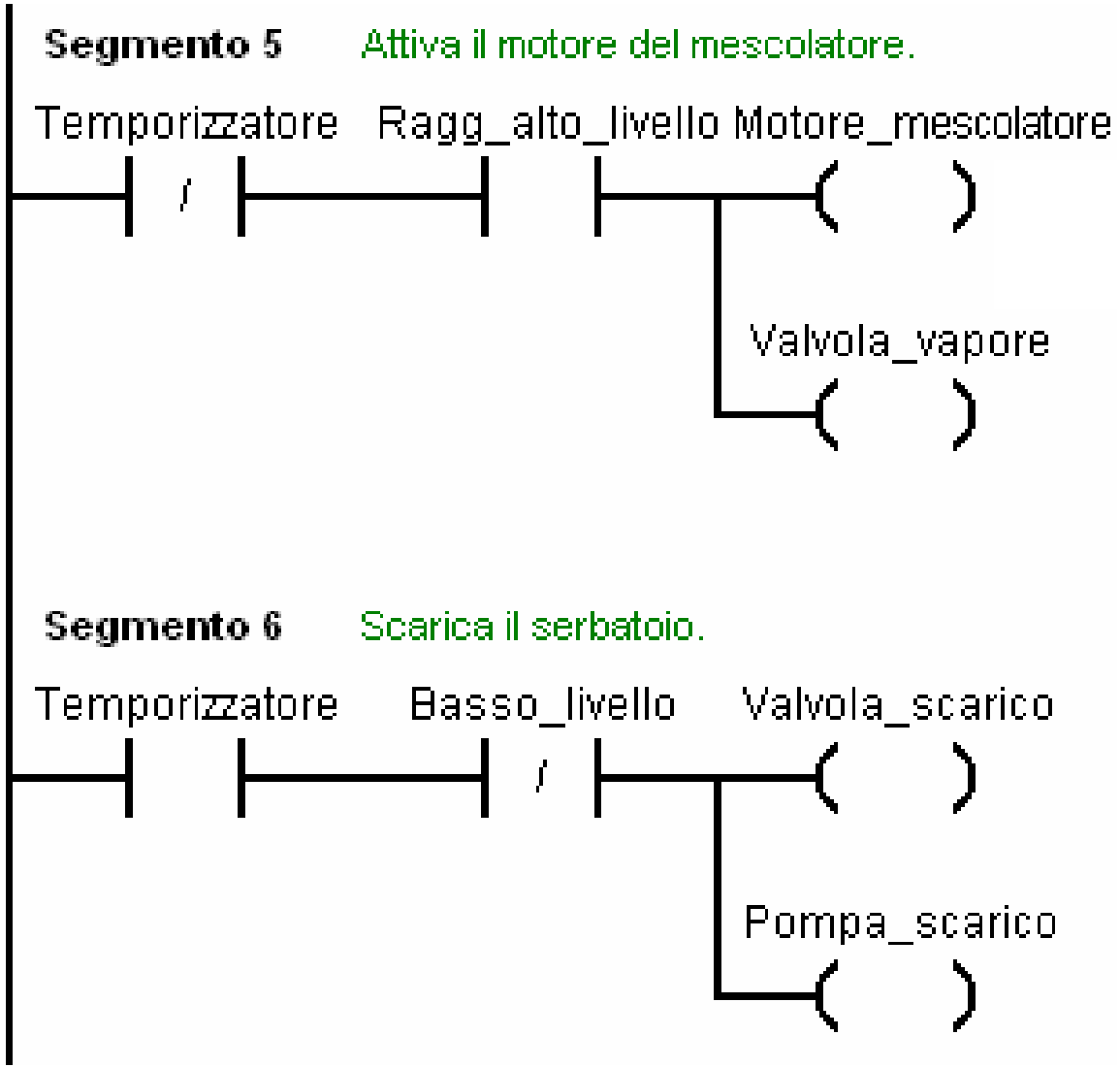


1. Fare doppio clic sopra o accanto all'etichetta del segmento con il numero per accedere al campo del titolo dell'editor di commenti
2. Spostare il cursore KOP nella prima posizione della colonna a sinistra
3. Selezionare il contatto normalmente aperto scegliendo "Operazioni a bit" dalla lista delle categorie di istruzioni e quindi selezionando con doppio click "Contatto normalmente aperto" dalla lista delle istruzioni; "Start_1" è il primo elemento richiesto per il segmento 1, premere INVIO per confermare il primo elemento ed il suo nome simbolico; il cursore KOP si sposta alla seconda posizione nella colonna
4. Selezionare con doppio clic un contatto normalmente chiuso, digitare il nome simbolico "Stop_1" e premere il tasto INVIO, il cursore si sposta sulla colonna successiva
5. Selezionare con doppio clic un contatto normalmente chiuso, digitare il nome simbolico "Alto_livello" e premere INVIO
6. Selezionare con doppio clic il pulsante assegna e digitare il nome simbolico "Pompa_1", premere INVIO per confermare la bobina e il suo nome simbolico evidenziato sopra di essa
7. Utilizzare il mouse o premere il tasto di freccia sinistra per riportare il cursore al primo elemento del segmento corrente
8. Fare clic sul pulsante di linea inferiore e tracciare una linea verticale tra il primo e il secondo contatto e poi fare click sul pulsante linea a sinistra; selezionare con doppio clic un contatto normalmente aperto e digitare il nome simbolico "Pompa_1" e premere il tasto INVIO

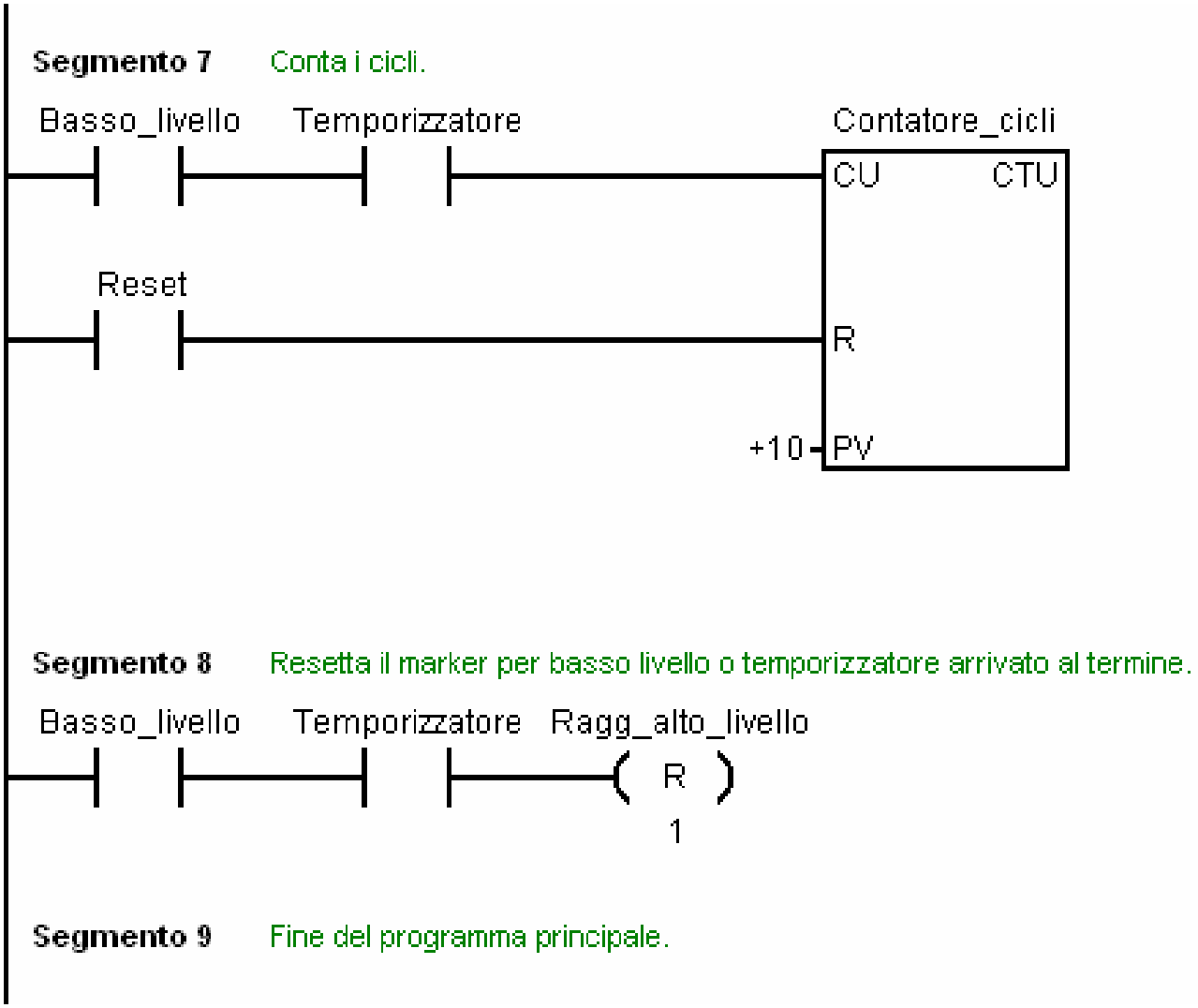
2°, 3°, 4° segmento



5° e 6° segmento

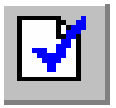


7° e 8° segmento

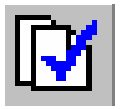


Compila (CPU > Compila)

Fare clic sul pulsante Compila o selezionare il comando di menu CPU > Compila per compilare la finestra attiva (blocco di codice o blocco dati)

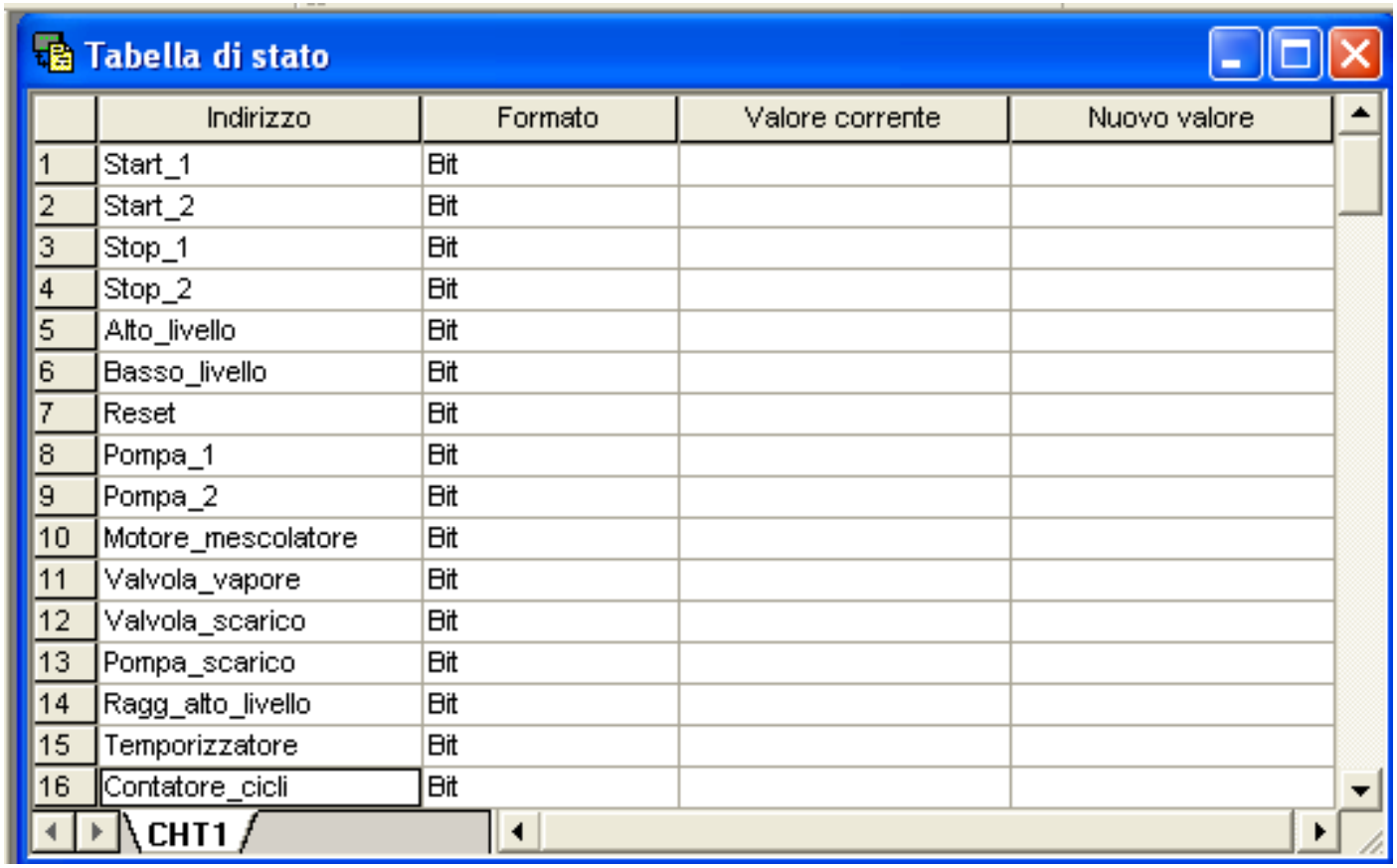


Fare clic sul pulsante Compila tutto o selezionare il comando di menu CPU > Compila tutto per compilare i componenti del progetto (blocco di codice, blocco dati e blocco di sistema)



Creazione di una tabella di stato

Per controllare lo stato degli elementi selezionati nel programma di esempio si costruirà una tabella di stato contenente gli elementi da controllare mentre viene eseguito il programma



Caricamento nella CPU e controllo del programma

Caricare a questo punto il programma nella CPU e porre la CPU nello stato di funzionamento RUN

Per controllare e eliminare gli errori di funzionamento del programma utente si possono utilizzare le funzioni di test.

Modi operativi della CPU

La CPU ha due modi operativi: STOP e RUN

In STOP è possibile creare e modificare il programma, ma non eseguirlo.

In RUN è invece possibile eseguire il programma e crearne, modificarne e controllarne il funzionamento e i dati.

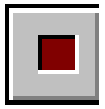
Alcune funzioni di test consentono di migliorare il controllo del programma e di identificare gli errori di programmazione.

Le funzioni di test, quali le funzioni di primo ciclo o di più cicli, possono essere utilizzate in STOP e determinano un passaggio da STOP a RUN per il numero di cicli specificato.

Mettere in STOP la CPU

Prima di caricare il progetto nella CPU, assicurarsi che la CPU si trovi nello stato di STOP eseguendo uno dei seguenti passi:

- Fare clic sul pulsante STOP per attivare il modo STOP
- Selezionare il comando di menu CPU > STOP per attivare il modo STOP
- Modificare manualmente l'interruttore dei modi operativi della CPU
- Inserire nel programma un'operazione di STOP

STOP 

Caricare il progetto nella CPU

Fare clic sul pulsante Carica nella CPU della barra degli strumenti o scegliere CPU>Carica nella CPU per visualizzare la finestra di dialogo "Carica nella CPU“.

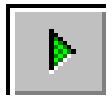
La prima volta che si utilizza il comando Carica nella CPU, per default sono selezionati i pulsanti di scelta del blocco di codice, del blocco dati e della configurazione della CPU (il blocco di sistema): se non si vuole caricare un blocco particolare, disattivare il relativo pulsante.

Avviare la procedura di caricamento facendo clic sul pulsante "OK“.

Se il caricamento viene effettuato correttamente, compare una finestra di conferma con il seguente messaggio: "Caricamento nella CPU eseguito correttamente“, e per poterlo eseguire nella CPU è necessario impostare la CPU da STOP a RUN.

Fare clic sul pulsante della barra degli strumenti o scegliere CPU>RUN per riportare la CPU in RUN.

Carica nella CPU 

CPU>RUN 

Errore di caricamento

Se il valore del tipo di CPU impostato non è adatto alla CPU utilizzata, compare una finestra con il seguente messaggio: "Il tipo di CPU selezionata per il progetto non corrisponde al tipo di CPU remota. Continuare il caricamento nella CPU?"

Per correggere il tipo di CPU, scegliere "No" e arrestare la procedura di caricamento

Selezionare CPU>Tipo nella barra dei menu per visualizzare la finestra di dialogo "Tipo CPU"

Per fare in modo che venga letto automaticamente il tipo corretto, selezionarlo dall'elenco a discesa oppure facendo clic sul pulsante "Leggi CPU"

Fare clic sul pulsante "OK" per confermare il tipo di CPU e chiudere la finestra di dialogo

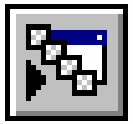
Riavviare la procedura di caricamento facendo clic sul pulsante Carica nella CPU della barra degli strumenti o scegliendo CPU>Carica nella CPU dalla barra dei menu

Controllo dello stato KOP

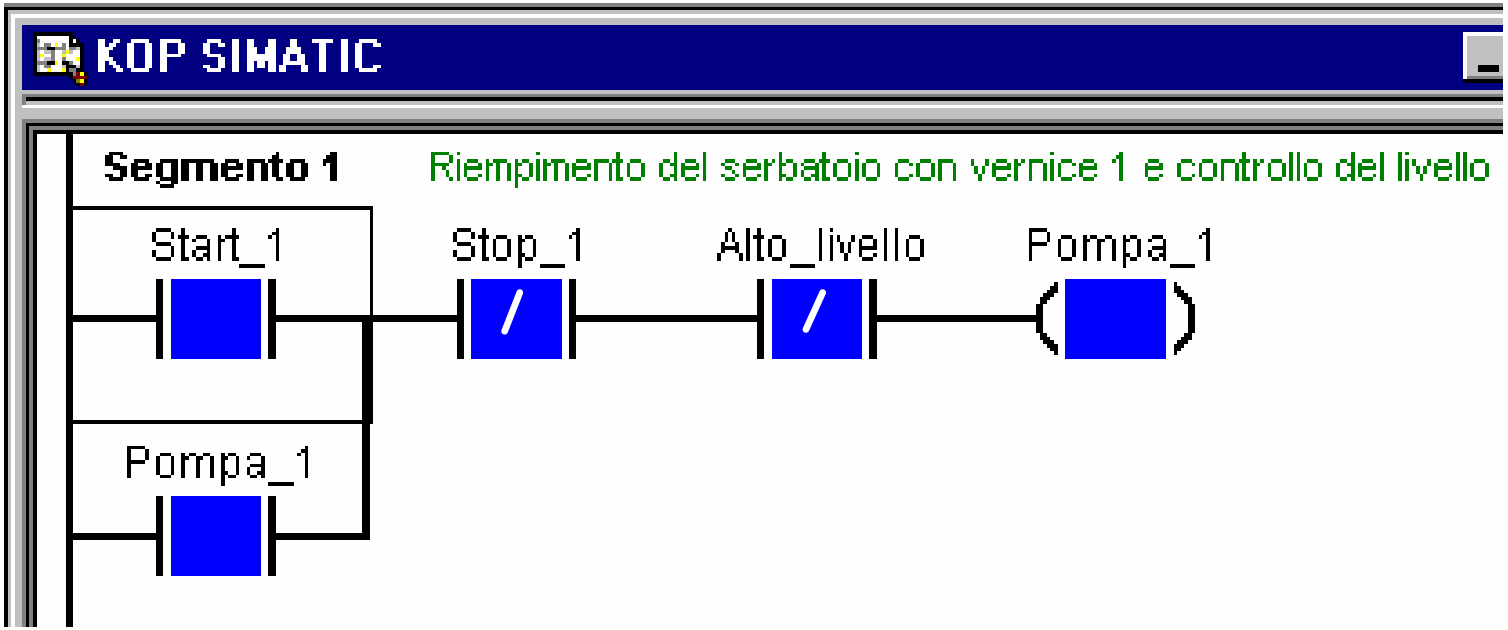
Lo stato KOP visualizza lo stato corrente degli eventi nel programma utente: riaprire se necessario la finestra dell'editor KOP e fare click sul pulsante Stato del programma nella barra degli strumenti di test oppure selezionare il comando di menu Test>Stato del programma.

Se si dispone di un simulatore di ingressi collegato ai terminali di ingresso sulla CPU, si potranno attivare gli interruttori per verificare il flusso di corrente e l'esecuzione della logica.

Se, ad esempio, si attivano gli interruttori I0.0 e I0.2 e l'interruttore di I0.4 ("Alto_livello") è disattivato, il flusso di corrente di Segmento 1 è completo: Il segmento sarà simile a quello seguente:

Stato del programma 

Flusso di corrente del Segmento 1



- **La colorazione del "flusso di corrente" non sempre significa che c'è passaggio di corrente**
- **Poiché lo stato del programma in KOP riferisce solo i valori di fine scansione, a volte può essere difficile interpretare il significato della rappresentazione di un "flusso di corrente“**
- **Nello stato del programma KOP i contatti e le bobine booleani vengono rappresentati a colori in base al valore del relativo operando a bit. Se il valore a bit è uguale a 1 (il bit è on), l'operazione viene colorata. Ciò non significa tuttavia che l'operazione è stata eseguita.**

ESERCIZI

Avviamento di un motore

Conoscenza della funzione di autoritenuta ed arresto macchina

- Compilazione delle liste d' occupazione degli ingressi e delle uscite
- Stesura dello schema elettrico
- Stesura del Flow-chart o stesura del Grafcet
- Stesura del programma
- Costruzione e prova del circuito

Descrizione

Premendo il pulsante di MARCIA, il motore parte, lasciando il pulsante il motore rimane in moto, premendo il pulsante STOP il motore si arresta.

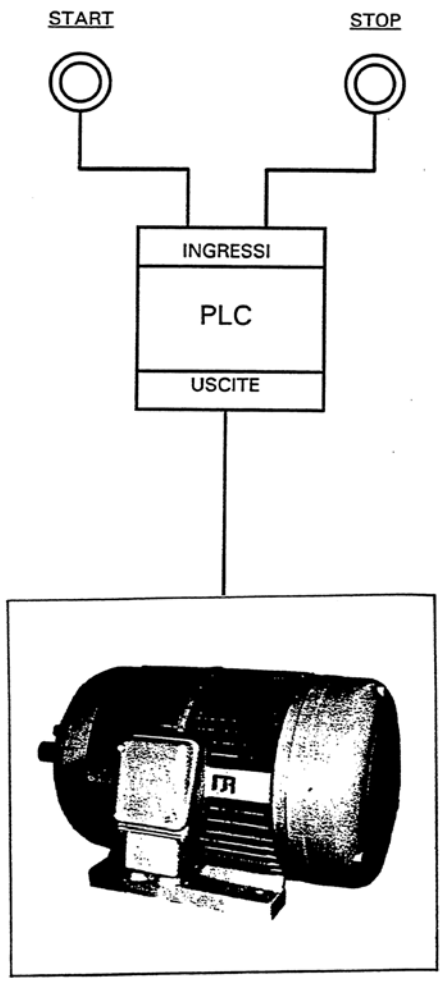
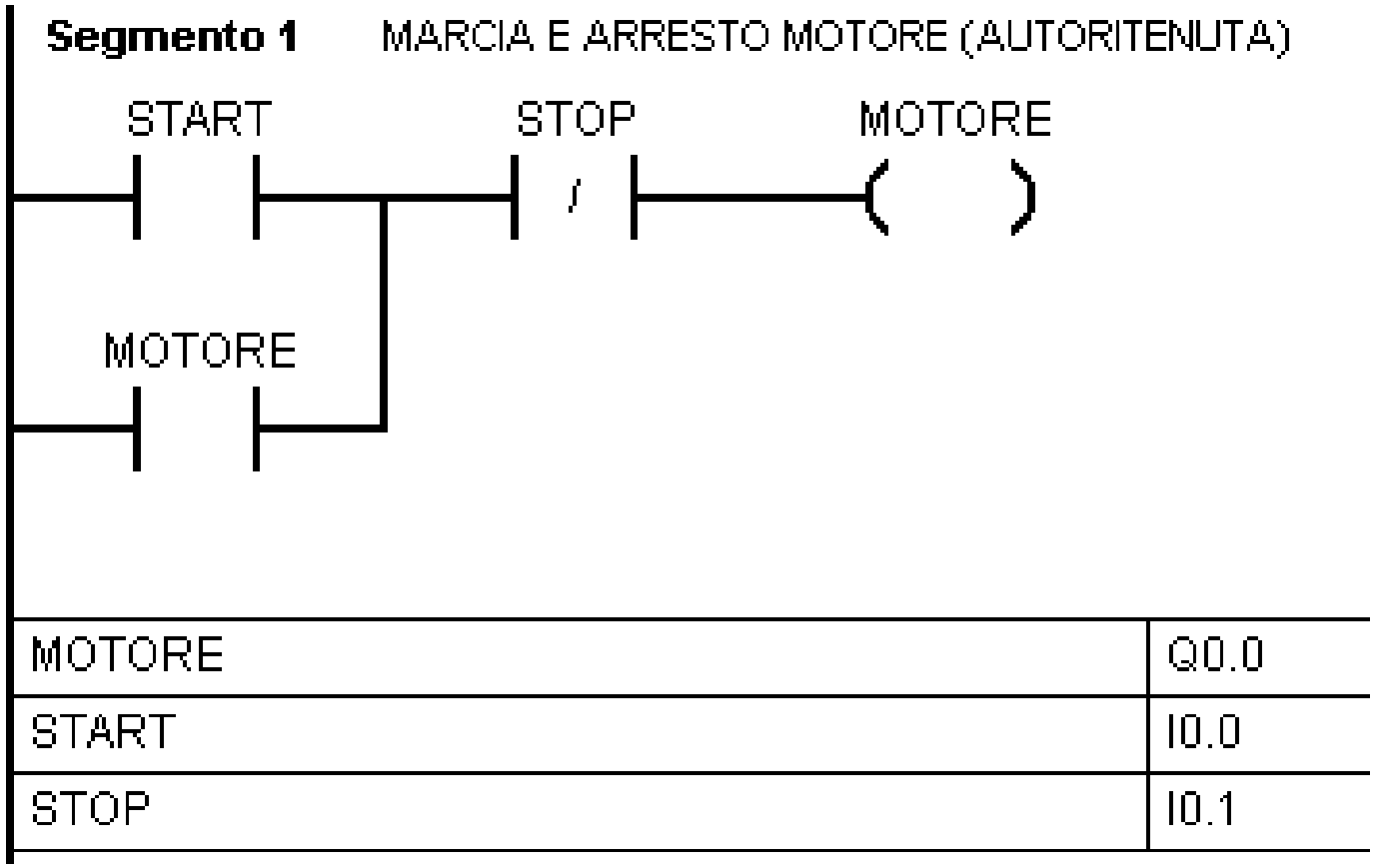


Tabella dei simboli

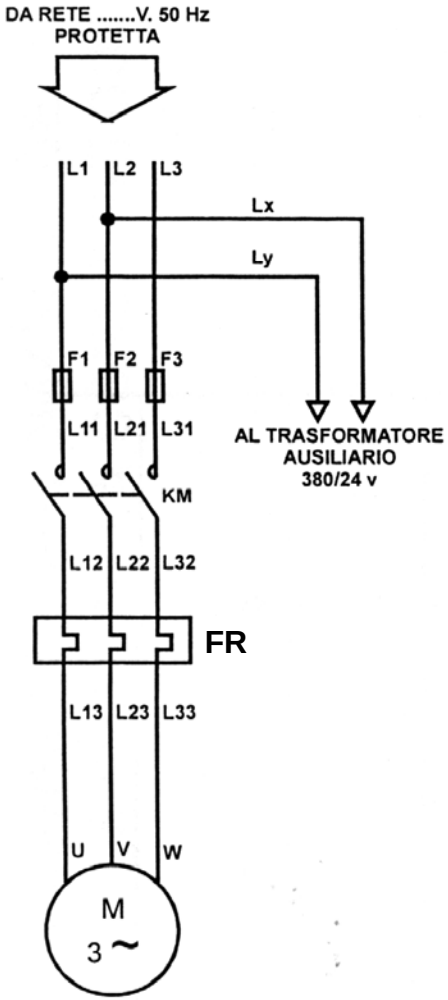
Tabella dei simboli

	Nome	Indirizzo	Commento
1	START	I0.0	
2	STOP	I0.1	
3	MOTORE	Q0.0	
4			

Diagramma KOP

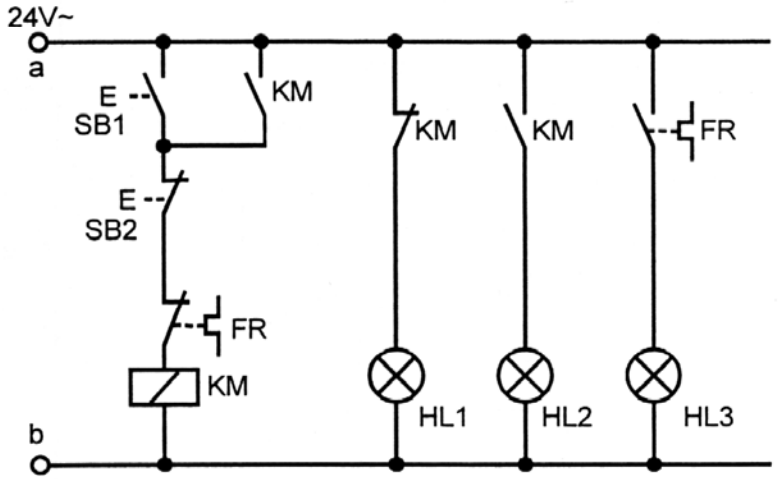


Avviamento diretto dalla linea di un motore asincrono trifase con rotore a gabbia di scoiattolo



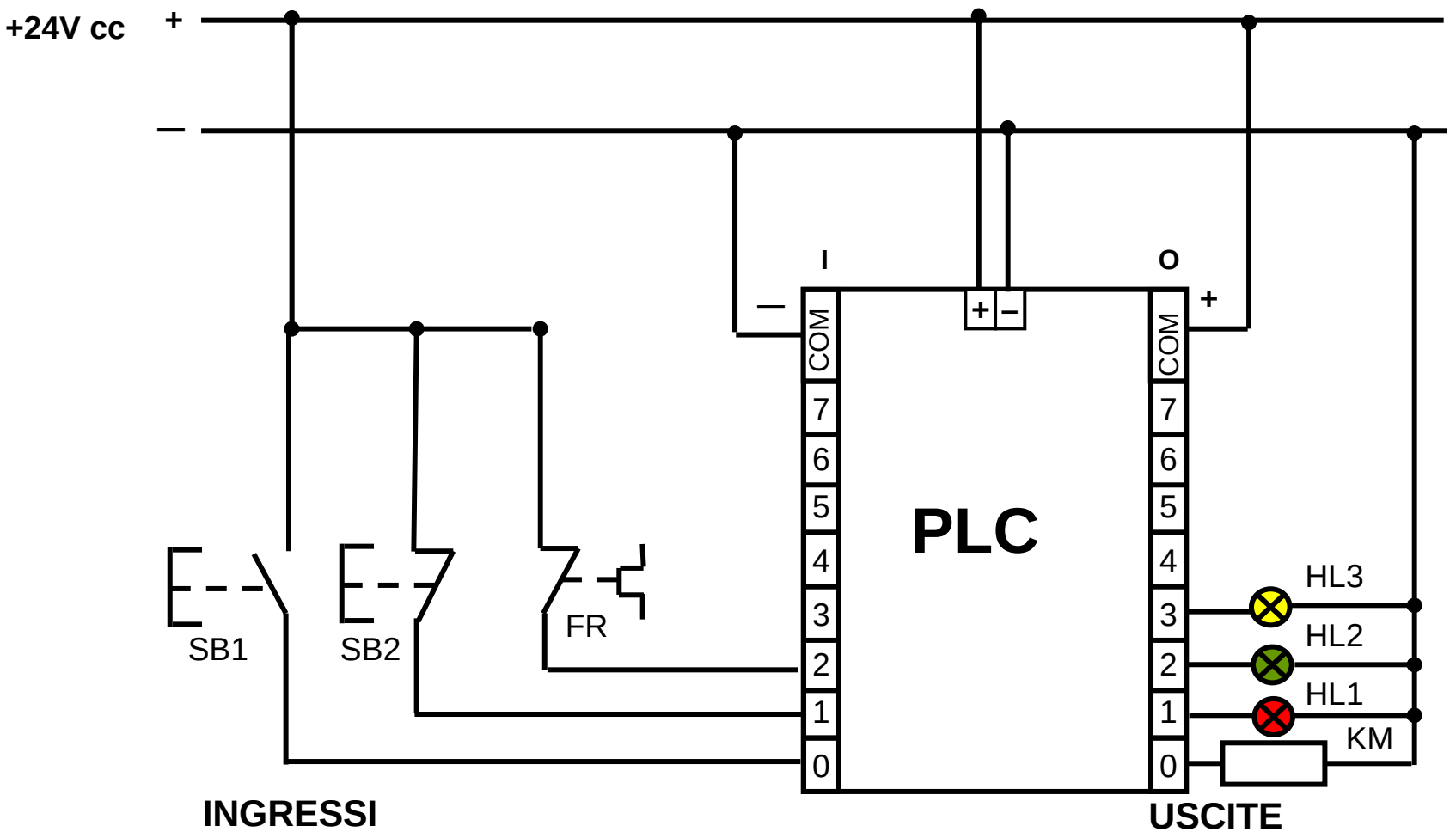
Schema di potenza

- **Chiusura del contattore ZM mediante l'azionamento del pulsante SB1 di marcia**
- **Autoalimentazione di KM per mezzo del contatto NA del KM**
- **Segnalazione di KM diseccitato per mezzo del contatto NC del KM (luce verde)**
- **Segnalazione di KM eccitato per mezzo del contatto NA del KM (luce rossa)**
- **Segnalazione dell'intervento del relè termico mediante il contatto NA del relè termico stesso (luce gialla intermittente)**
- **Apertura del contattore mediante l'azionamento del pulsante SB2 di arresto**



Schema di comando

Avviamento diretto dalla linea di un motore asincrono trifase con rotore a gabbia di scoiattolo



INGRESSI

Pulsante di marcia	I0.0
Pulsante di arresto	I0.1
Contatto relè termico	I0.2

USCITE

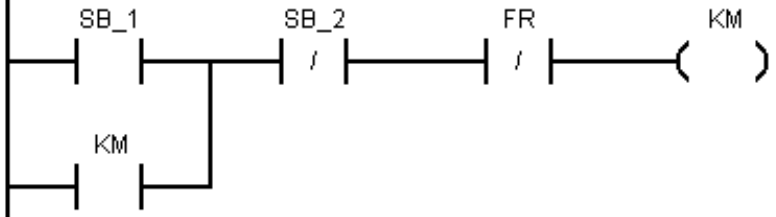
Contattore KM	Q0.0
Lampada rossa	Q0.1
Lampada verde	Q0.2
Lampada gialla lampeggiante	Q0.3

Tabella dei simboli

Nome	Indirizzo	Commento
SB_1	I0.0	PULSANTE DI MARCIA
SB_2	I0.1	PULSANTE DI ARRESTO
FR	I0.2	RELE' TERMICO
KM	Q0.0	CONTATTORE
HL1	Q0.1	LAMPADA ROSSA
HL2	Q0.2	LAMPADA VERDE
HL3	Q0.3	LAMPADA GIALLA LAMPEGGIANTE

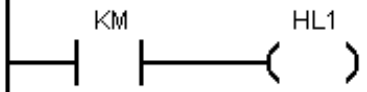
Diagramma KOP

Segmento 1 AVVIA IL CONTATTORE E MANTIENILO AUTOALIMENTATO



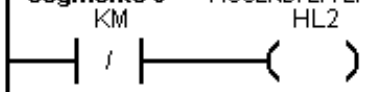
FR	I0.2	RELE' TERMICO
KM	Q0.0	CONTATTORE
SB_1	I0.0	PULSANTE DI MARCIA
SB_2	I0.1	PULSANTE D'ARRESTO

Segmento 2 ACCENDI LA LAMPADA ROSSA



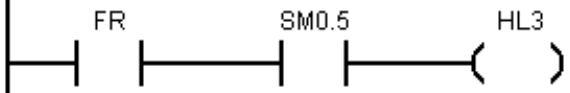
HL1	Q0.1	LAMPADA ROSSA
KM	Q0.0	CONTATTORE

Segmento 3 ACCENDI LA LAMPADA VERDE



HL2	Q0.2	LAMPADA VERDE
KM	Q0.0	CONTATTORE

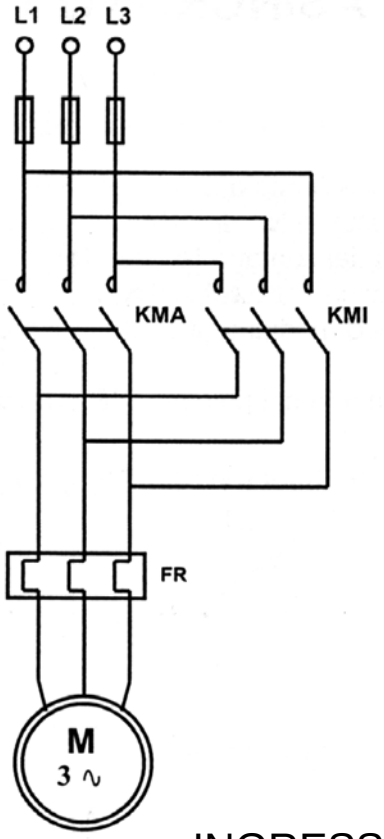
Segmento 4 ACCENDI LA LAMPADA GIALLA LAMPEGGIANTE

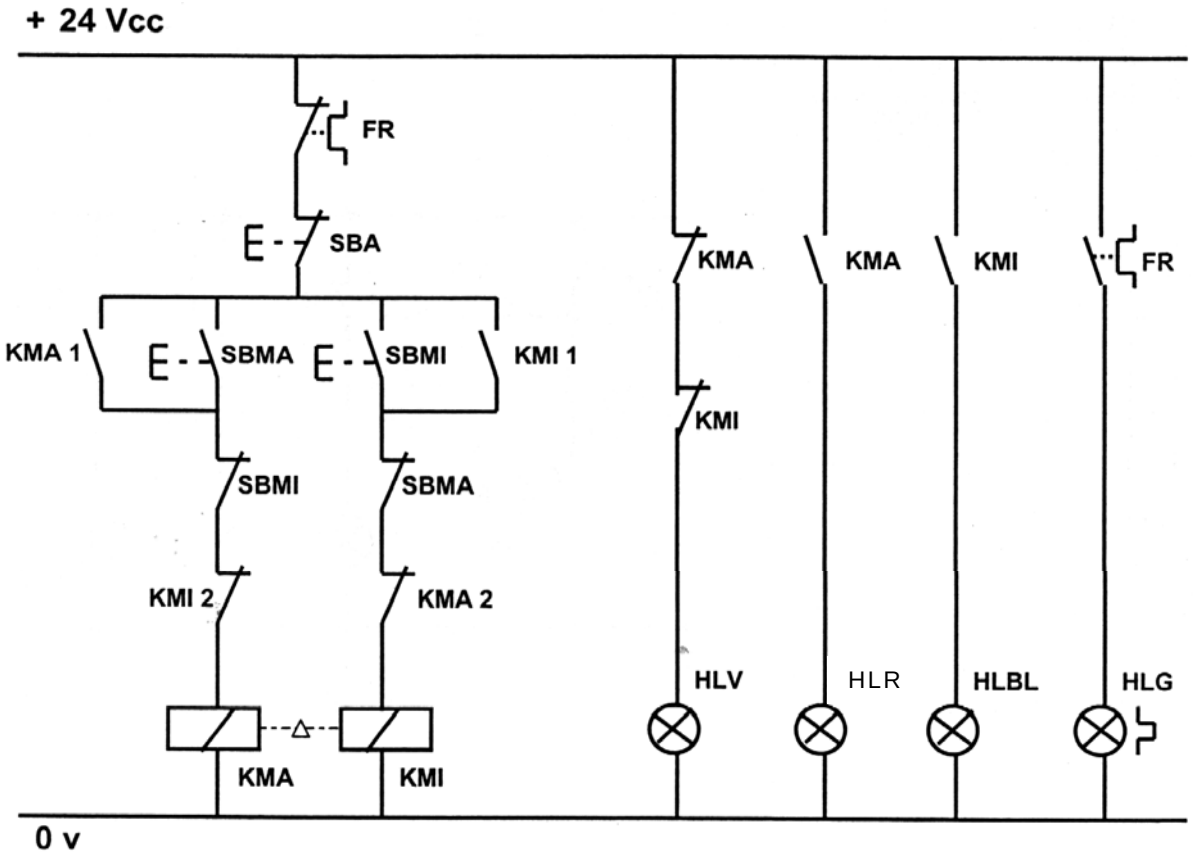


FR	I0.2	RELE' TERMICO
HL3	Q0.3	LAMPADA GIALLA LAMPEGGIANTE

Il relé termico è scattato per surriscaldamento all'accensione

Tele-inversione di marcia di un motore asincrono trifase



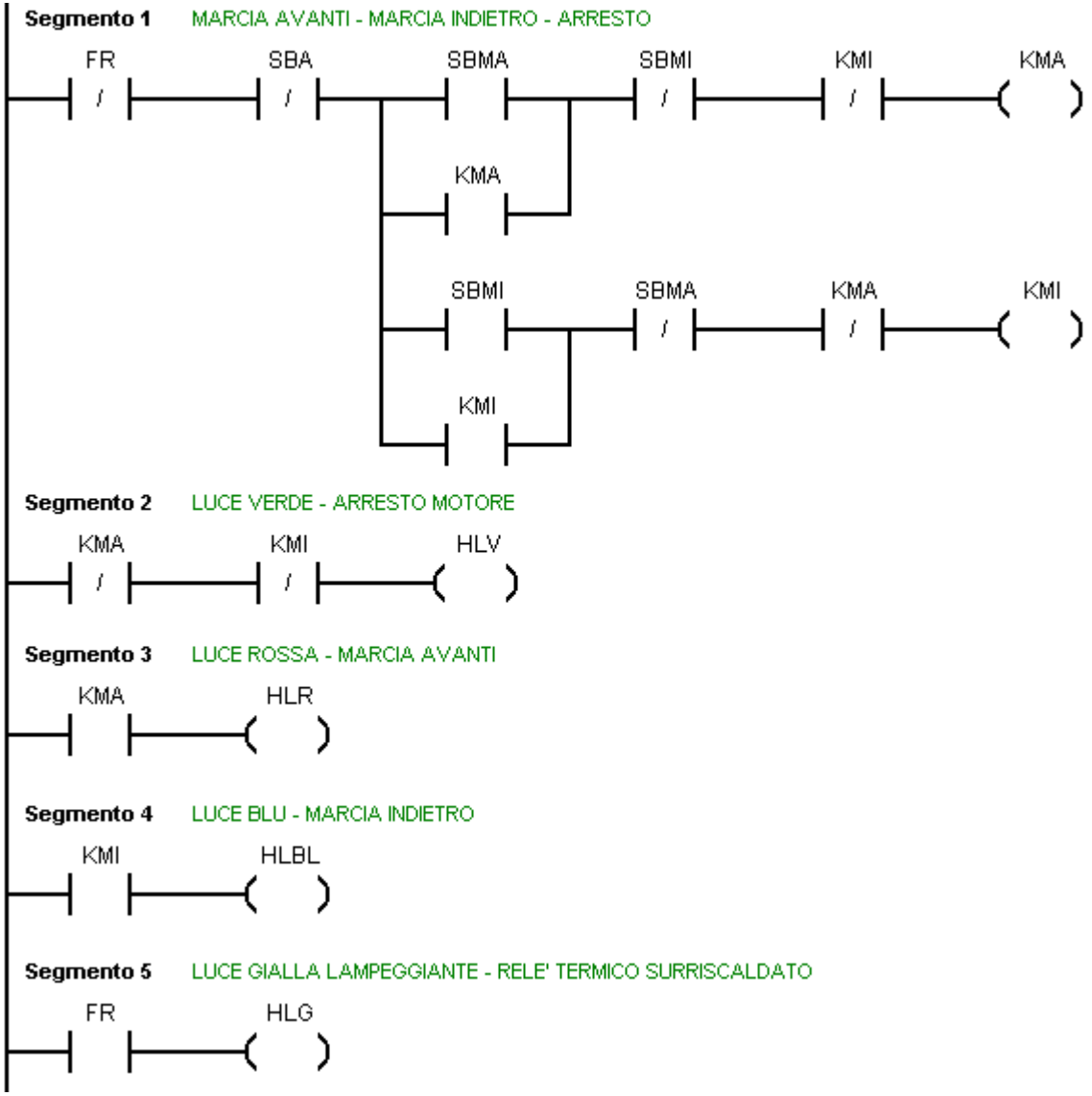


INGRESSI		USCITE	
Pulsante di marcia avanti	I0.0	Contattore marcia Avanti	Q0.0
Pulsante di marcia indietro	I0.1	Contattore marcia Indietro	Q0.1
Pulsante di arresto	I0.2		
Contatto relè termico	I0.3		

Tabella dei simboli

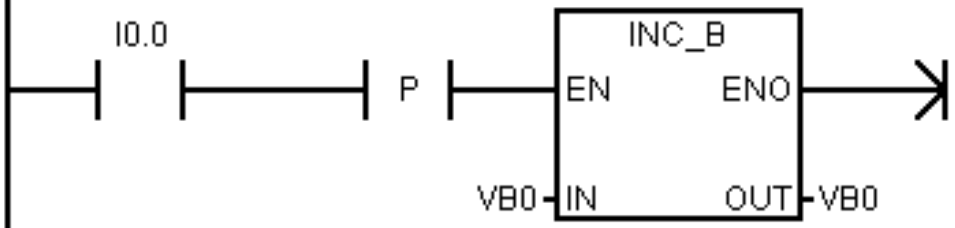
Nome	Indirizzo	Commento
SBMA	I0.0	PULSANTE DI MARCIA AVANTI
SBMI	I0.1	PULSANTE DI MARCIA INDIETRO
SBA	I0.2	PULSANTE DI ARRESTO
FR	I0.3	RELE' TERMICO
KMA	Q0.0	CONTATTORE MARCIA AVANTI
KMI	Q0.1	CONTATTORE MARCIA INDIETRO
HLR	Q0.2	LAMPADA ROSSA MARCIA IN AVANTI
HLBL	Q0.3	LAMPADA BLU MARCIA INDIETRO
HLV	Q0.4	LAMPADA VERDE ARRESTO
HLG	Q0.5	LAMPADA GIALLA LAMPEGGIANTE RELE' TERMICO

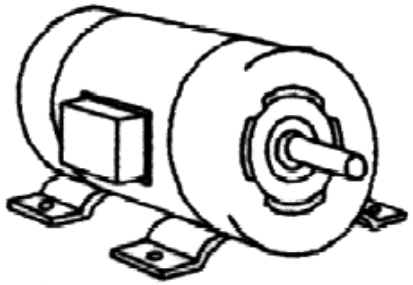
Diagramma KOP



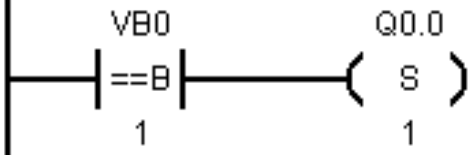
Avviamento e Arresto di un motore con un solo pulsante

Segmento 1 INCREMENTO DI UNA UNITÀ DEL CONTENUTO DEL BYTE ZERO

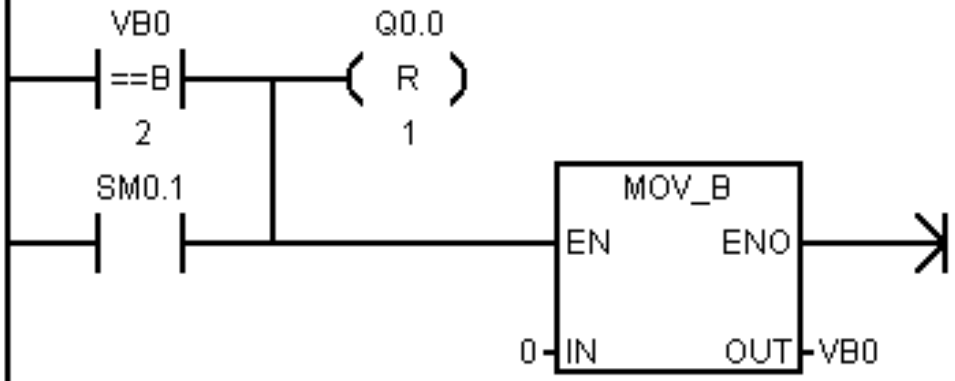


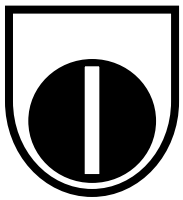


Segmento 2 COMPARAZIONE: SE VB0=1, ALLORA SET Q0.0



Segmento 3 IL BIT SM0.1 RIMANE A LIVELLO LOGICO 1 SOLO PER IL PRIMO CICLO

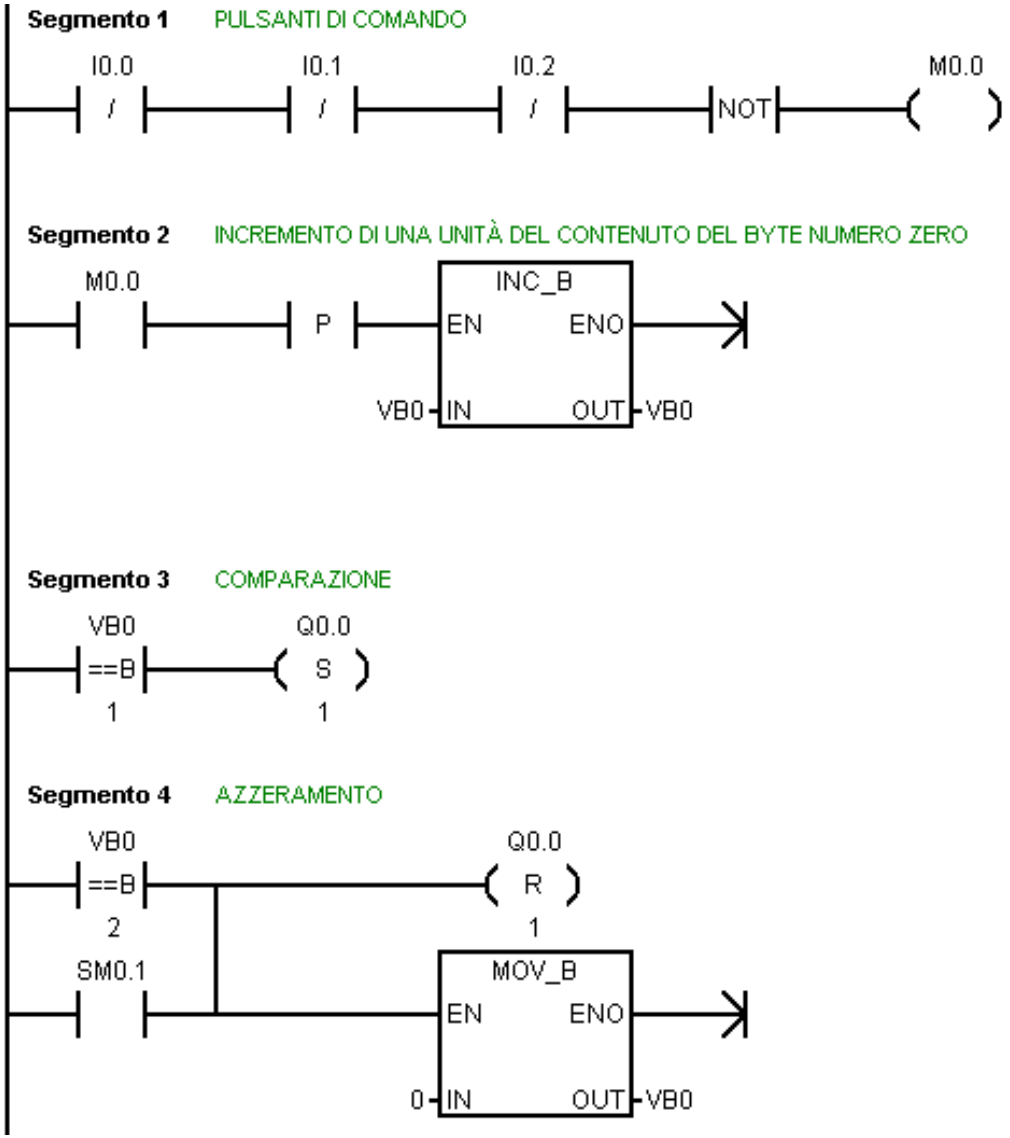




MARCIA

ARRESTO

Impianto luce di una invertita



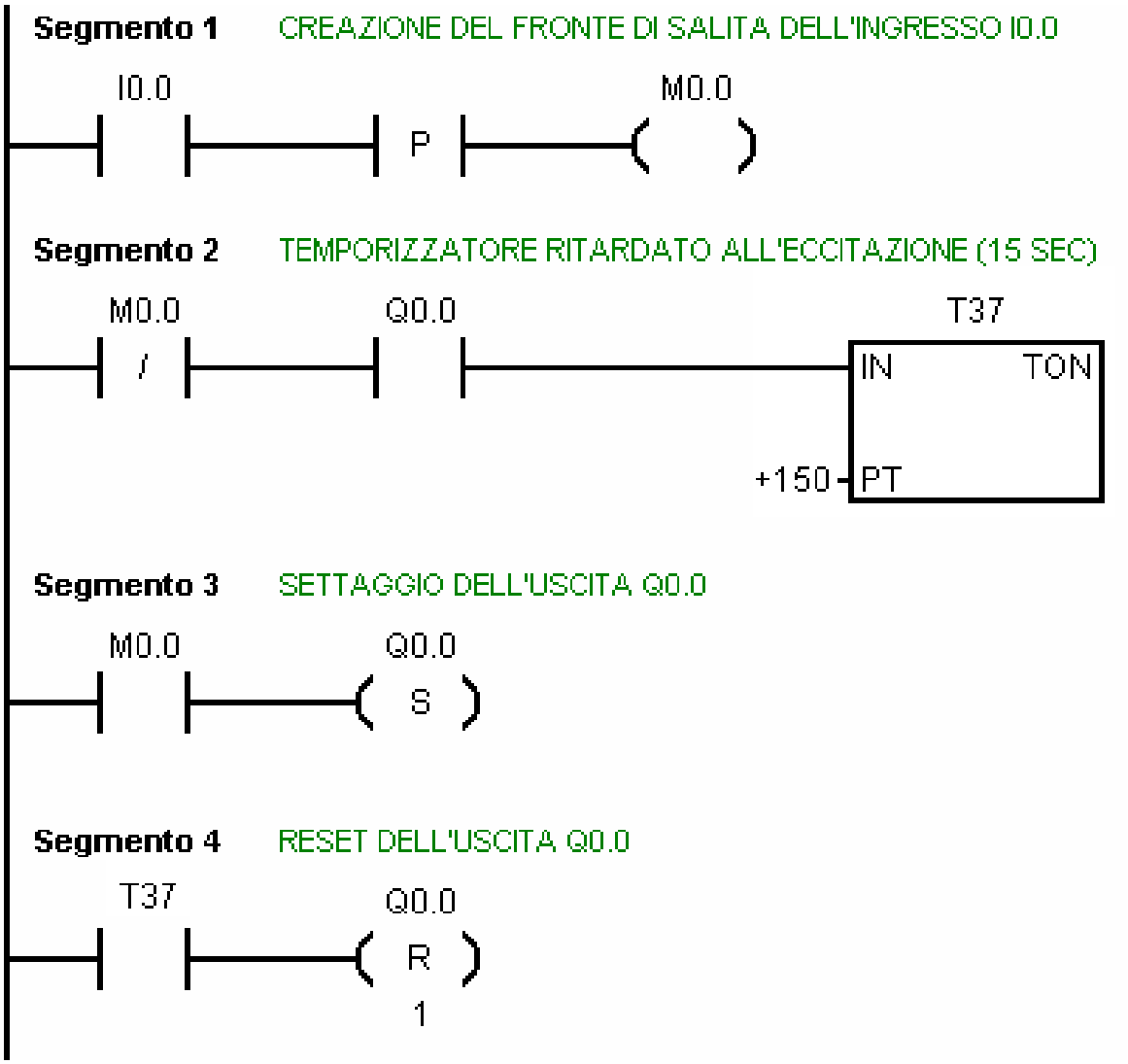
LA SERIE NEGATA DEGLI INGRESSI EQUIVALE AL PARALLELO NON NEGATO DI QUEST' ULTIMI

Impianto luce scale

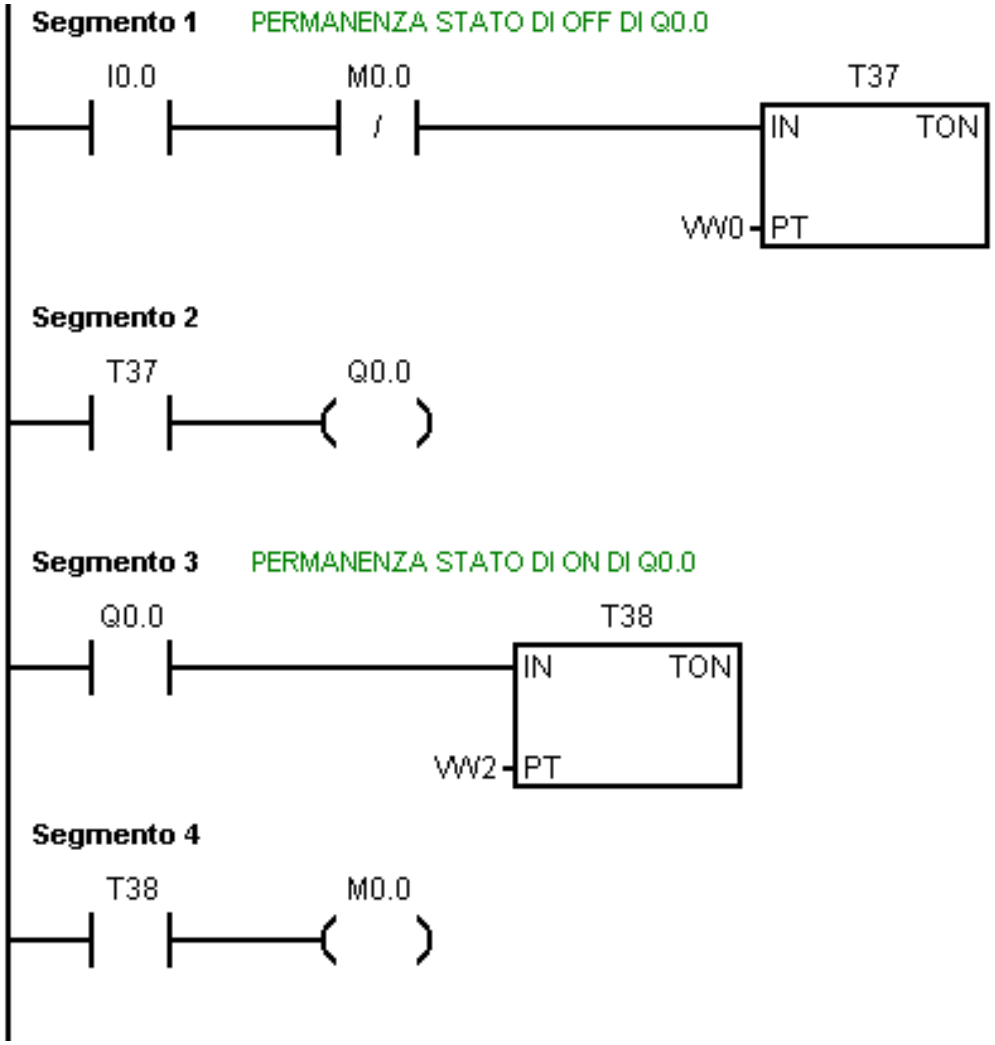
Premendo un qualsiasi pulsante si deve garantire l’accensione temporizzata di tutte le lampade costituenti l’impianto d’illuminazione.

Quando il ciclo di accensione è iniziato, ogni qualvolta che si ripreme un pulsante, il tempo viene automaticamente resettato mantenendo ovviamente accese le lampade fino alla scadenza della temporizzazione impostata.

Impianto luce scale



Oscillatore



Cronometro

Realizzare un programma che visualizzi i secondi e i minuti.

Il conteggio inizia quando viene chiuso il contatto di start e termina quando si riapre.

Deve essere possibile resettare il cronometro, chiudendo un contatto di azzeramento.

Porre in VW0 i secondi

e in VW2 i minuti.

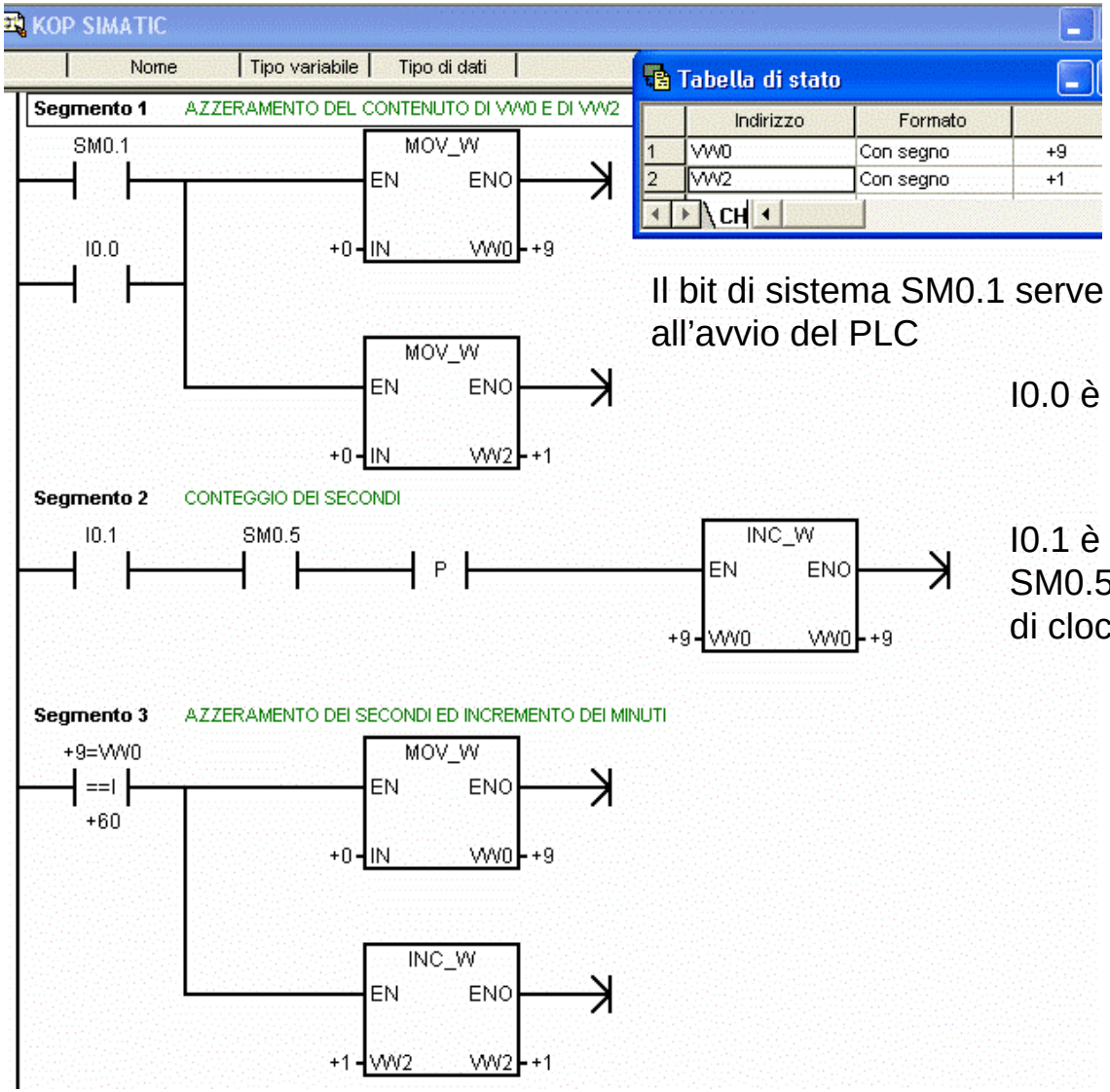
I0.1 = Start

I0.0 = Azzeramento

Rammentare i bit di stato:

- SM0.1 Questo bit è sempre ON per il primo ciclo di scansione. Viene utilizzato, ad esempio, per richiamare un sottoprogramma di inizializzazione.
- SM0.5 Questo bit mette a disposizione un impulso di clock di 1 secondo (on per 0,5 secondi, off per altri 0,5 secondi). Viene così fornito un tempo di ritardo facile da programmare o un impulso di clock di un secondo.

Cronometro



VISUALIZZARE IN VW0 I SEC
E IN VW2 I MINUTI

Il bit di sistema SM0.1 serve ad azzerare automaticamente all'avvio del PLC

I0.0 è l'azzeramento

I0.1 è lo START
SM0.5 è il bit di sistema generatore di clock con base tempi di 1 sec.

Scorrimento luce

Realizzare un dispositivo che simuli un movimento lineare di un bit sulle uscite del PLC.

Le uscite alimentano 6 lampadine.

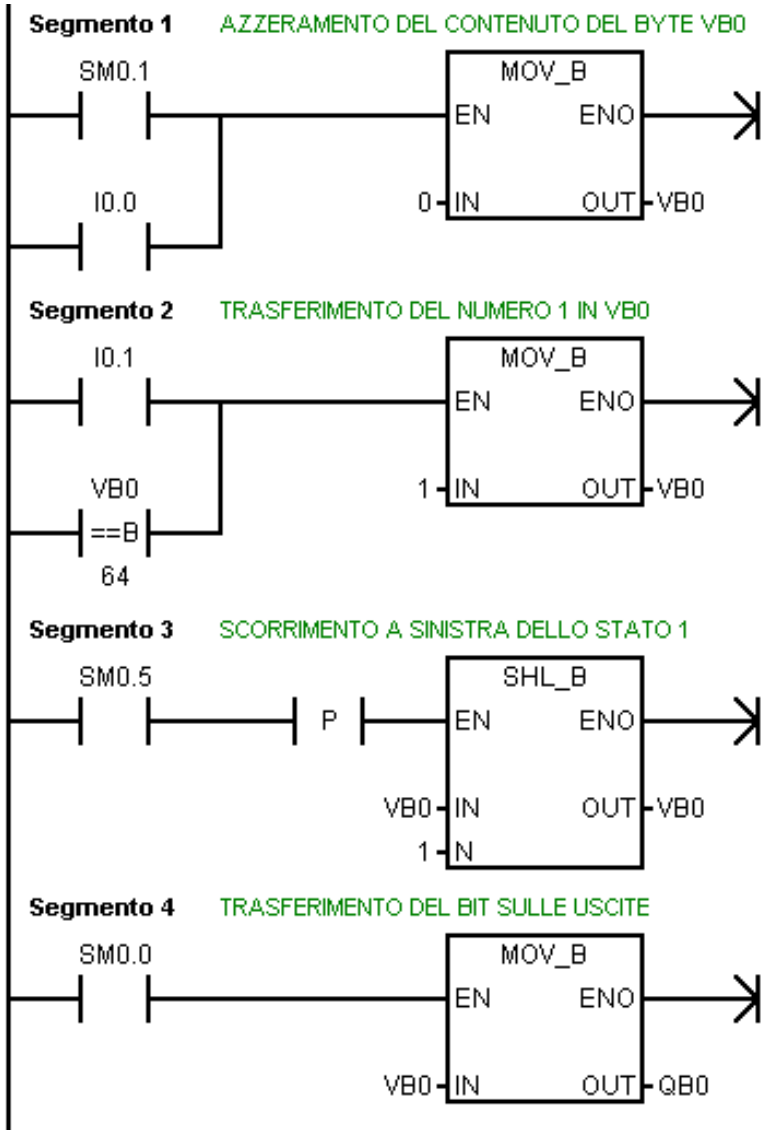
La luce quando arriva all'uscita Q0.5 deve rimbalzare e ripartire nella direzione opposta.

La posizione di partenza è associata all'indirizzo Q0.0.

Premendo il pulsante di Start deve partire lo scorrimento.

Premendo il pulsante di stop/reset tutte le uscite si dovranno disattivare.

Scorrimento luce



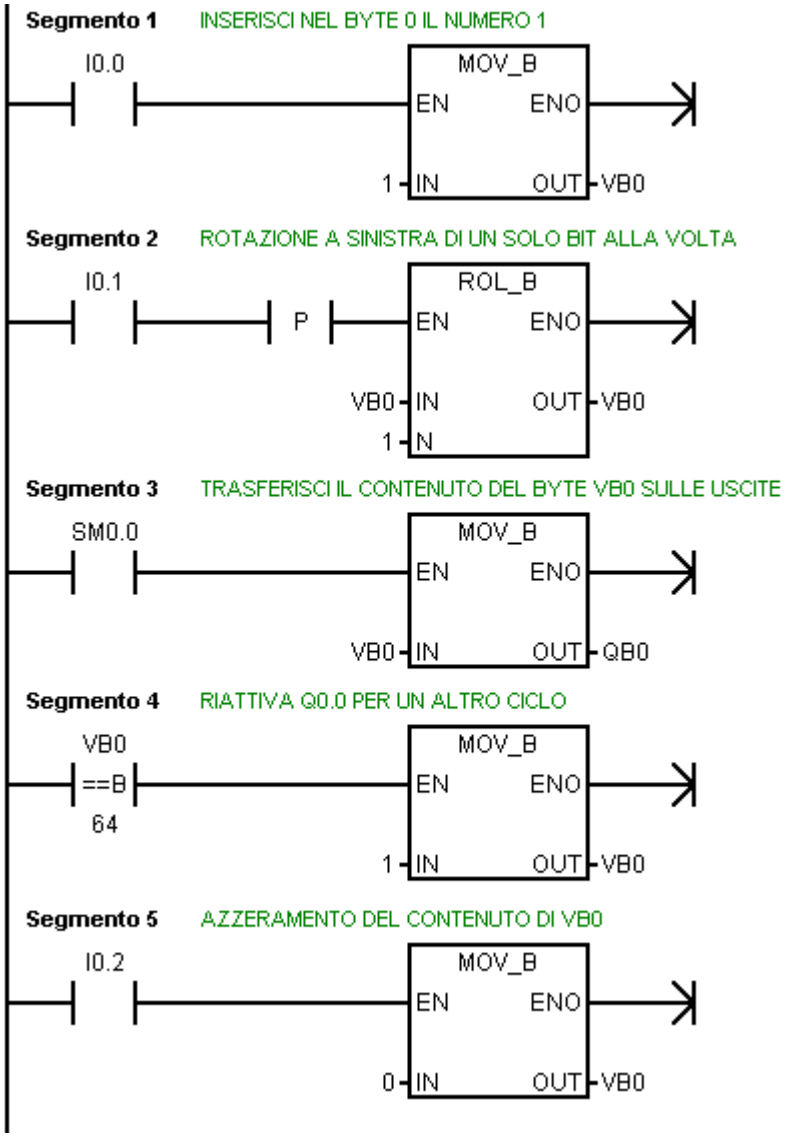
SM0.1 ha valore logico 1 solo al primo ciclo.

Il trasferimento avviene solo quando il contenuto di VB0 = 64 (1000000) oppure premendo il pulsante associato a I0.1

Il bit in ON viene fatto scorrere a sinistra con la cadenza di 1 sec.

SM0.0 è sempre in ON.

Scorrimento luce



ROTAZIONE A SINISTRA CON ATTIVAZIONE DELLE USCITE

Conteggio fronte di salita di un ingresso

VW0 conteggio corrente numero impulsi d'ingresso

VW2 secondi

VW4 memorizzazione numero d'impulsi d'ingresso

I0.1 fronte di salita

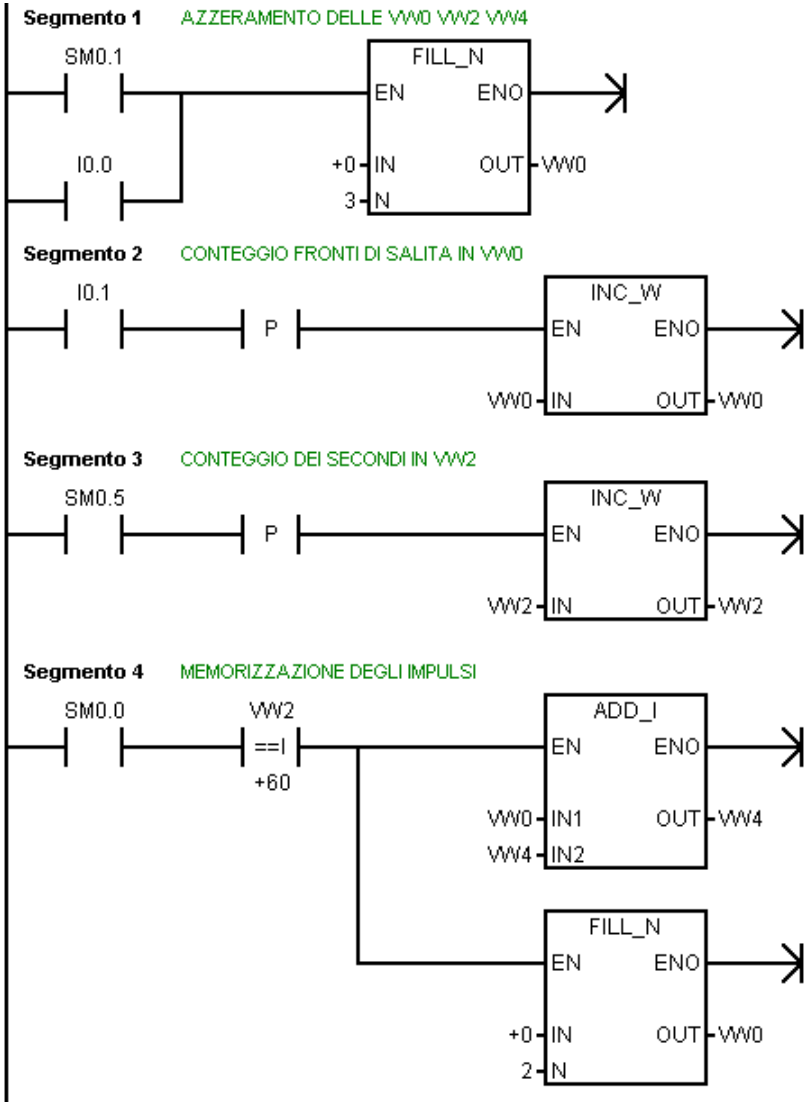
I0.0 azzeramento

Conteggiare i fronti di salita di un segnale d'ingresso (VW0)

Ad ogni minuto inserire tale valore in una prefissata area di memoria (VW4)

Contemporaneamente azzerare l'area predisposta al conteggio (VW0)

Conteggio fronte di salita di un ingresso



VW0 vengono incrementati gli impulsi
VW2 Contiene il numero dei secondi
VW4 Viene memorizzato il numero degli impulsi effettuati nell'arco di un minuto

Vengono memorizzati gli impulsi effettuati in un minuto e inoltre viene azzerato automaticamente il conteggio dei secondi e il conteggio parziale degli impulsi contenuto di VW0

Cambio preset di un timer

Si vuole cambiare il valore di Preset di un timer utilizzando gli ingressi di un PLC che scrivono sulla parola VW0 utilizzata per il PRESET

I0.0 azzera VW0

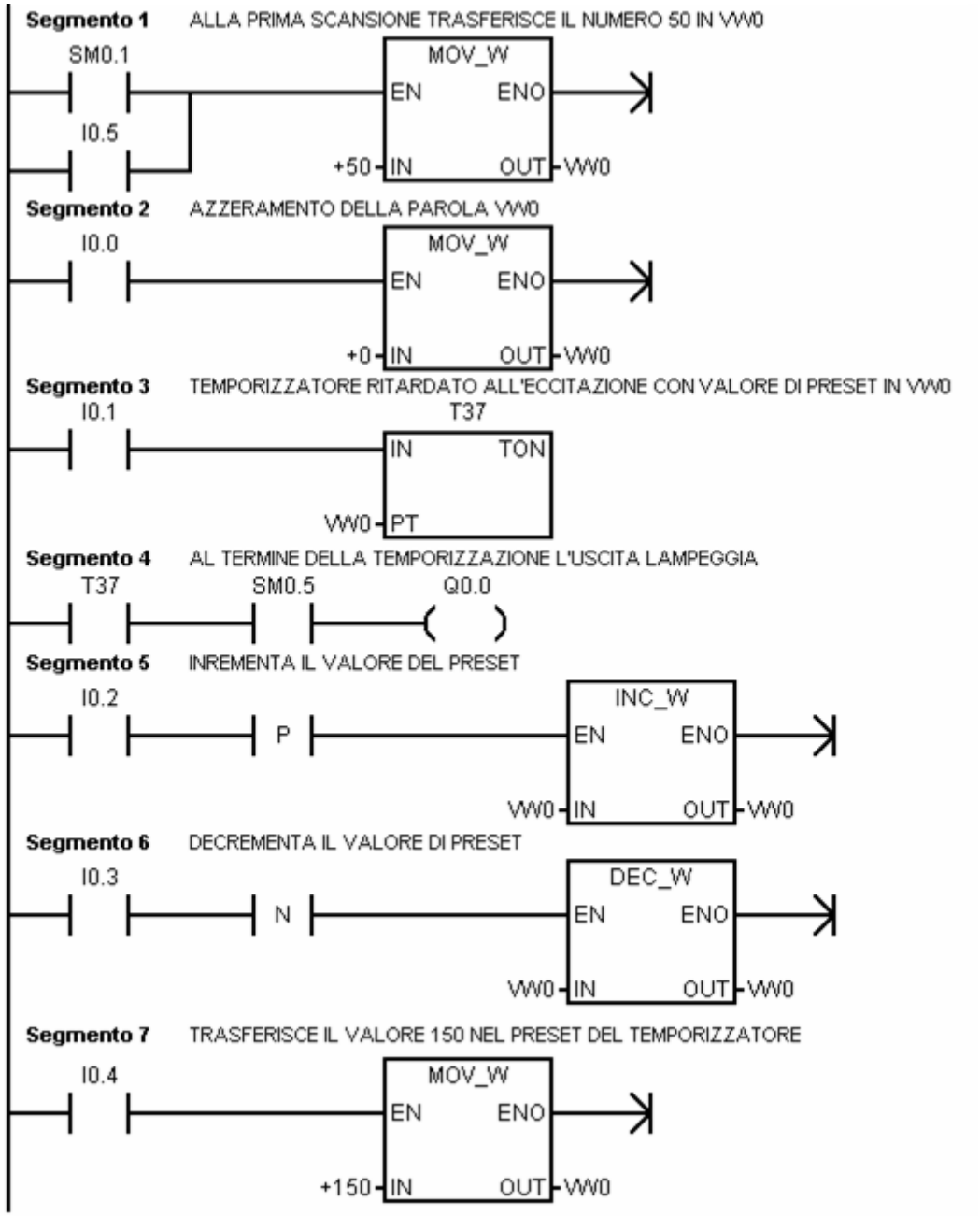
I0.1 avvia il temporizzatore

I0.2 incrementa il valore di Preset (VW0)

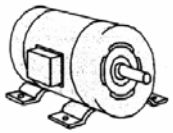
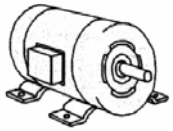
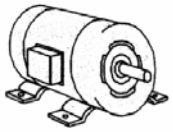
I0.3 decrementa il valore di Preset (VW0)

I0.4 trasferisce 150 in VW0

I0.5 trasferisce 50 in VW0


Segmento 1
ALLA PRIMA SCANSIONE TRASFERISCE IL NUMERO 50 IN VW0
SM0.1
I0.5
MOV_W
EN
ENO
+50
IN
OUT
VW0
Segmento 2
AZZERAMENTO DELLA PAROLA VW0
I0.0
MOV_W
EN
ENO
+0
IN
OUT
VW0
Segmento 3
TEMPORIZZATORE RITARDATO ALL'ECCITAZIONE CON VALORE DI PRESET IN VW0
I0.1
T37
IN
TON
VW0
PT
Segmento 4
AL TERMINE DELLA TEMPORIZZAZIONE L'USCITA LAMPEGGIA
T37
SM0.5
Q0.0
Segmento 5
INREMENTA IL VALORE DEL PRESET
I0.2
P
INC_W
EN
ENO
VW0
IN
OUT
VW0
Segmento 6
DECREMENTA IL VALORE DI PRESET
I0.3
N
DEC_W
EN
ENO
VW0
IN
OUT
VW0
Segmento 7
TRASFERISCE IL VALORE 150 NEL PRESET DEL TEMPORIZZATORE
I0.4
MOV_W
EN
ENO
+150
IN
OUT
VW0

Cicli motori ad azionamento manuale

	Q 0.3	Q 0.2	Q 0.1	Q 0.0		
						**CON PIÙ PULSANTI**
I 0.0	**0**	**0**	**0**	**0**	=	**0**
I 0.1	**1**	**1**	**1**	**1**	=	**15**
I 0.2	**1**	**0**	**1**	**0**	=	**10**
I 0.3	**0**	**1**	**0**	**1**	=	**5**

I 0.0 Inserimento e arresto impianto

I 0.1 Azionamento fase 1

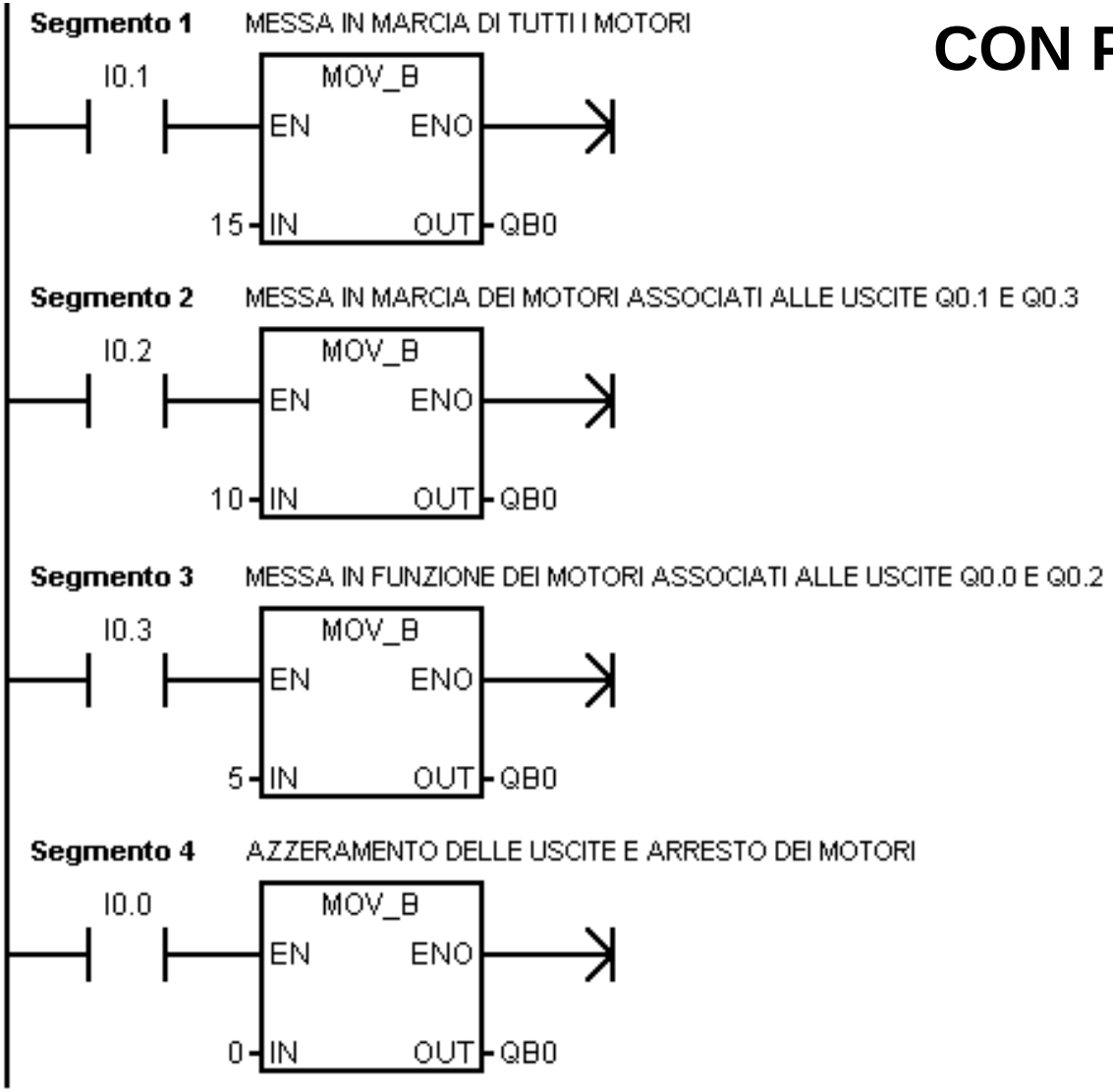
I 0.2 Azionamento fase 2

I 0.3 Azionamento fase 3

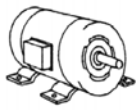
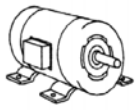
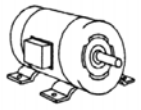
Premendo un pulsante verrà eseguito l'azionamento ad esso associato

Cicli motori ad azionamento manuale

CON PIÙ PULSANTI



Cicli motori ad azionamento manuale

	I 0.3	I0.2	I 0.1	I 0.0	
					
Azzeramento	**0**	**0**	**0**	**0**	
Primo impulso	**1**	**1**	**1**	**1**	CICLO
Secondo impulso	**1**	**0**	**1**	**0**	
Terzo impulso	**0**	**1**	**0**	**1**	

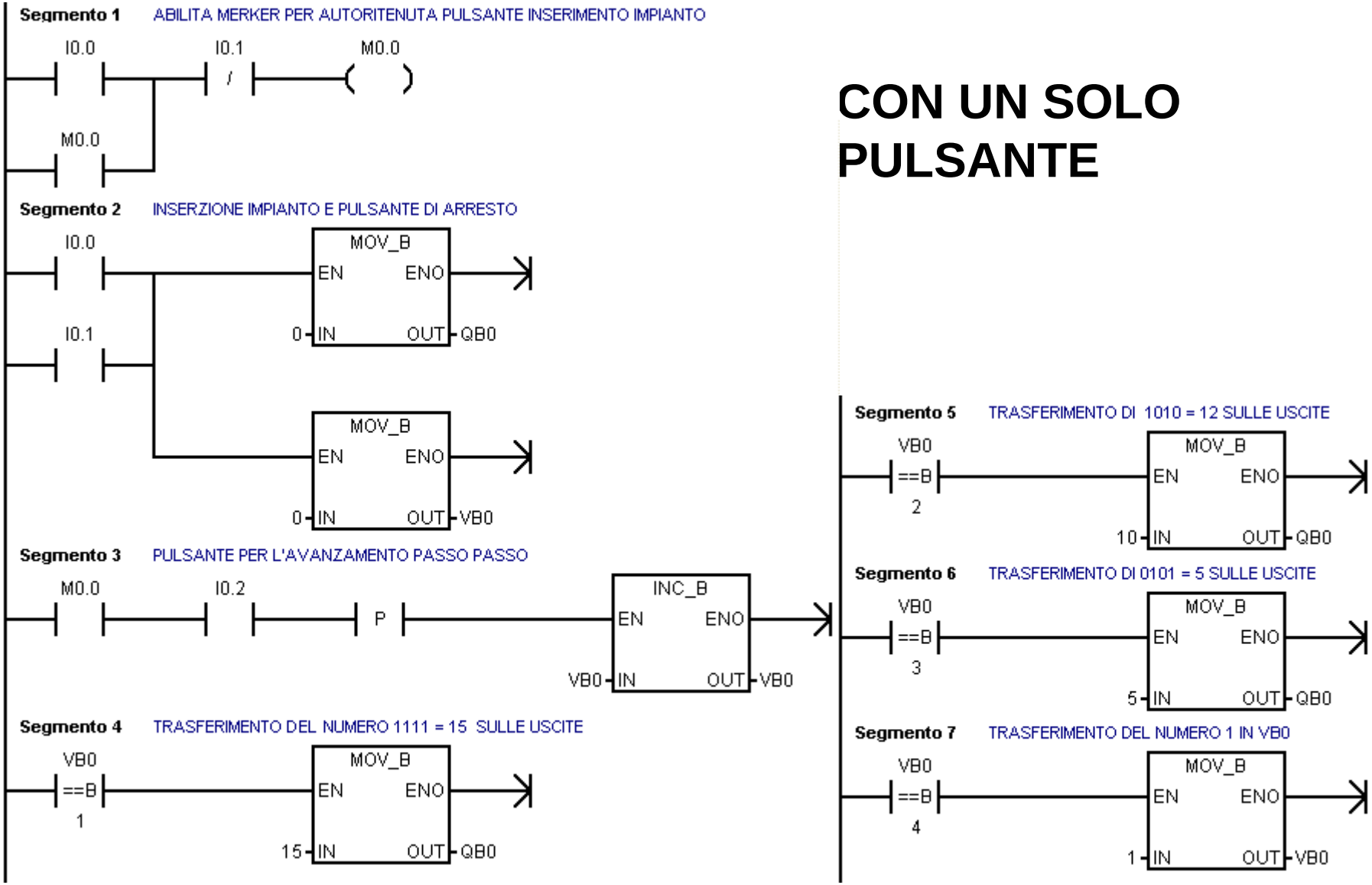
I 0.0 (ON) Inserimento impianto

I 0.1 Arresto (N.C.)

I 0.2 Avanzamento passo

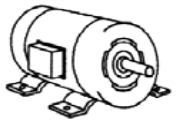
CON UN SOLO PULSANTE

Cicli motori ad azionamento manuale

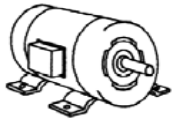


CON UN SOLO PULSANTE

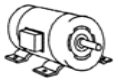
Ciclo Continuo



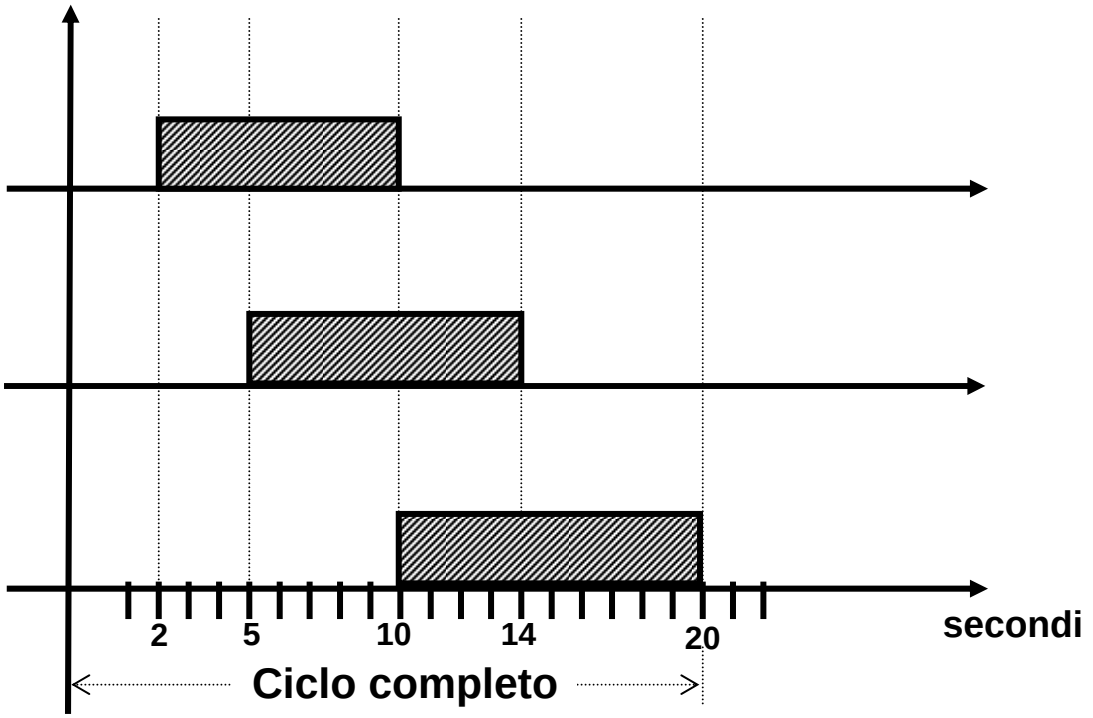
Q0.0 Motore 1



Q0.1 Motore 2



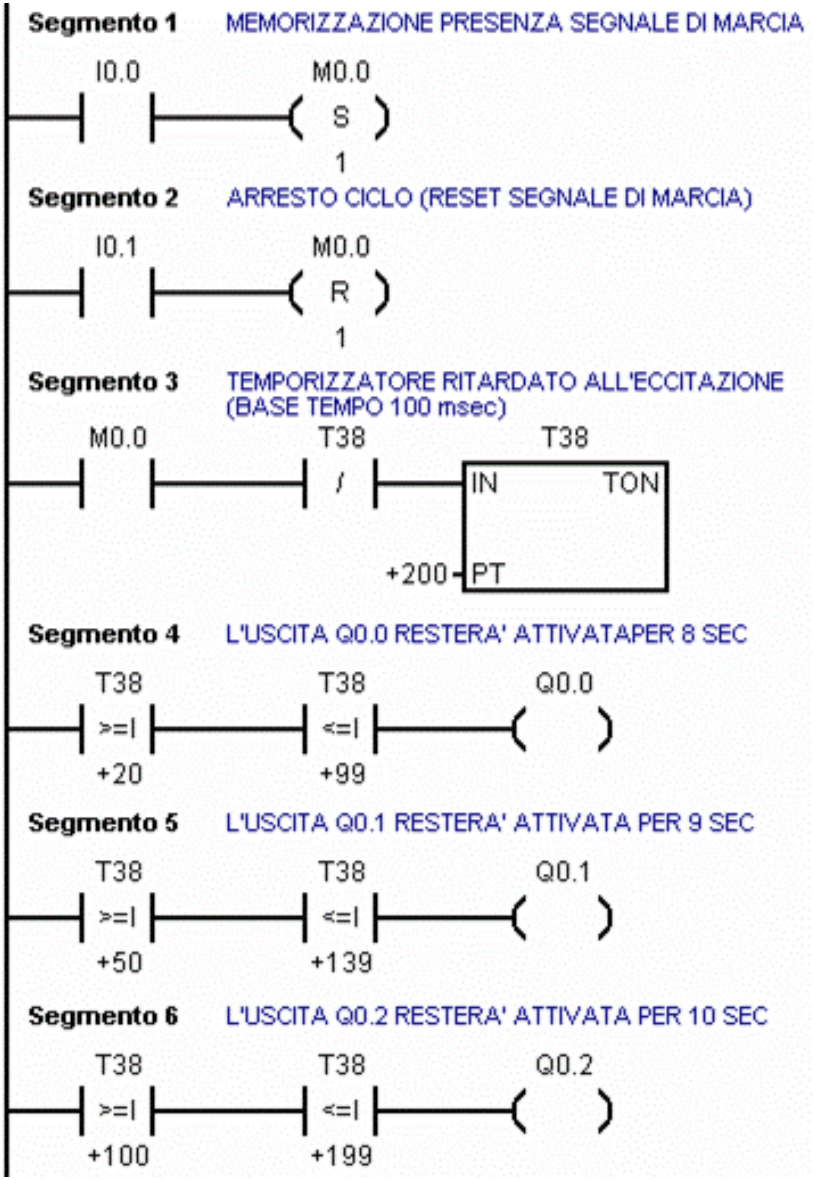
Q0.2 Ventola



I0.0 = Marcia (NA)
I0.1 = Arresto (NC)

Realizzare il programma che permette di avere la sequenza sopra indicata

Ciclo Continuo



ZERO MACCHINA

Fare in modo che all'accensione dell'apparecchiatura venga trasferita sulle uscita una specifica configurazione, come ad esempio:

0 0 1 0 1 0 1 0

Dove ad ogni bit corrisponde la posizione di un pistone:

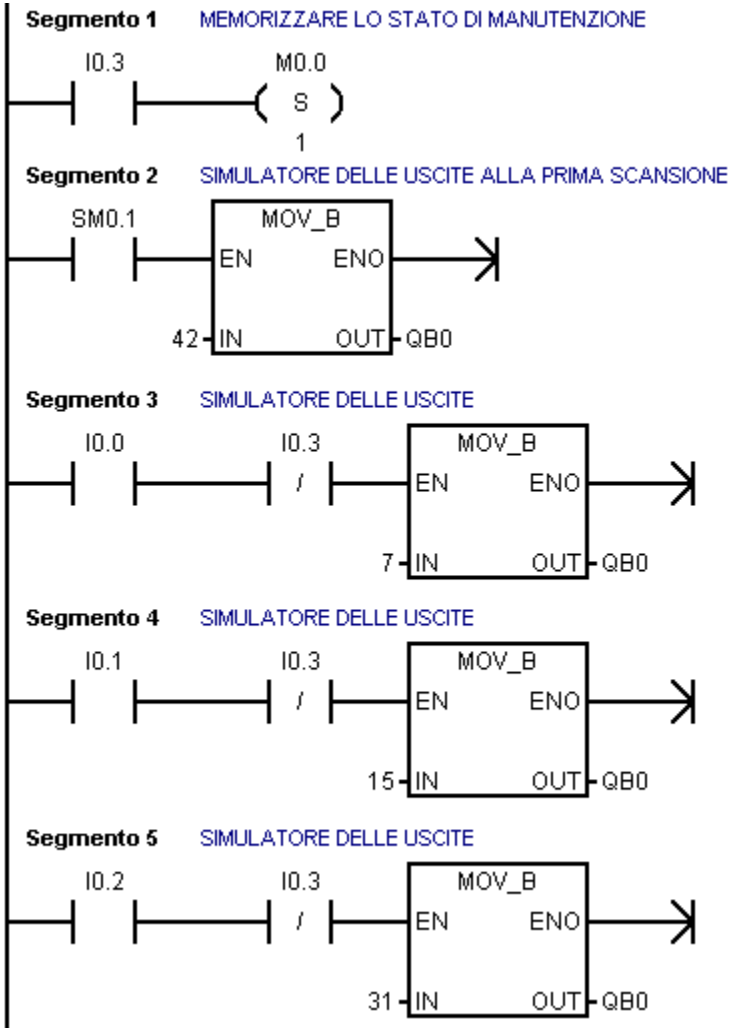
0 = posizione arretrata
1 = posizione avanzata

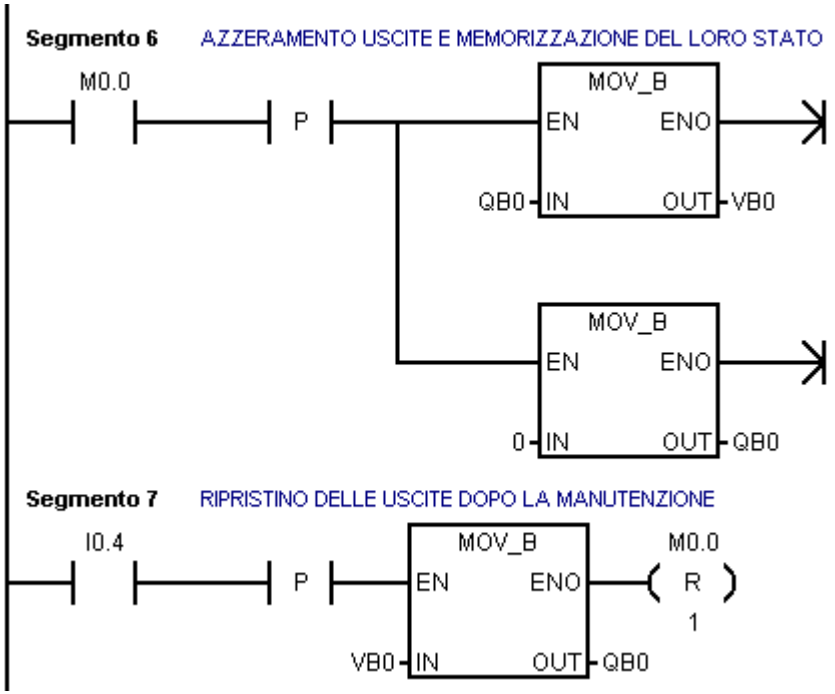
Per avere la possibilità di effettuare la manutenzione alla macchina è necessario portare tutti i pistoni in posizione arretrata, dopo avere memorizzata la posizione assunta dai pistoni prima della manutenzione, in modo che al termine della manutenzione si possa ripristinarla.

Simulare diverse uscite di lavoro, per esempio:
I0.0 ➔ 7 (0000-0111)
I0.1 ➔ 15 (0000-1111)
I0.2 ➔ 31 (0001-1111)

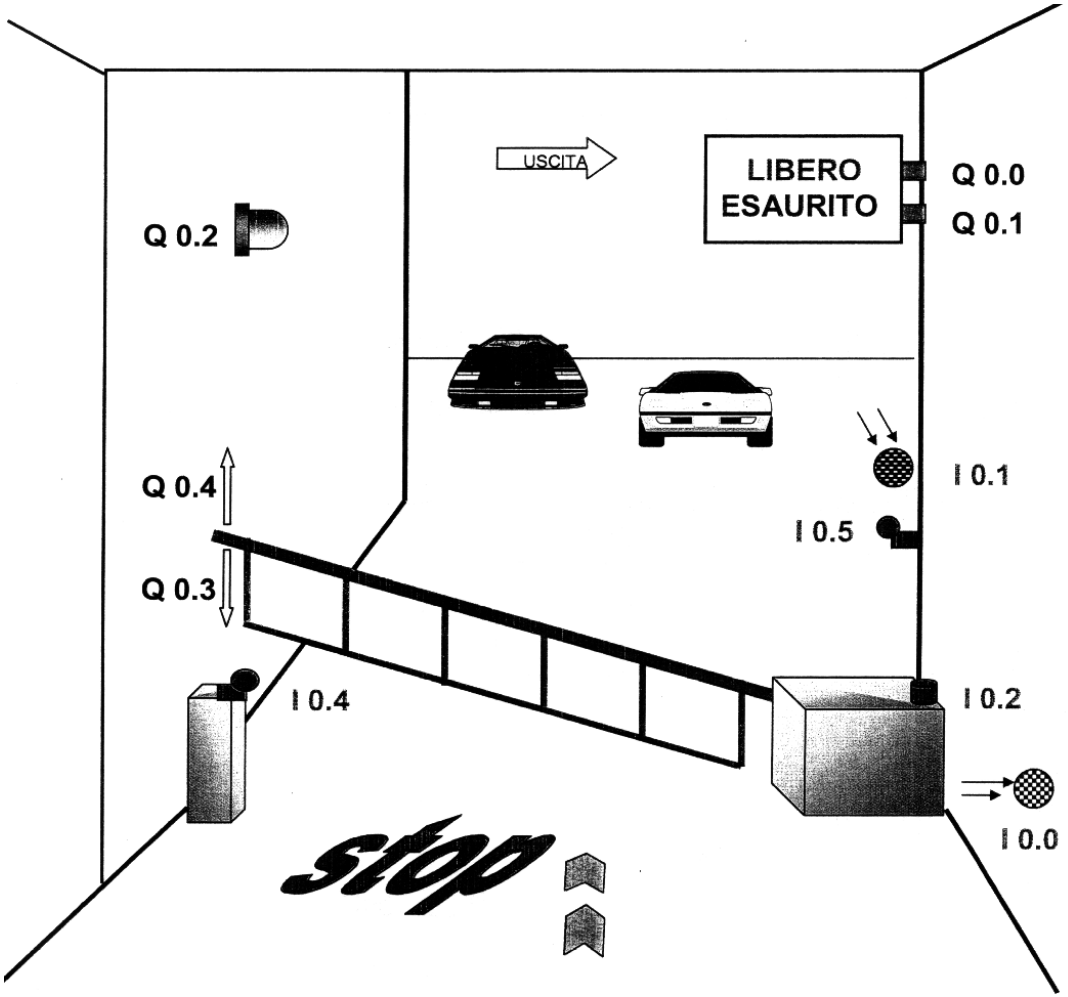
Mentre l'operazione di manutenzione sarà gestita con gli ingressi:
I0.3 ➔ avvio manutenzione
I0.4 ➔ ripristino operatività della macchina

Zero macchina





PARCHEGGIO



Ingressi

I0.0	fotocellula ingresso
I0.1	fotocellula uscita
I0.2	reset manuale
I0.4	fc sbarra in basso
I0.5	fc sbarra in alto

Uscite

Q0.0	indicazione libero
Q0.1	indicazione esaurito
Q0.2	lampeggio
Q0.3	ev sbarra in basso
Q0.4	ev sbarra in alto

PARCHEGGIO

Condizioni di partenza

- Il parcheggio può ospitare 10 macchine
- Sbarra di bloccaggio in posizione alta
- Lampeggiante di segnalazione spento solo quando la sbarra non è in movimento
- Il cartello luminoso visualizza “libero”

Funzionamento dell’impianto

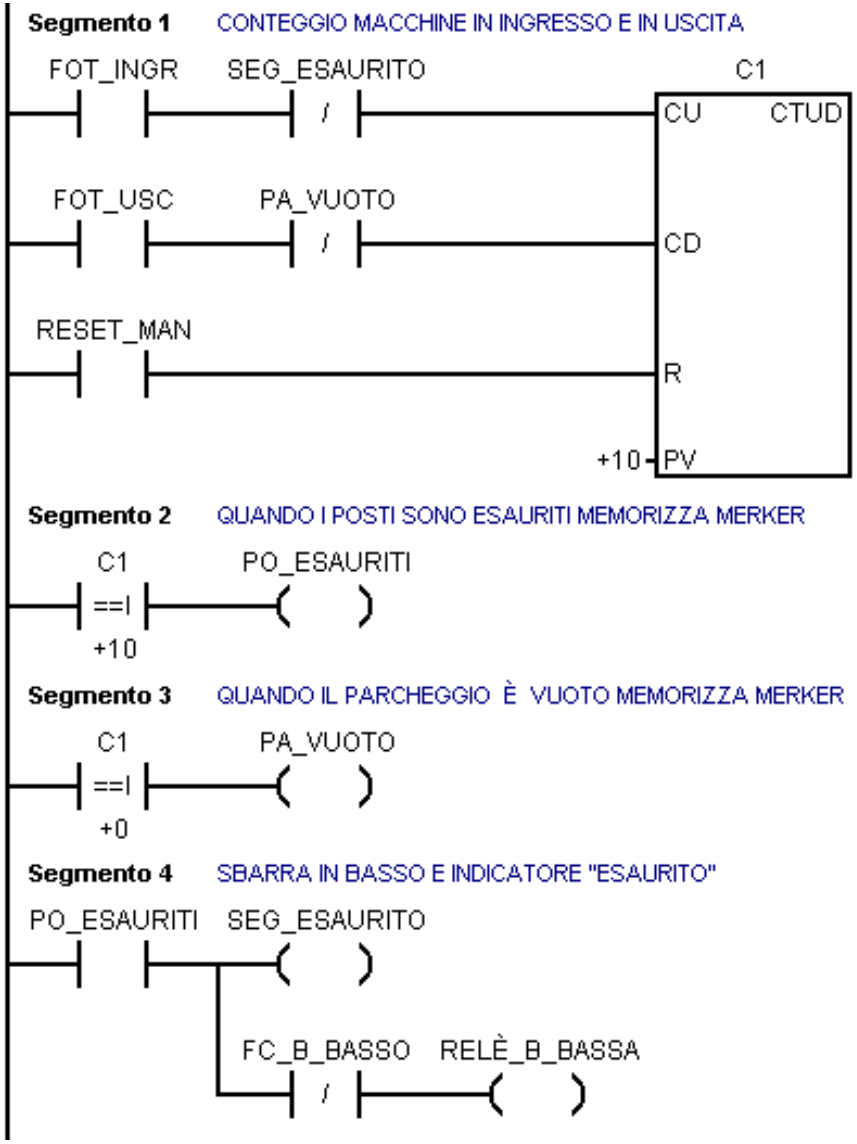
- Quando i posti riservati alle macchine sono tutti occupati, e con la presenza del segnale di reset manuale, si dovranno verificare le seguenti condizioni:
- Discesa della sbarra
- Il cartello luminoso dovrà segnalare “Esaurito”
- Appena si rende disponibile un posto macchina l’impianto si ripristina alle condizioni di partenza

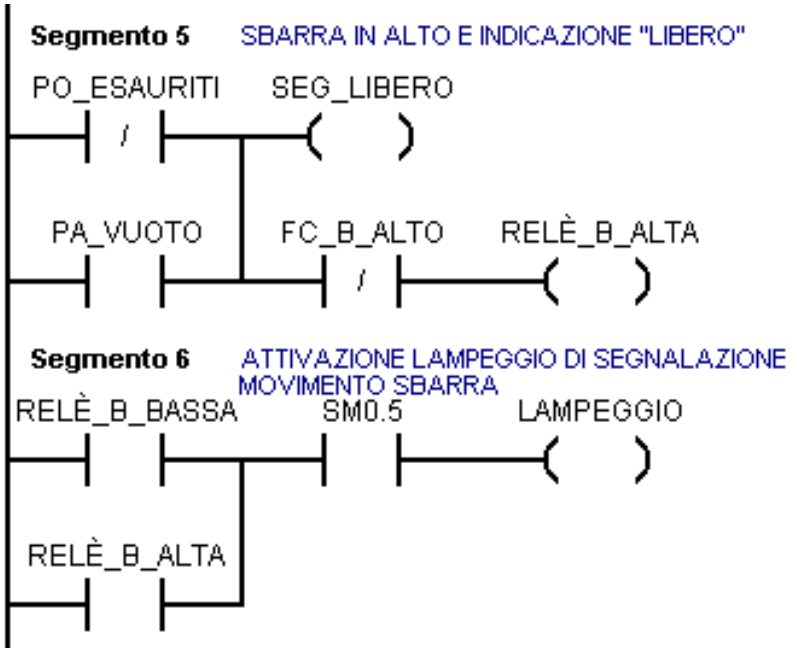
Parcheggio

Compilare la tabella dei simboli

Nome	Indirizzo	Commento
FOT_INGR	I0.0	FOTOCELLULA SEGNALAZIONE MACCHINA IN INGRESSO
FOT_USC	I0.1	FOTOCELLULA SEGNALAZIONE MACCHINA IN USCITA
RESET_MAN	I0.2	PULSANTE DI RESET MANUALE(AZZERAMENTO CONTATORE)
NON_UTILIZZATO	I0.3	
FC_B_BASSO	I0.4	FINE CORSA BARRA IN BASSO
FC_B_ALTO	I0.5	FINE CORSA BARRA IN ALTO
SEG_LIBERO	Q0.0	CARTELLO INDICATORE LUMINOSO DI "LIBERO"
SEG_ESAURITO	Q0.1	CARTELLO INDICATORE LUMINOSO DI "ESAURITO"
LAMPEGGIO	Q0.2	LAMPEGGIATORE DI SEGNALAZIONE MOVIMENTO BARRA
RELÈ_B_BASSA	Q0.3	RELÈ PER IL COMANDO BARRA IN BASSO
RELÈ_B_ALTA	Q0.4	RELÈ PER IL COMANDO BARRA IN ALTO
PO_ESAURITI	M0.0	MERKER POSTI ESAURITI
PA_VUOTO	M0.1	MERKER PARCHEGGIO VUOTO

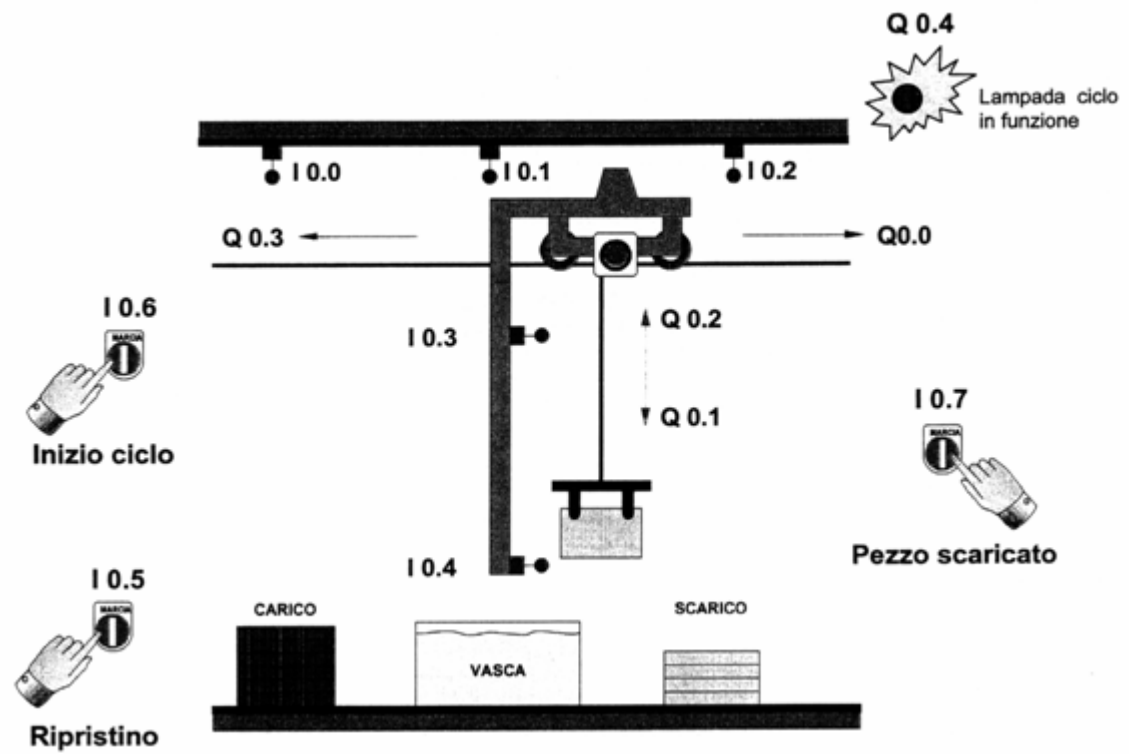
Parcheggio


Segmento 1
CONTEGGIO MACCHINE IN INGRESSO E IN USCITA
FOT_INGR
SEG_ESAURITO
/
C1
CU
CTUD
FOT_USC
PA_VUOTO
/
CD
RESET_MAN
R
+10
PV
Segmento 2
QUANDO I POSTI SONO ESAURITI MEMORIZZA MERKER
C1
==I
+10
PO_ESAURITI
Segmento 3
QUANDO IL PARCHEGGIO È VUOTO MEMORIZZA MERKER
C1
==I
+0
PA_VUOTO
Segmento 4
SBARRA IN BASSO E INDICATORE "ESAURITO"
PO_ESAURITI
SEG_ESAURITO
FC_B_BASSO
RELÈ_B_BASSA
/


Segmento 5
SBARRA IN ALTO E INDICAZIONE "LIBERO"
PO_ESAURITI
SEG_LIBERO
/
PA_VUOTO
FC_B_ALTO
/
RELÈ_B_ALTA
Segmento 6
ATTIVAZIONE LAMPEGGIO DI SEGNALAZIONE MOVIMENTO SBARRA
RELÈ_B_BASSA
SM0.5
LAMPEGGIO
RELÈ_B_ALTA

LAVAGGIO PEZZI

Un carro si sposta lungo una rotaia, posizionandosi sopra una vasca, in modo da lavare dei pezzi tenendoli immersi per 30 secondi.
A fine lavaggio si effettuerà l'operazione di scarico.



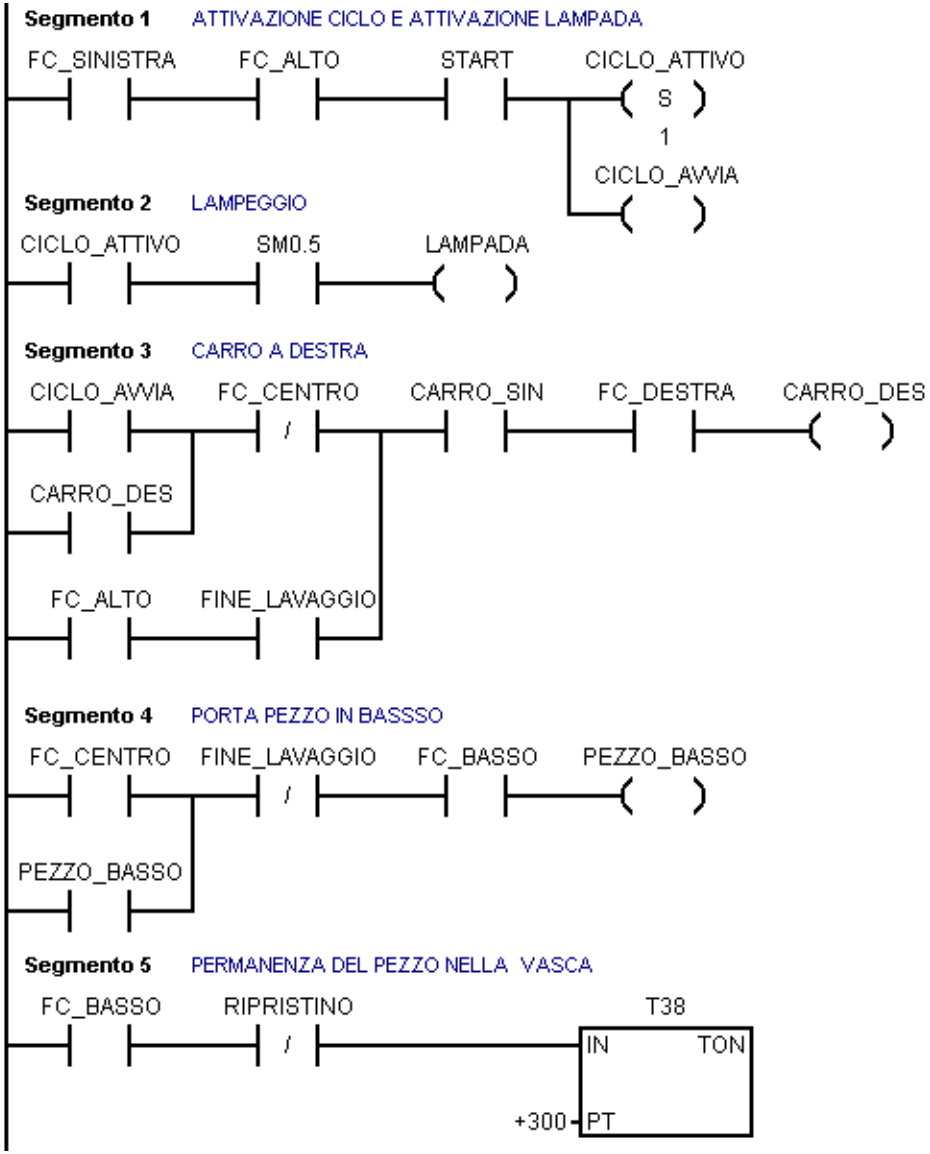
Il carico e lo scarico sono effettuati manualmente.
Il comando di partenza e quello di scarico sono dati dall'operatore.
Il carro parte se è a sinistra e il porta pezzi in posizione alta.

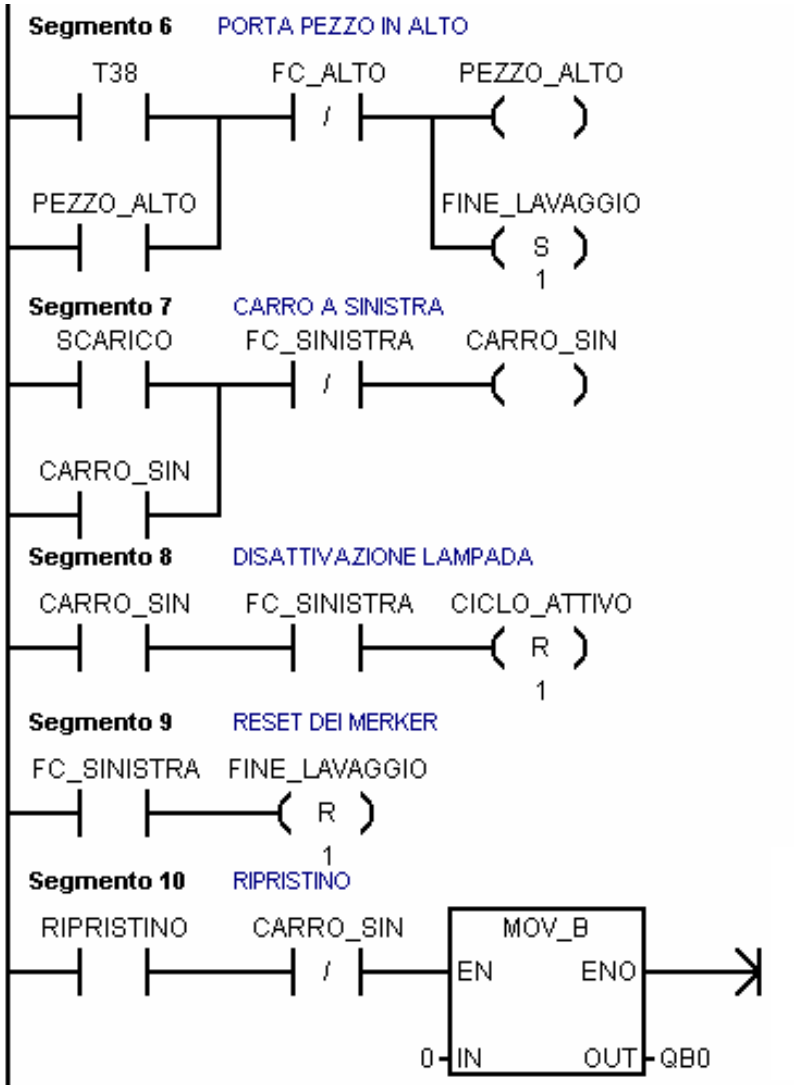
Automatismo per lavaggio pezzi

Compilare la tabella dei simboli

Nome	Indirizzo	Commento
FC_SINISTRA	I0.0	FINE CORSA PRESENZA CARRO A SINISTRA
FC_CENTRO	I0.1	FINE CORSA PRESENZA CARRO IN CENTRO
FC_DESTRA	I0.2	FINE CORSA PRESENZA CARRO A DESTRA
FC_ALTO	I0.3	FINE CORSA PRESENZA PORTA PEZZO IN ALTO
FC_BASSO	I0.4	FINE CORSA PRESENZA PORTA PEZZO IN BASSO
RIPRISTINO	I0.5	PULSANTE DI RIPRISTINO
START	I0.6	PULSANTE DI MARCIA
SCARICO	I0.7	PULSANTE PER RITORNO CARRO TUTTO A SINISTRA
CARRO_DES	Q0.0	ATTUATORE PER MOVIMENTO CARRO A DESTRA
PEZZO_BASSO	Q0.1	ATTUATORE PER MOVIMENTO PEZZO IN BASSO
PEZZO_ALTO	Q0.2	ATTUATORE PER MOVIMENTO PEZZO IN ALTO
CARRO_SIN	Q0.3	ATTUATORE PER MOVIMENTO CARRO A SINISTRA
LAMPADA	Q0.4	LAMPEGGIO
CICLO_ATTIVO	M0.0	MEMORIZZAZIONE CICLO IN FUNZIONE
CICLO_AVVIA	M0.2	MEMORIZZAZIONE AVVIO CICLO
FINE_LAVAGGIO	M0.1	MERKER FINE LAVAGGIO

Automatismo per lavaggio pezzi





Realizzare un comando di sicurezza bimanuale

Semilavorati in acciaio devono essere formati usando una pressa comandata da un cilindro a doppio effetto

Al fine di soddisfare le norme di sicurezza si deve prevedere un sistema con due pulsanti, in modo che entrambe le mani siano occupate

Per ottenere l'azionamento a due pulsanti, questi vanno premuti in un intervallo di tempo di 0,5 sec massimo

Lo stelo del cilindro deve effettuare la corsa di ritorno immediatamente se uno o entrambi i pulsanti vengono rilasciati

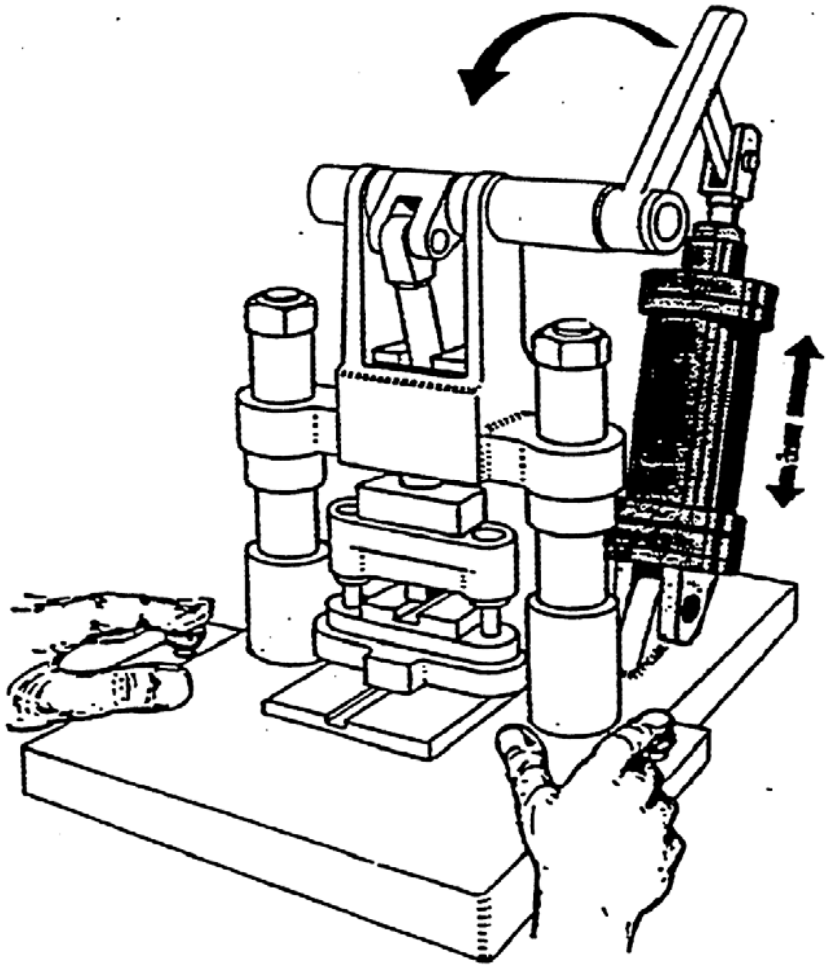
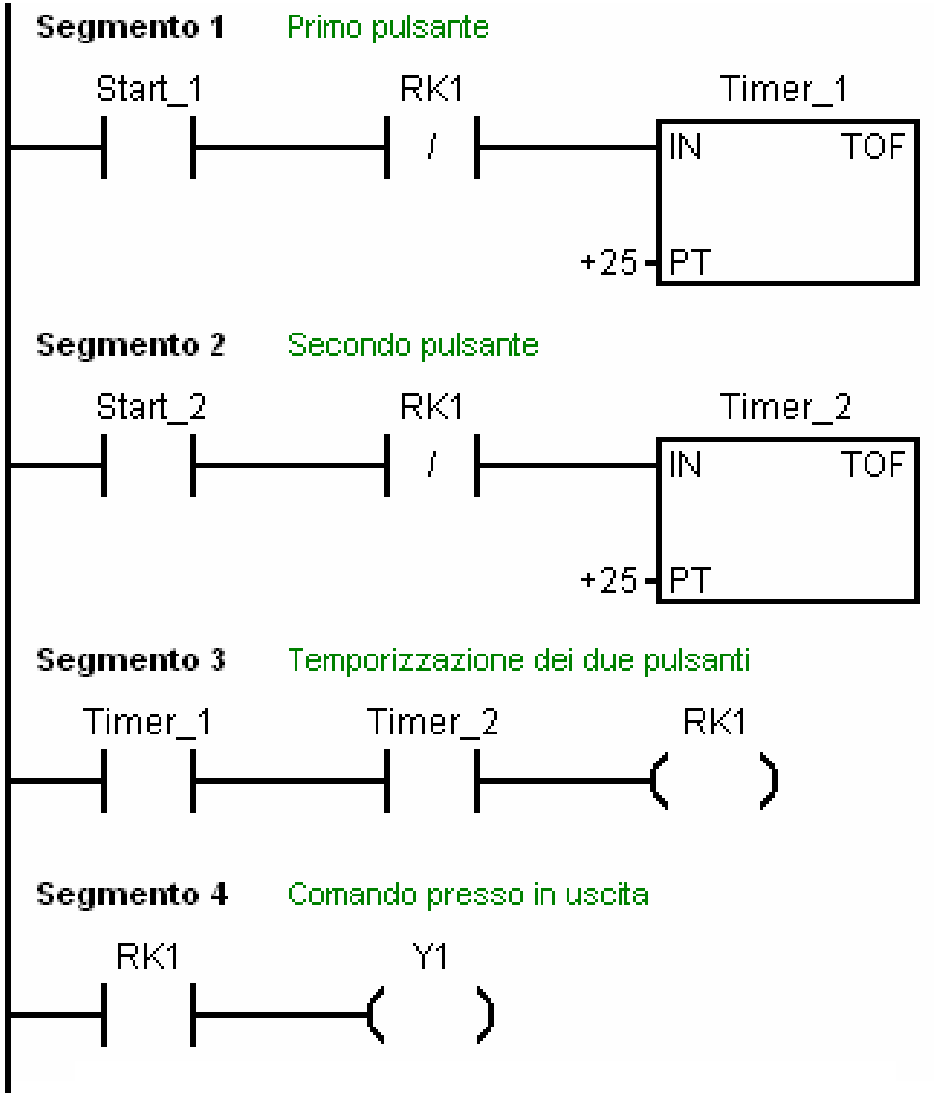


Tabella dei simboli

Operando	Simbolo	Commento
I0.0	START_1	PULSANTE MARCIA 1 N.A.
I0.1	START_2	PULSANTE MARCIA 2 N.A.
Q0.0	Y1	PRESSA (SIMUL. CILINDRO)
M0.0	RK1	MEMORIA AUSILIARIA
T97	TIMER_1	TEMP.SICUREZZA RIT. DISECC
T98	TIMER_2	TEMP.SICUREZZA RIT. DISECC.

Diagramma KOP



Obiettivo

- Conoscere le caratteristiche tecniche e funzionali dei sensori magnetici (Reed)

Richieste

- Compilazione delle liste d' occupazione degli ingressi e delle uscite
- Stesura dello schema elettrico
- Stesura del Flow-chart o stesura del Grafcet
- Stesura, caricamento e prova del programma

Dispositivo di ribaltamento

Un cilindro deve ribaltare un recipiente contenente merci sfuse

All' azionamento di un pulsante il cilindro deve effettuare il ribaltamento e poi rientrare automaticamente anche se rimane azionato il pulsante di START.

Data la particolare applicazione del cilindro, non è possibile utilizzare finecorsa meccanici, ma vengono impiegati sensori elettronici ad azionamento magnetico.

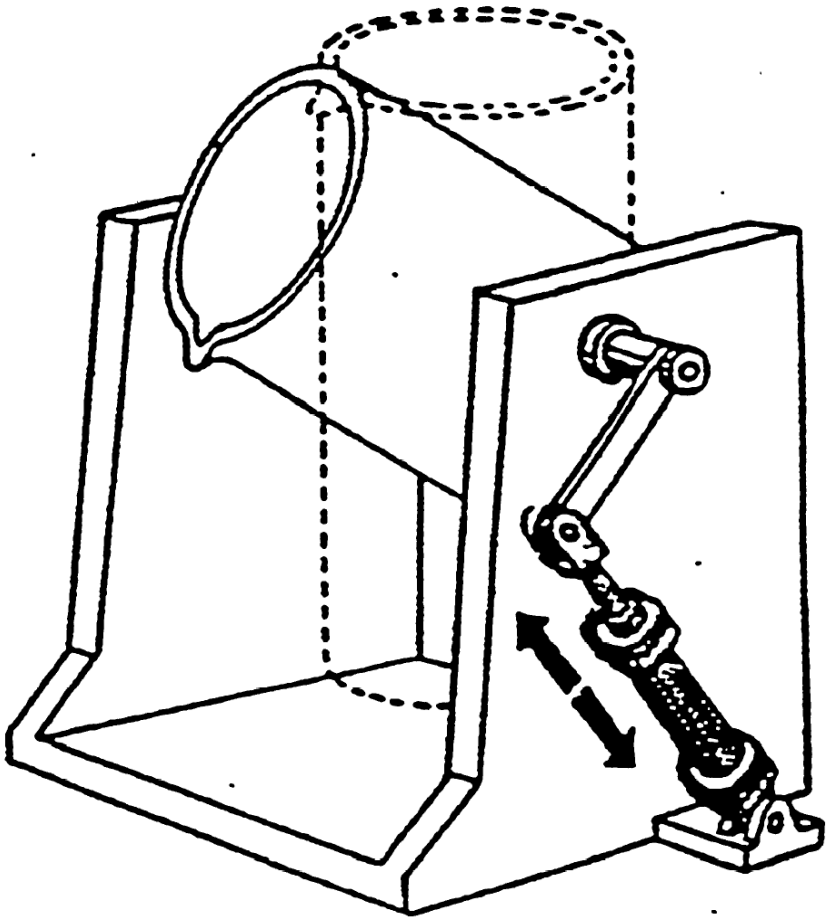
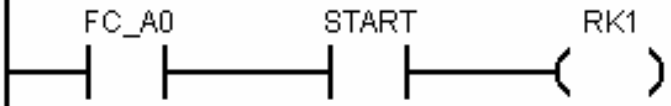


Tabella simboli

Operando	Simbolo	Commento
I0.0	START	PULSANTE MARCIA N.A.
I0.1	FC_A0	FINECORSA CILINDRO DENTRO
I0.2	FC_A1	FINECORSA CILINDRO FUORI
Q0.0	Y1	ABILIT.CILINDRO FUORI
Q0.1	Y2	RESET CILINDRO FUORI
M0.0	RK1	MEMORIA AUSILIARIA 1
M0.1	RK2	MEMORIA AUSILIARIA 2

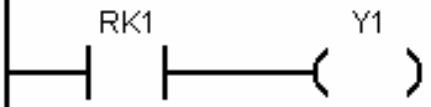
Diagramma KOP

Segmento 1 AZIONAMENTO PULSANTE



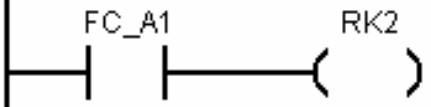
FC_A0	I0.1	FINECORSA CILIND.DENTRO
RK1	M0.0	MEMORIA AUSILIARIA 1
START	I0.0	PULSANTE MARCIA N.O.

Segmento 2 AZIONAMENTO CILINDRO IN USCITA



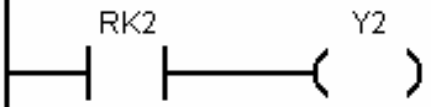
RK1	M0.0	MEMORIA AUSILIARIA 1
Y1	Q0.0	ABILIT.CILINDRO FUORI

Segmento 3 SEGNALAZIONE CILINDRO FUORI



FC_A1	I0.2	FINECORSA CILINDRO FUORI
RK2	M0.1	MEMORIA AUSILIARIA 2

Segmento 4 AZIONAMENTO CILINDRO IN ENTRATA



RK2	M0.1	MEMORIA AUSILIARIA 2
Y2	Q0.1	RESET CILINDRO FUORI