

INTRODUZIONE ALLA PROGRAMMAZIONE DEI PLC IN LINGUAGGIO LADDER

DESCRIZIONE DELLE PRINCIPALI
CARATTERISTICHE DEL SOFTWARE S.7

Introduzione

Il PLC (Programmable Logic Controller) ha ormai quasi sostituito la tecnologia a relè, specialmente in campo industriale.

L' utilizzo di questa apparecchiatura comporta diversi vantaggi, quali:

- riduzione dei costi
- ingombri ridotti
- minore probabilità di guasti, spesso dovuti all' usura dei relè
- possibilità di adattare la macchina alle diverse esigenze produttive, semplicemente modificando il programma.

Un PLC è costituito da una CPU che comunica con l' ambiente esterno attraverso terminali di ingresso ed uscita.

I terminali di ingresso rilevano lo stato dei sensori (es. finecorsa, pulsanti, fotocellule) installati sulla macchina; mediante i terminali di uscita vengono invece comandati teleruttori, elettrovalvole, segnalazioni luminose ecc.

I moderni PLC possono essere programmati mediante PC, utilizzando il software fornito dalle case costruttrici, che mette a disposizione diversi linguaggi. Tra questi il più semplice è il linguaggio ladder.

Un diagramma ladder si basa su una logica piuttosto simile a quella di uno schema funzionale elettromeccanico: i suoi elementi base non sono altro che relè virtuali.

Nelle pagine che seguono mi concentrerò sul software relativo ai PLC S7200, descrivendo le principali procedure per la compilazione, il caricamento e la verifica dei programmi.

Operazioni preliminari

Prima di iniziare ad utilizzare il PLC è opportuno verificare alcune impostazioni hardware e software, facendo riferimento al manuale fornito dalla casa costruttrice.

Si raccomanda di controllare la tensione di alimentazione (non è uguale in tutti i modelli).

Per caricare il programma nel PLC occorre accertarsi che quest' ultimo sia in modalità **stop** (led giallo). Eventualmente cliccare **cpu – stop** (barra strumenti in alto).

A questo punto cliccare **file – carica nella cpu** e confermare; dopo alcuni secondi, se non vengono riscontrati errori, appare un messaggio di conferma.

Per avviare l' esecuzione del programma cliccare **cpu – run** (si accenderà il led verde).

N.B. la modalità run-stop può essere impostata anche tramite un selettore fisico, situato sotto lo sportellino del PLC.

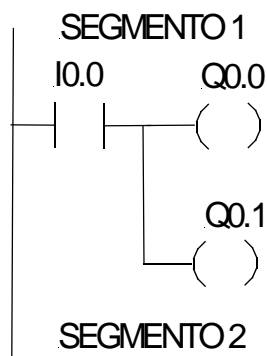
Durante l' esecuzione del programma è possibile visualizzare lo stato logico e di conteggio dei relè, temporizzatori e contatori. Per attivare questa funzione occorre cliccare **test – stato del programma** (barra strumenti in alto).

Programmazione di base

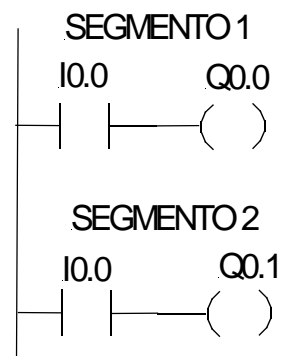
Per accedere alla programmazione in linguaggio ladder occorre cliccare l' icona **blocco di codice** (in alto a sinistra). Si aprirà una schermata suddivisa in segmenti. Gli elementi necessari per la compilazione del programma sono raggruppati in cartelle. Per inserire un elemento nel diagramma dobbiamo aprire la relativa cartella, selezionarlo e trascinarlo (tenendo cliccato).

Per una corretta compilazione del programma occorre siglare gli elementi seguendo precise tabelle (non è quindi ammesso assegnare sigle a piacere, come avviene negli schemi elettromeccanici).

Evitare inoltre di inserire più bobine nello stesso segmento (esempio sotto)



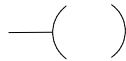
CONFIGURAZIONE ERRATA



CONFIGURAZIONE CORRETTA

Bobine e contatti

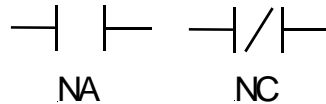
Le bobine dei relè ed i contatti(*) si trovano nella cartella **logica a bit**.
La bobina di un relè viene rappresentata con questo simbolo



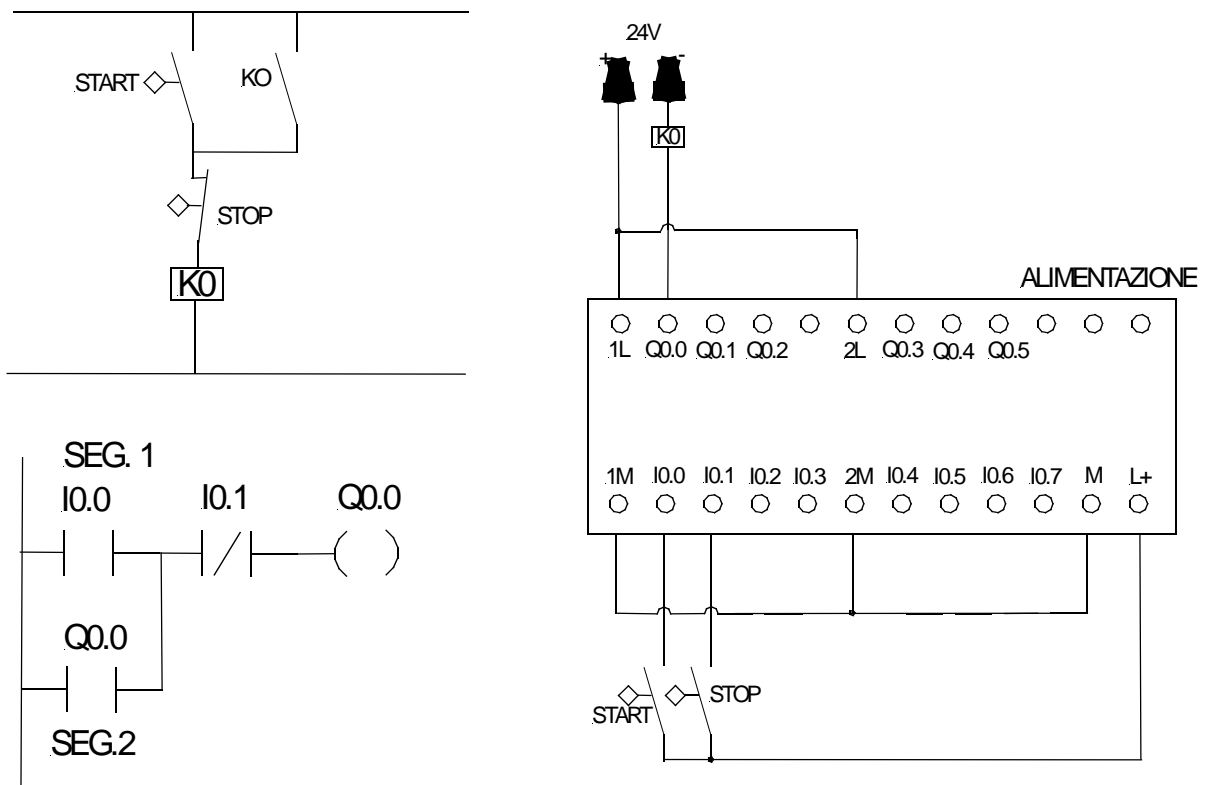
I relè siglati Q0.0...Q0.7, Q1.0...Q1.7 controllano lo stato delle relative uscite (possiamo paragonarli ai contattori).

I relè siglati M0.0...M0.7, M1.0...M1.7... sono ausiliari (merker)

Il contatto viene rappresentato così:



I contatti siglati I0.0...I0.7, I1.0...I1.7 sono comandati dai sensori sui relativi ingressi.
Per comprendere meglio consideriamo un semplice schema elettromeccanico, che trasformeremo in diagramma ladder.



Non appena il programma va in esecuzione (modalità run) i contatti sono a riposo, come disegnati nel diagramma. Premendo il pulsante START abilitiamo l'ingresso I0.0; di conseguenza il relativo contatto sul diagramma si chiude eccitando Q0.0. Quest'ultimo abilita la relativa uscita, quindi il teleruttore di marcia. Premendo il pulsante STOP abilitiamo l'ingresso I0.1: il relativo contatto sul diagramma (che è normalmente chiuso) si apre diseccitando Q0.0.

(*) I contatti di confronto, che vedremo più avanti, si trovano in una cartella a parte

Temporizzatori

I temporizzatori possono essere ritardati alla diseccitazione (TOF) o all' eccitazione (TON, TONR). Vengono siglati con la lettera T seguita da un numero, da 0 a 255. Il TON si eccita solo se abilitato per il periodo di tempo stabilito senza interruzioni; il TONR ha una funzione di memoria che gli permette di eccitarsi anche in caso di interruzione del conteggio.

I temporizzatori si differenziano anche in base alla **risoluzione**, ovvero il rapporto tra il ritardo ed il valore impostato (PT).

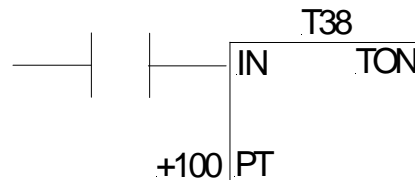
$$\text{valore impostato} = \frac{\text{ritardo}}{\text{risoluzione}}$$

	SIGLA TEMPORIZZATORE	RISOLUZIONE (secondi/unità)
TON - TOF	T32, T96	0,001
	da T33 a T36; da T97 a T100	0,01
	da T37 a T63; da T101 a T255	0,1
TONR	T0, T64	0,001
	da T1 a T4; da T65 a T68	0,01
	da T5 a T31; da T69 a T95	0,1

Supponiamo di dover impostare un ritardo di 10 secondi su un temporizzatore di tipo TON, che nominiamo T38.

Dalla tabella risulta che la risoluzione di T38 è 0,1. Il valore impostato si ottiene dividendo 10 per 0,1.

Per ottenere un ritardo di 10 secondi occorre quindi impostare un valore pari a 100 unità.

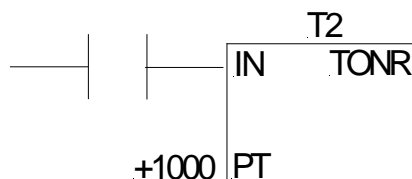


BOBINA TEMPORIZZATORE

Nel caso appena descritto il temporizzatore si eccita solo se la bobina viene alimentata per 10 secondi consecutivi.

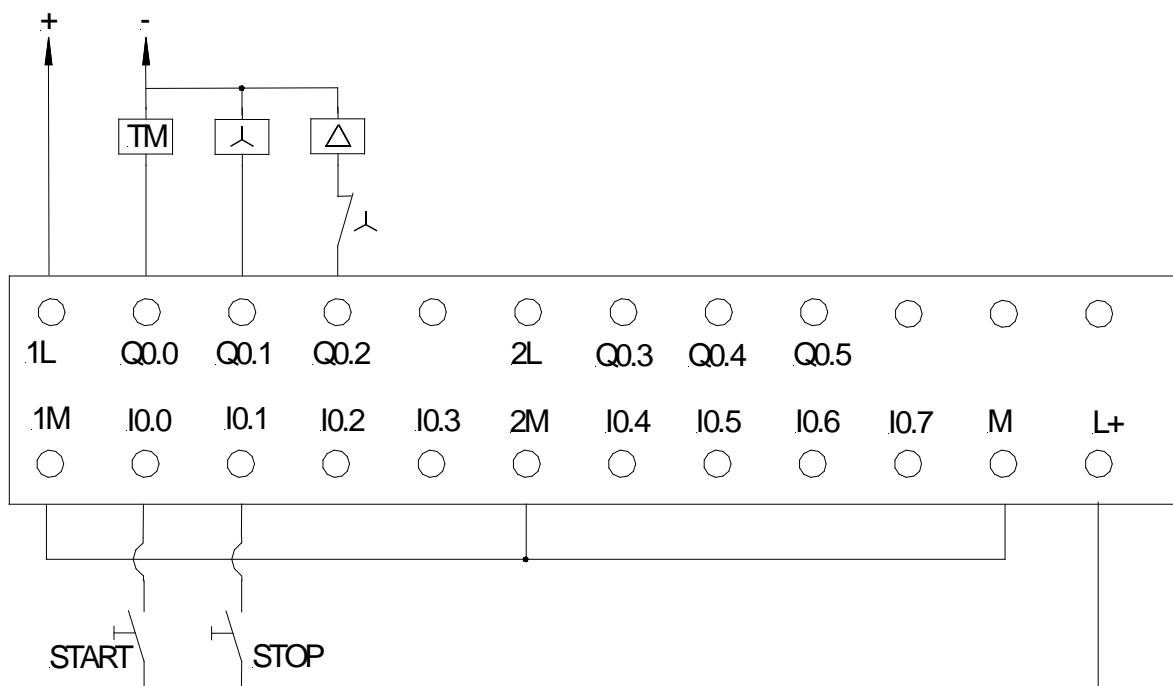
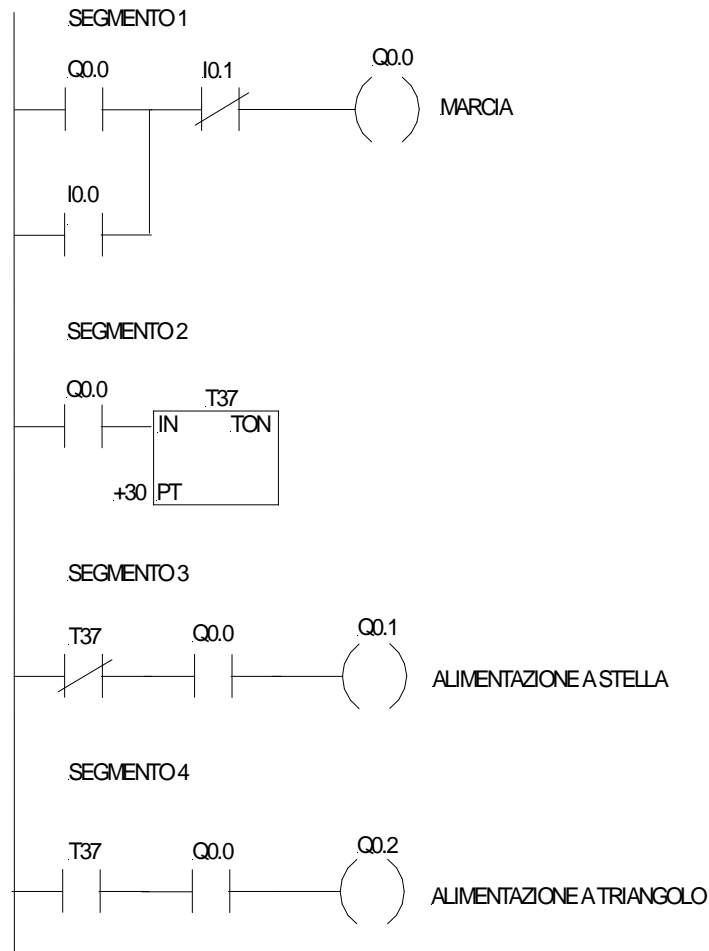
Consideriamo ora un temporizzatore di tipo TONR, a cui assegnamo la sigla T2. Anche in questo caso il ritardo è 10 secondi. Dato che T2 ha una risoluzione pari a 0,01, il valore impostato sarà 1000.

Se ad esempio alimentiamo la bobina per soli 6 secondi il conteggio non si azzerà; di conseguenza, nel momento in cui riabilitiamo la bobina, quest' ultima si ecciterà dopo 4 secondi.



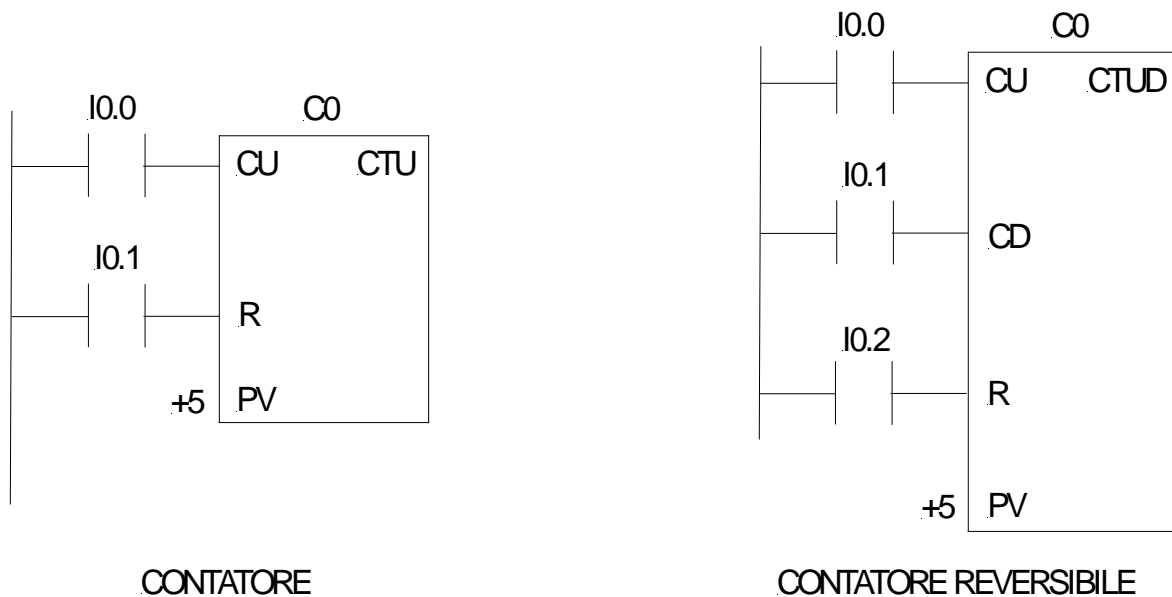
Avviamento stella-triangolo

Premendo il pulsante di avvio si eccita Q0.0 e subito dopo (frazione di secondo) Q0.1. Trascorsi 3 secondi T37 si eccita, disabilitando Q0.1 ed abilitando Q0.2.



Contatori

I contatori vengono siglati con la lettera C seguita da un numero da 0 a 255. Hanno uno o due terminali per il conteggio (nel secondo caso si tratta di contatori reversibili) ed un terminale di azzeramento.



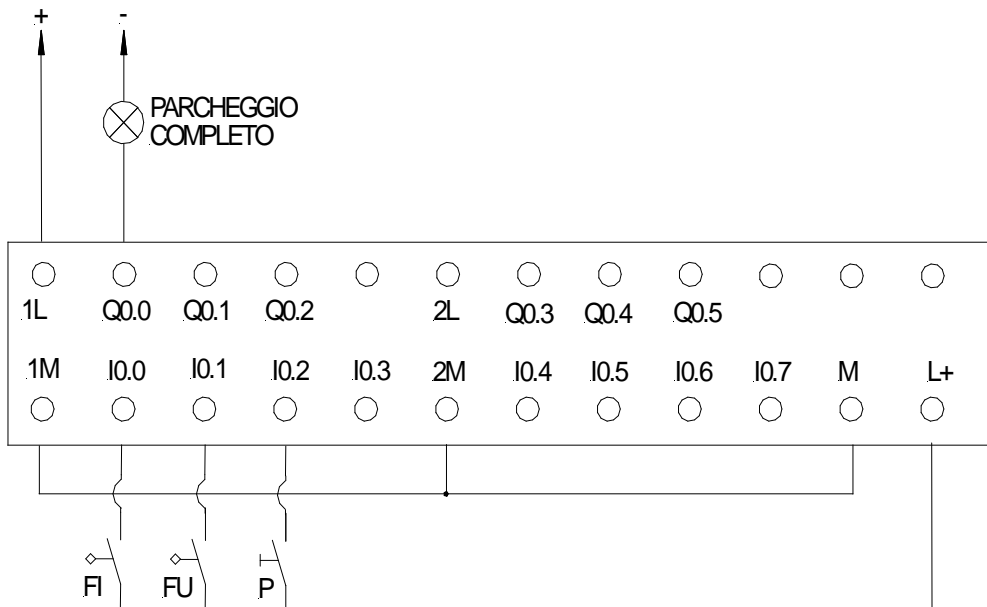
La figura sulla sinistra rappresenta un contatore di tipo CTU. Inizialmente il suo valore è 0 quindi è in condizione di riposo. Ad ogni chiusura e riapertura del contatto I0.0, ovvero ad ogni **impulso** sul terminale CU, suddetto valore aumenta di una unità. Nel momento in cui raggiunge il valore impostato, in questo caso 5, il contatore si eccita.

Abilitando l'ingresso I0.1 e quindi il terminale R il valore si azzerà, diseccitando il contatore.

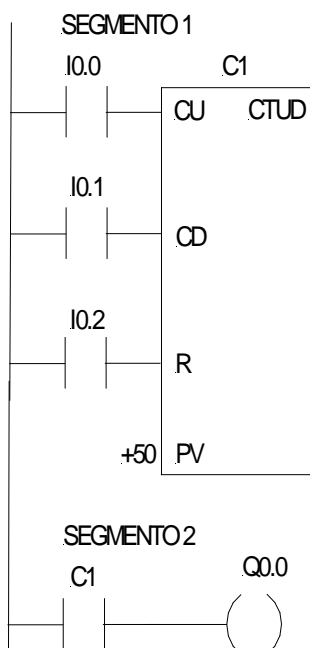
La figura accanto rappresenta un contatore reversibile (CTUD). In questo caso è presente anche un terminale di conteggio negativo (CD): ad ogni impulso su quest'ultimo il valore diminuisce di una unità.

Controllo degli accessi ad un parcheggio

Agli ingressi del PLC colleghiamo le due fotocellule ed un comando di azzeramento; in uscita colleghiamo una lampada che si accenderà a parcheggio completo. Quando un' autovettura accede al parcheggio la relativa fotocellula trasmette un impulso all' ingresso I0.0 e quindi al terminale CU del contatore: il valore aumenta di una unità. Quando un' autovettura abbandona il parcheggio la relativa fotocellula trasmette un impulso al terminale CD del contatore: il valore diminuisce di una unità. Nel momento in cui suddetto valore raggiunge il valore impostato (supponiamo una capienza di 50 autovetture) il contatore si eccita ed il relativo contatto n.a. abilita la bobina Q0.0. Di conseguenza la lampada si accende.



FI = FOTOCELLULA INGRESSO PARCHEGGIO
 FU = FOTOCELLULA USCITA PARCHEGGIO
 P = COMANDO AZZERAMENTO



Istruzioni word

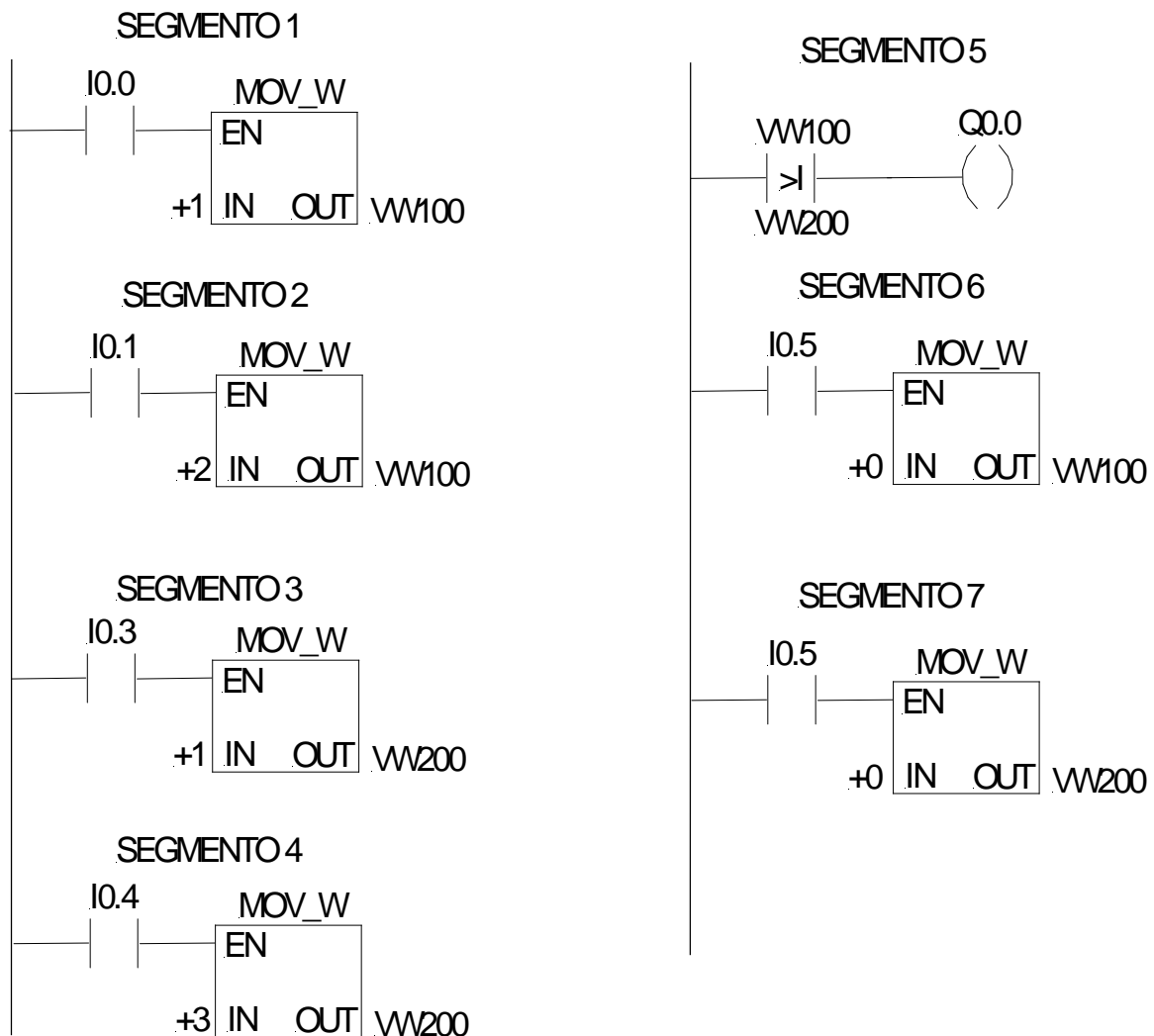
La **word** è un particolare contatore il cui valore viene aggiornato da una serie di istruzioni durante l'esecuzione del programma.

A differenza degli elementi visti fino ad ora, una word non ha stato logico (eccitato o diseccitato). Utilizzando uno speciale contatto è possibile impostare un **confronto** tra due word, oppure tra una word ed un valore stabilito. In base a tale confronto il contattorisulterà aperto o chiuso.

Istruzioni di trasferimento(MOV_W)

L'istruzione di trasferimento, quando abilitata (è sufficiente un impulso), assegna alla relativa word un determinato valore. Per impostarla occorre inserire sulla destra (OUT) la sigla della word alla quale è indirizzata e sulla sinistra (IN) il valore.

Quest'ultimo può essere anche copiato da un'altra word (di cui digiteremo la sigla)



Nell' esempio sopra riportato vengono configurate due word, che nominiamo VW100 e VW200. Nei segmenti 1,2,3,4,6,7 impostiamo le istruzioni di trasferimento.

Nel segmento 5 impostiamo il confronto tra le due word: in questo caso il contatto risulta chiuso quando il valore di VW100 è maggiore rispetto a quello di VW200.

Non appena il programma va in esecuzione entrambe le word hanno valore 0, di conseguenza il contatto di confronto risulta aperto.

Diamo un impulso all' ingresso I0.1: VW100 assume valore 2, come impostato nel segmento 2; pertanto il contatto di confronto si chiude, abilitando la bobina, ovvero l'uscita Q0.0.

Diamo un impulso all' ingresso I0.3: la VW200 assume valore 1 (segmento 3); il contatto di confronto rimane chiuso in quanto $VW100 = 2$ e $VW200 = 1$

Diamo un impulso all' ingresso I0.4 : la VW200 assume valore 3 (segmento 4); di conseguenza il contatto di confronto si apre ($VW100 = 2$, $VW200 = 3$) disabilitando Q0.0.

Se infine diamo un impulso all' ingresso I0.5 li valori di entrambe le word tornano a 0 (segmenti 6 e 7).

N.B. Selezionando **stato del programma** dal menù **test** (barra degli strumenti in alto) i valori delle word vengono visualizzati sul contatto di confronto.

Istruzioni matematiche incrementa e decrementa (INC_W, DEC_W)

Questa istruzione assegna ad una word un determinato valore, aumentato o diminuito di una unità.

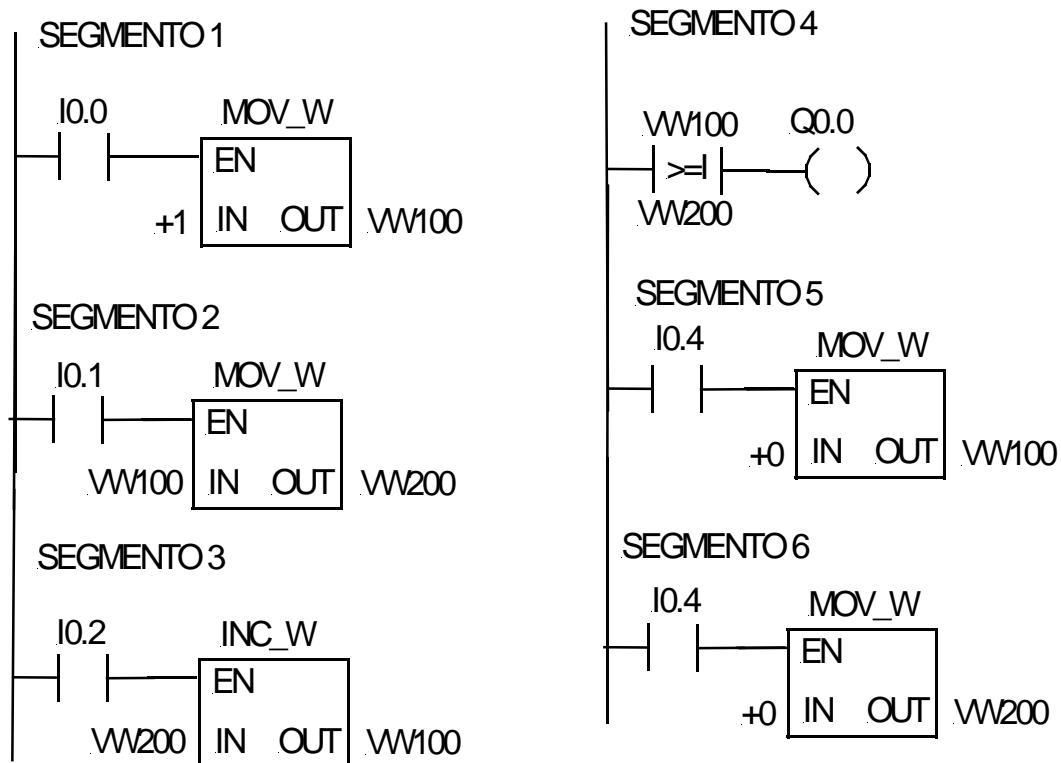
Nel diagramma vengono configurate due word. Nei segmenti 1, 2, 5, 6 sono state impostate le istruzioni di trasferimento, già viste; nel segmento 3 l'istruzione di incremento. Nel segmento 4 è stato impostato il contatto di confronto, che in questo caso risulta chiuso quando il valore di VW100 è maggiore o uguale rispetto al valore di VW200.

Abilitiamo con un impulso l'ingresso I0.0: la VW100 assume valore 1 (VW100 = 1; VW200 = 0).

Diamo un impulso all'ingresso I0.1: l'istruzione del 2° segmento assegna alla VW200 il valore copiato dalla VW100, quindi 1 (VW100 = 1; VW200 = 1).

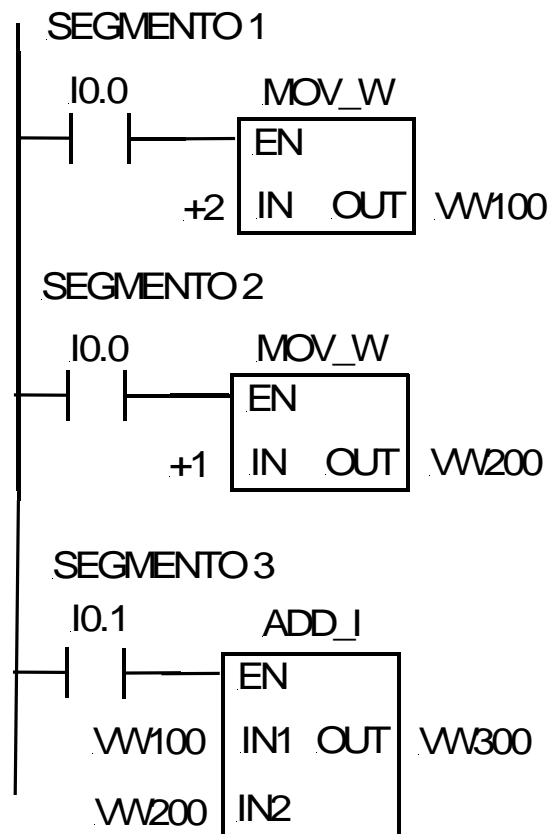
Diamo un impulso all'ingresso I0.2: l'istruzione di incremento assegna alla VW100 valore 2, ovvero il valore della VW200 aumentato di una unità (VW100 = 2; VW200 = 1).

Con un impulso all'ingresso I0.4 azzeriamo i valori di entrambe le word.



Operazioni matematiche (ADD_W, SUB_W, MUL_W, DIV_W)

Questa istruzione assegna ad una word il valore pari alla somma, differenza, prodotto o quoziente tra due valori.



Nei primi due segmenti sono state impostate due istruzioni di trasferimento, nel segmento 3 un'istruzione di addizione.

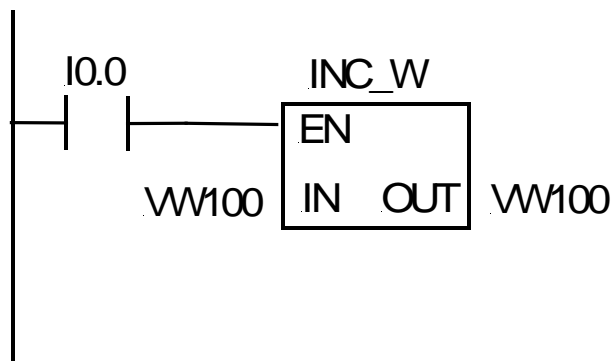
Abilitando con un impulso l'ingresso I0.0 le word VW100 e VW200 assumono rispettivamente valori 2 e 1.

Diamo ora un impulso all'ingresso I0.1: la VW300 assume valore 3.

Aggiornamento del valore di una word

Un'istruzione, quando abilitata, aggiorna **continuamente** il valore della relativa word. Di conseguenza, in alcuni casi, il valore assegnato dipende anche dalla durata dell'impulso che abilita l'istruzione.

L'istruzione raffigurata assegna alla VW100 il suo stesso valore aumentato di una unità. Non appena abilitiamo l'ingresso I0.0 la word assume valore 1, ovvero il suo valore iniziale (che è 0) aumentato di 1. Nell'istante immediatamente successivo (parliamo di frazioni di secondo) assumerà valore 2, pari al valore appena assunto aumentato di 1. Il valore continuerà quindi ad aumentare fino a quando l'ingresso non viene disabilitato.



Distributore automatico

Come esempio pratico consideriamo un distributore di prodotti.

Agli ingressi del PLC colleghiamo tre sensori che leggeranno le monete da 10, 20 e 50 cent. e tre pulsanti per la selezione del prodotto.

Alle uscite colleghiamo tre attuatori che provvederanno all'erogazione di altrettanti prodotti (spesso si tratta di un motorino che aziona un albero elicoidale, tra le cui spire viene sistemato il prodotto).

Passiamo ora al diagramma.

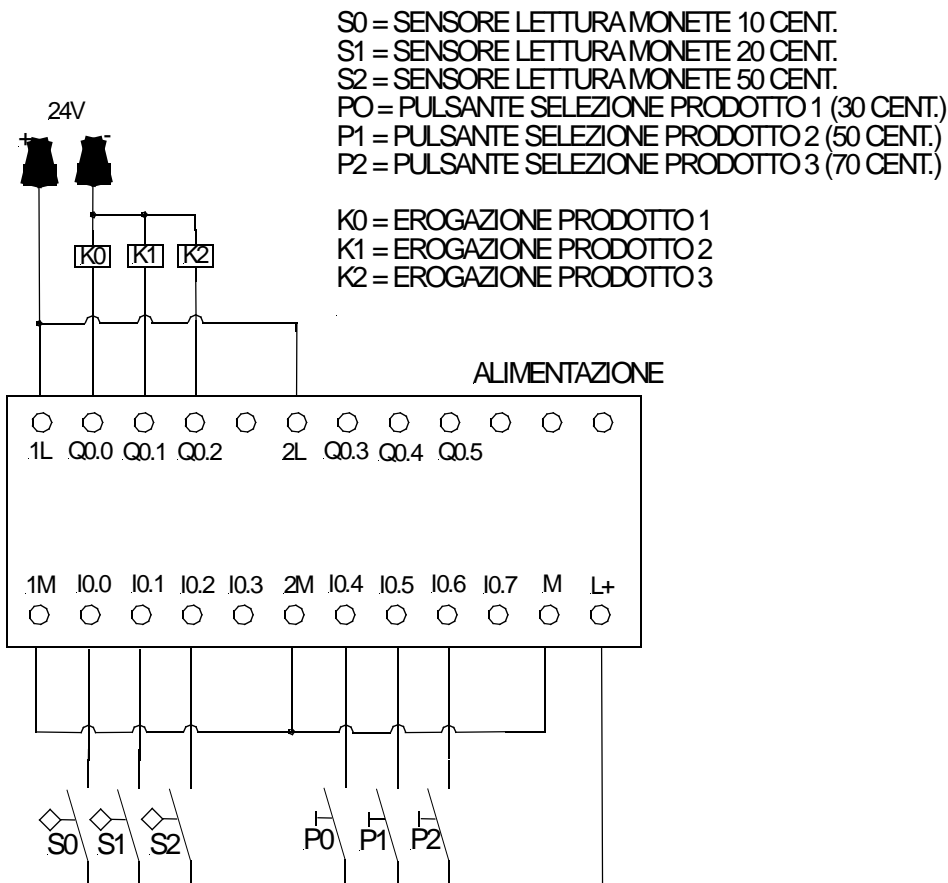
Nei segmenti da 1 a 6 sono state impostate le istruzioni per il conteggio del credito inserito, che viene cumulato nella VW200. Il credito utilizzato viene cumulato nella VW500 mediante le istruzioni dei segmenti da 8 a 13.

Nel segmento 14 viene impostata la sottrazione che assegna alla VW700 il valore corrispondente al credito residuo ($VW200 - VW500$).

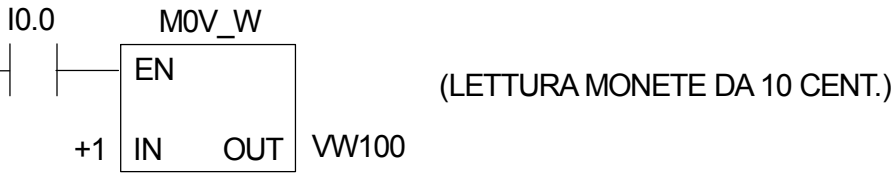
Nei segmenti 15-16-17 vengono configurate le uscite che comandano gli attuatori.

Nel momento in cui una moneta impegna il relativo sensore, il valore corrispondente viene assegnato alla VW100 (segmenti 1-2-3). Non appena il sensore torna a riposo il valore della VW100 si azzerà (seg. 4), ma un istante prima (frazione di secondo) viene caricato nella VW200 (seg.5). Di conseguenza occorre ritardare di un istante l'istruzione di azzeramento, rispetto all'addizione: a tale scopo è stata configurata la VW300. Il meccanismo è analogo nei segmenti da 8 a 13.

Per impedire l'erogazione del prodotto con credito insufficiente è stato impostato un confronto tra il valore di VW700 ed il valore corrispondente all'importo di ciascun prodotto.



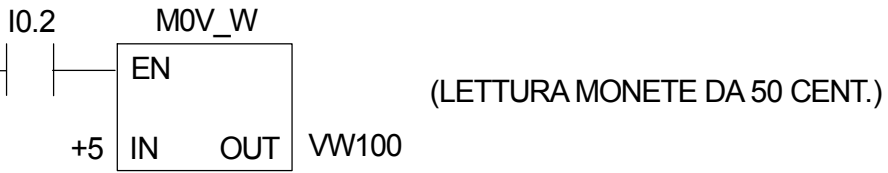
SEGMENTO 1



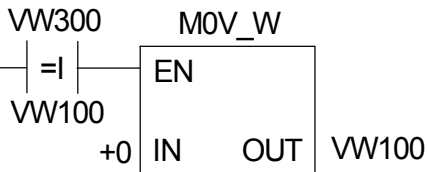
SEGMENTO 2



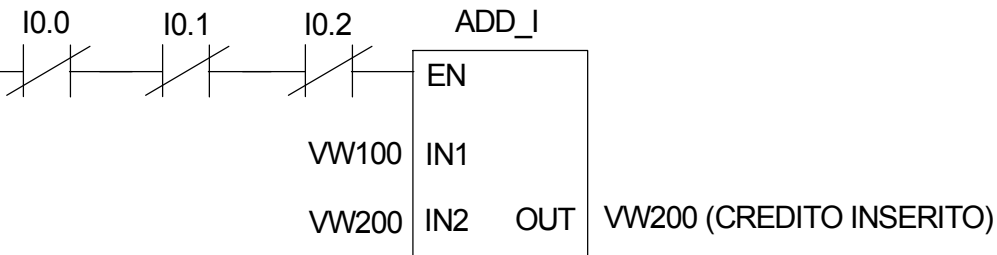
SEGMENTO 3



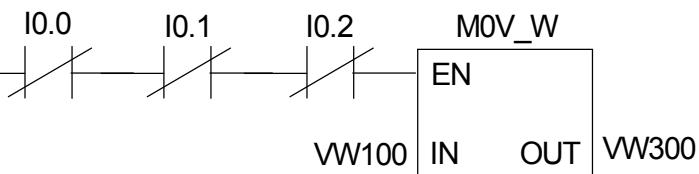
SEGMENTO 4



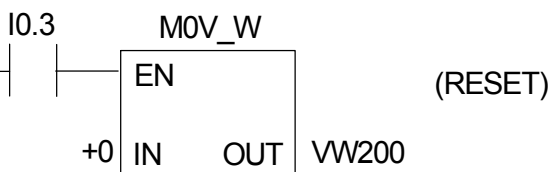
SEGMENTO 5



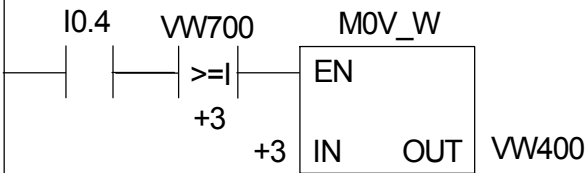
SEGMENTO 6



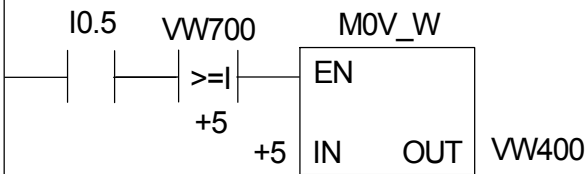
SEGMENTO 7



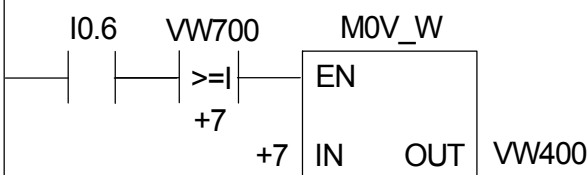
SEGMENTO 8



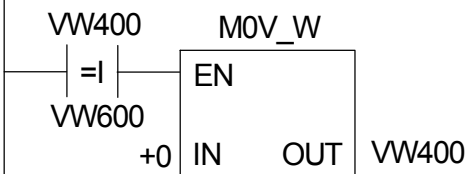
SEGMENTO 9



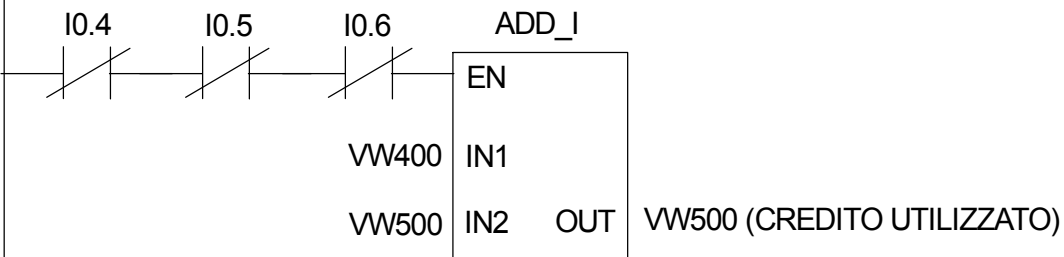
SEGMENTO 10



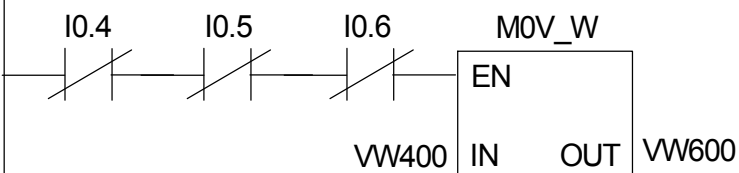
SEGMENTO 11



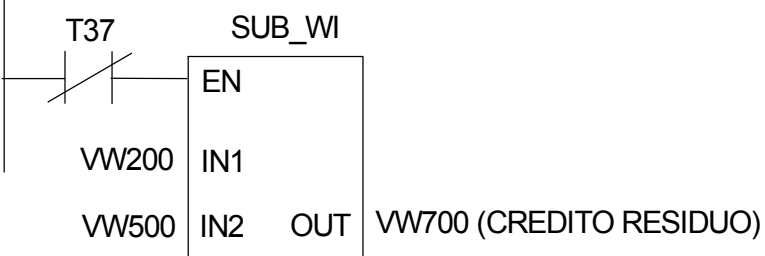
SEGMENTO 12



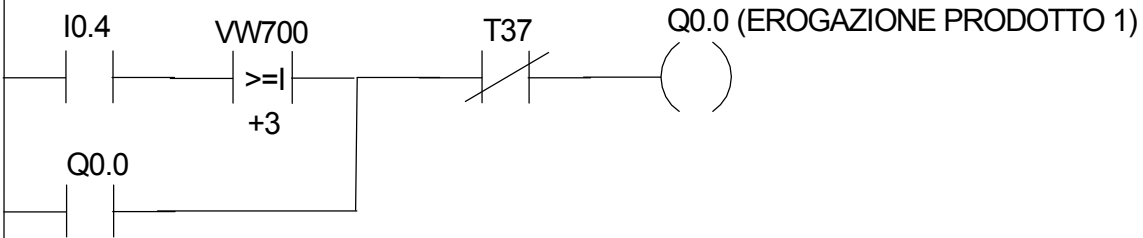
SEGMENTO 13



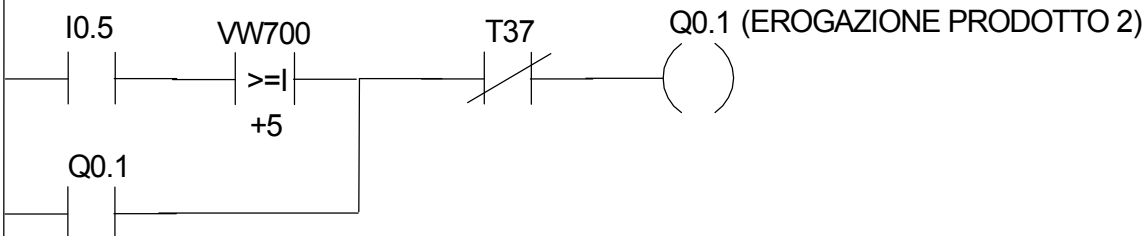
SEGMENTO 14



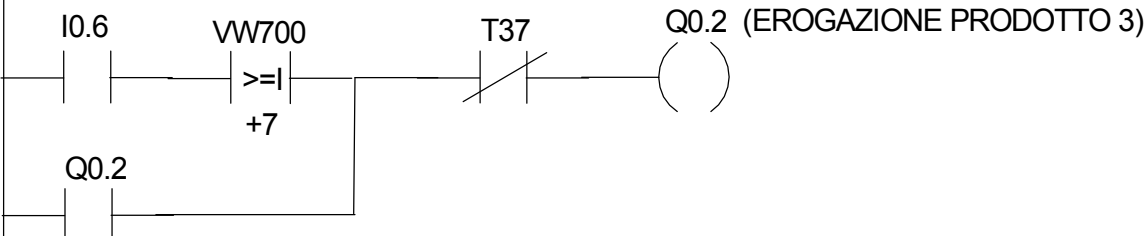
SEGMENTO 15



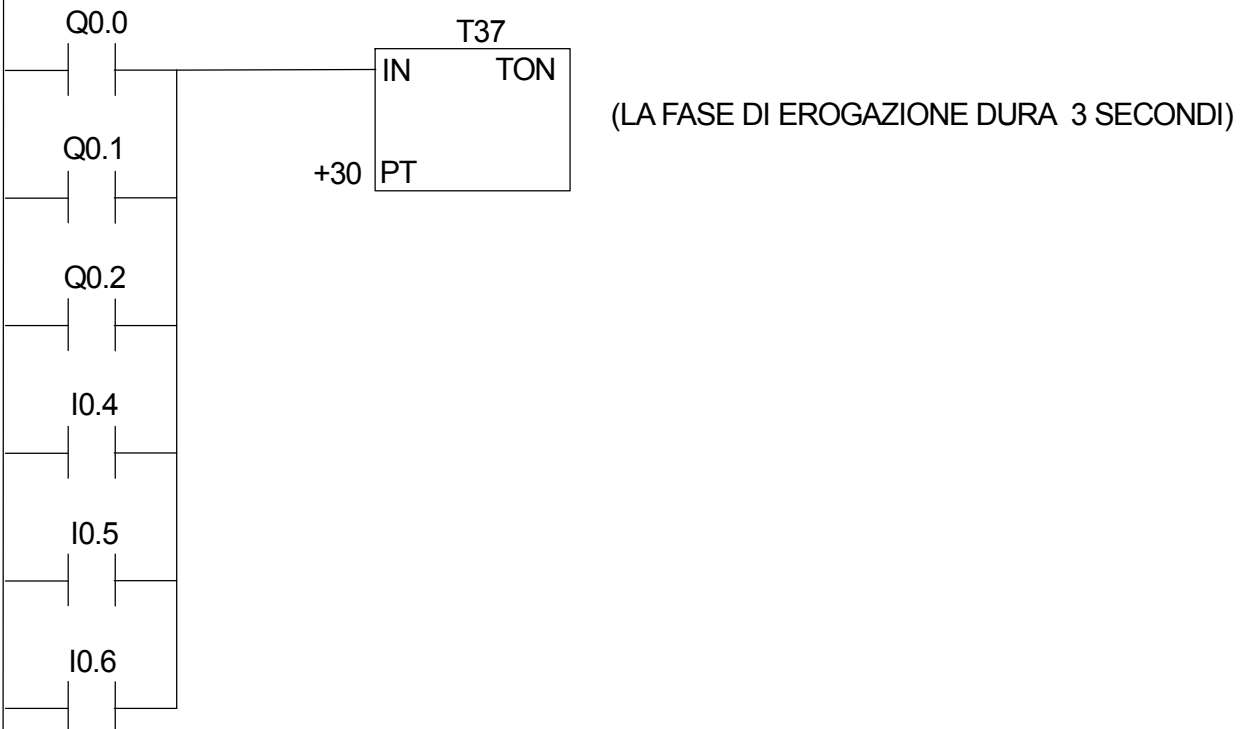
SEGMENTO 16



SEGMENTO 17



SEGMENTO 18



SEGMENTO 18

