



## 2.7 LINGUAGGIO LADDER

“ Si fa riferimento al linguaggio LADDER messo a punto della società FESTO a cui va un particolare ringraziamento dell’Autore per la disponibilità sempre dimostrata dai suoi tecnici, nel fornire delucidazioni e consigli e la concessione ottenuta ad utilizzare sia il materiale fotografico che la documentazione fornita dalla Società.”

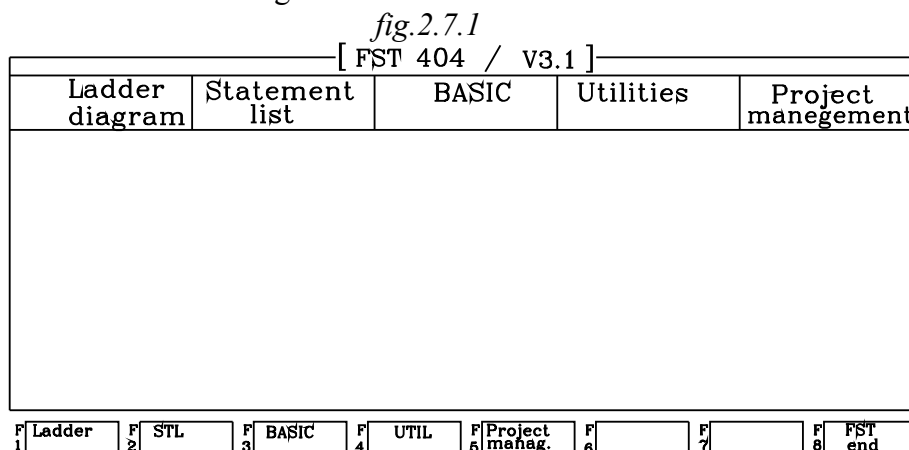
### SCHEMA A CONTATTI PROGRAMMA FST FESTO

Il file di lancio è:

F S T 4 0 4

Si ottiene una prima schermata con la presentazione del programma.

Pigiato un qualsiasi tasto si ottiene la seguente videata:



Sul bordo superiore vi è la barra dei menu con le diverse opzioni.

Nel bordo inferiore, entro riquadri, compaiono i comandi di attivazione delle opzioni del menu, che si possono effettuare pigiando i tasti funzione o cliccando ivi con il mouse.

Selezionare l'opzione :

**Projet  
Manegement**

La selezione si può ottenere o portandosi sull'opzione con il tasto freccia → e pigiando l'invio ↵, oppure cliccando l'opzione con il mouse.

Lo stesso risultato si può ottenere anche pigiando il tasto funzione F5

*fig.2.7.2*

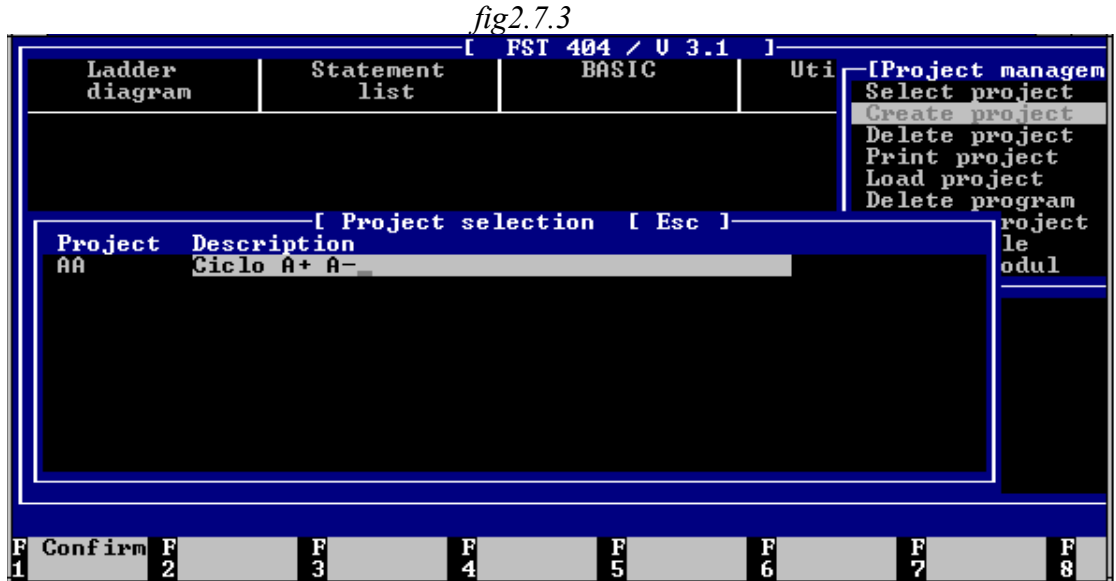


Si apre un menu a tendina con varie opzioni: per selezionare un progetto già memorizzato, oppure per crearne uno nuovo, o cancellare un progetto memorizzato ecc.

Si vuole ora creare un nuovo progetto: occorre quindi selezionare:

**Create project**

Si apre un riquadro di dialogo:



nel quale viene richiesto di dare un nome al file da creare: sia *AA*.

Pigiato l'INVIO ↵ il cursore si sposta nella zona di commento. Si scriva, per esempio, "*Ciclo A+ A-*".

Osservando la schermata generale della figura *fig.2.7.3* si nota che nel primo riquadro, a sinistra della schermata generale, riferentesi al tasto funzione *F1*, vi è il comando di conferma.

Pigiare il tasto funzione *F1*

Compare il nome del programma "*AA*" con il commento in fondo a sinistra della finestra.

Supponiamo di dovere effettuare il ciclo:

$$A^+ A^-$$

con l'impiego di una valvola di potenza monostabile.

### 2.7.1 INGRESSI USCITE

Come si è già detto il PLC è composto da più moduli contrassegnati da un numero. Il principale è il modulo 0.

Ogni modulo è composto da un numero di ingressi e uscite, corrispondenti ad una potenza della base 2.

Nel PLC in analisi ogni modulo ha 8 ingressi e 8 uscite

Nel programma a contatti *FST 404* gli ingressi sono indicati con *I*.

Nel modulo principale gli ingressi si indicano con *I0, I1, I2...I7*, sottintendendo che si riferiscono al modulo 0 (zero).

Negli altri moduli viene prima espresso il numero corrispondente al modulo e poi quello del particolare ingresso considerato. Così l'ingresso 0 del modulo 1 si indica con i caratteri *I1.0*

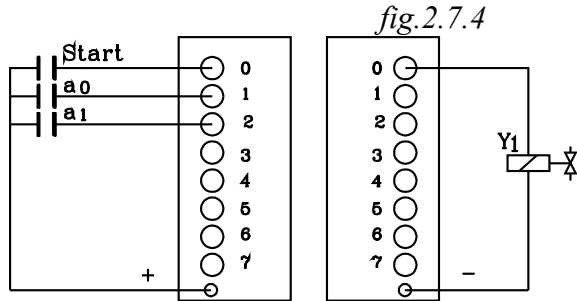
La scrittura *I2.1* sta ad indicare l'ingresso 1 del 2° modulo.

Le uscite vengono indicate con *O*.

Nel modulo principale le uscite si indicano con:  $O0, O1, \dots, O7$ , sottintendendo che si riferiscono al modulo 0.

Negli altri moduli viene prima espresso il numero corrispondente al modulo e poi quello della particolare uscita considerata. Così l'uscita 0 del modulo 1 si indica con  $O1.0$ .

La scrittura  $O3.1$  sta ad indicare l'uscita 1 del 3° modulo



Agli ingressi vengono collegati i finecorsa, gli Start, i sensori di prossimità ecc..

Il collegamento va fatto: partendo dal potenziale positivo, andando ad un capo dell'elemento fisico di ingresso e connettendo l'altro capo di questo ad un ingresso della PLC.

Per esempio nella figura è rappresentato il

finecorsa  $a_0$  collegata con un estremo al potenziale positivo comune e l'altro all'ingresso 2. Se il modulo è il principale l'ingresso si indicherà con  $I2$ . Se si trattasse del modulo 1, tale ingresso si indicherebbe con  $I1.2$ .

### 2.7.2 Lista occupazioni

Prima di iniziare il programma occorre stendere la lista delle occupazioni, nella quale vengono definite le occupazioni degli ingressi e delle uscite; in modo tale che sia definita l'associazione di ogni elemento degli ingressi ad uno specifico di un modulo e così pure per le uscite: ogni attuatore venga associato ad una uscita specifica di un modulo.

Occorre porre attenzione che il programma richiede non meno di tre caratteri, per indicare gli operatori di ingresso e di uscita simbolici posti dall'utente.

TABELLA DELLE OCCUPAZIONI					
INGRESSI			USCITE		
Tipo	Simbolo	Operatore assoluto	Tipo	Simbolo	Operatore assoluto
Start	Start	I0	Alettrovalvola $Y_1$	EVY1	O0
finecorsa $a_0$	fca0	I1			
finecorsa $a_1$	fca1	I2			

Le equazioni logiche già considerate nella elettro pneumatica sono:

$$mem = (Start \cdot a_0 + mem) \cdot \bar{a}_1$$

$$Y_1 = mem$$

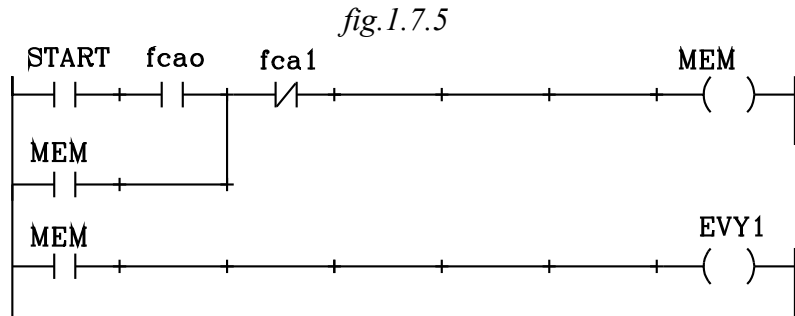
Dove  $mem$  indica una bobina ausiliaria che comanda l'elettrovalvola  $Y_1$ .


### 7.2.3 Inserimento rami di corrente

Nel PLC la bobina ausiliaria è sostituita da un *flag*: elemento di memoria che registra lo stato logico di un segnale e che può essere scritto e letto più volte.

Qui lo stesso circuito che verrebbe effettuato con tecnica elettropneumatica viene impiegato per realizzarlo attraverso il *PLC*. In effetti con il *PLC* può essere direttamente interrogato lo stato dell'elettrovalvola  $Y_I$  e non servirebbe l'impiego del *flag* "mem" (vedi oltre).


Le equazioni logiche si traducono nel circuito riportato in figura, dove i contatti sono contrassegnati con simboli con più di 3 caratteri.



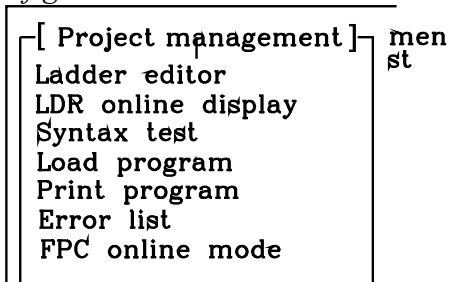
Sulla barra dei menu, con il tasto freccia  portarsi sulla opzione

**Leader  
Diagram**

e pigiare *INVIO* ↵

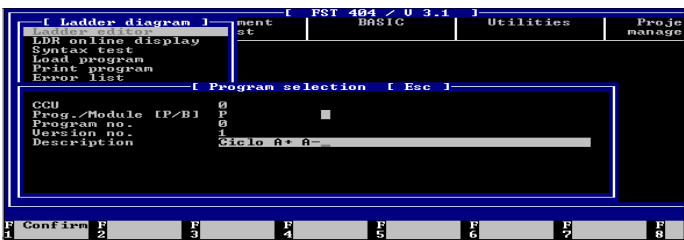
Lo stesso risultato si ottiene pigiando il tasto funzione  , oppure cliccando con il mouse su detta opzione.

*fig.2.7.6*



Si apre un menu a tendina con più opzioni.  
Selezionare *Ladder editor*

Con la selezione dell'opzione *Ladder editor* si apre un riquadro di dialogo



*fig.2.7.7*

Interessante è l'opzione:

### Version no. 1

Con lo stesso nome del progetto si possono avere diverse versioni.

Supponiamo infatti di volere un'altra versione del ciclo A+ A-, che si possa avviare con due start posti in AND e il cui file abbia lo stesso nome. Ebbene, è possibile (vedi oltre) richiamare il file AA , effettuare la modifica suddetta nel programma precedentemente digitato, quindi salvarlo con lo stesso nome AA ma con versione diversa.

### Version no. 2

Nel nostro caso si tratta di digitare la prima versione, quindi non occorre cambiare il numero 1 posto di default.

Un'altra opzione è :

### Description no comment

Portiamo il cursore con il tasto freccia su detta opzione e scriviamo:

*Description* Ciclo A+ A-

Pigiare quindi il tasto funzione **F1** (confirm).

Pigiato F1 si entra nella finestra di editor, dove compare una prima riga di corrente:

fig2.7.8



In basso, compare una barra dei menu, contenente le opzioni per editare il circuito a contatti.

Dette opzioni a loro volta contengono

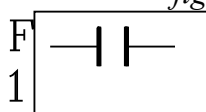
dei sottomenu.

Descriviamo i principali , lasciando lo studio degli altri in occasione di soluzioni di operazioni specifiche alle quali essi si riferiscono.

Le opzioni selezionate dai tasti funzione **F1** **F3** servono per poter porre sulla zona di test della linea di corrente i contatti con le rispettive etichette.

Tasto funzione **F1**

fig.2.7.9



Con il tasto funzione **F1** si digitano i contatti che possono essere **NA** (normalmente aperti) **NC** (normalmente chiusi)

Pigiando **F1** si apre un riquadro con l'indicazione di diversi tipi di contatti da poter selezionare

fig.2.7.10

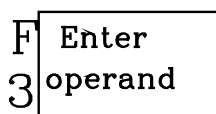


Tra questi si noti il contatto normalmente aperto e quello normalmente chiuso

Per selezionare un tipo si porta con il tasto freccia l'evidenziatore su di esso e si pigia **↵**, oppure si clicca l'opzione con il mouse.

Tasto funzione **F3**

fig.2.7.11



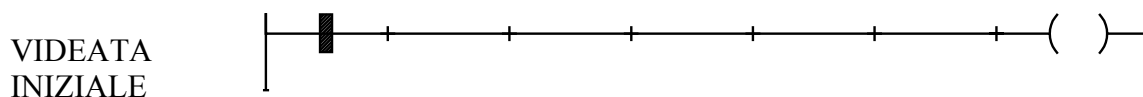
Serve per introdurre i simboli dei contatti.

I contatti si possono indicare con dei nomi simbolici significativi. A detti nomi va poi associato l'operatore assoluto ( di ingresso, uscita, flag...) riconosciuto dal programma; come, ad esempio, è stato indicato nella tabella delle occupazioni

Per meglio comprendere la manualità della digitazione conviene proseguire, seguendo, passo - passo le singole operazioni necessarie per scrivere il circuito precedentemente indicato

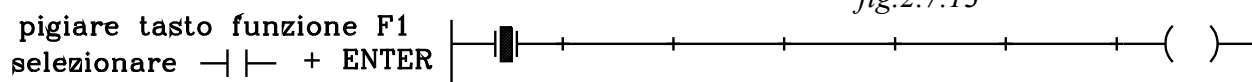
La videata iniziale, senza contatti si presenta con una linea con più passi nella zona di test e l'indicazione di un contattore nella zona di azione:

fig.2.7.12



- Per porre il primo contatto pigiare il tasto funzione **F1** , selezionare il contatto normalmente aperto" **Ä**] [**Ä**" , quindi pigiare l'INVIO . Si ottiene:

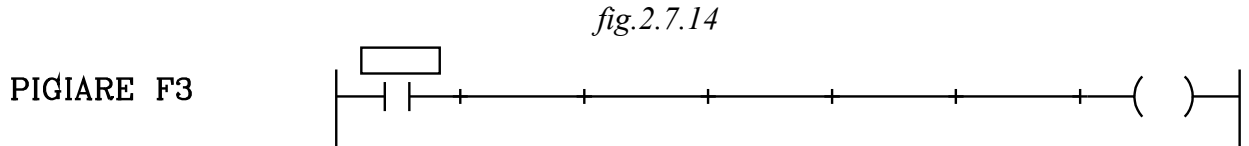
fig.2.7.13



Occorre ora designare il contatto con il nome simbolico: START

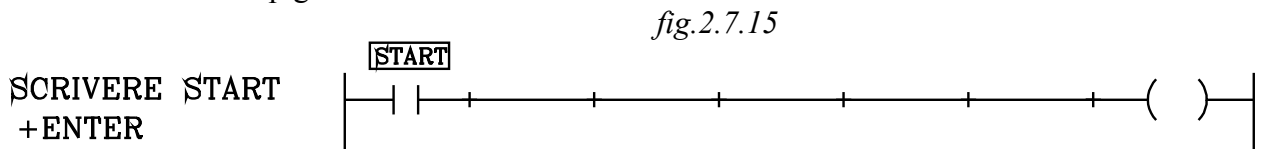
- Pigiare il tasto funzione *F3* (enter operand)

Sopra il simbolo del contatto si evidenzia un rettangolo di scrittura nella quale verrà digitato il nome simbolico:

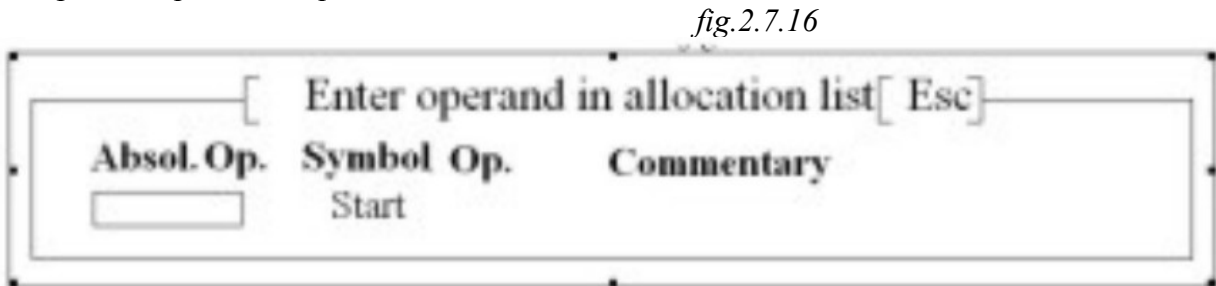


Per il simbolo del contatto usare non meno di 3 caratteri.

- Scrivere *START* e pigiare *l'INVIO* ↵

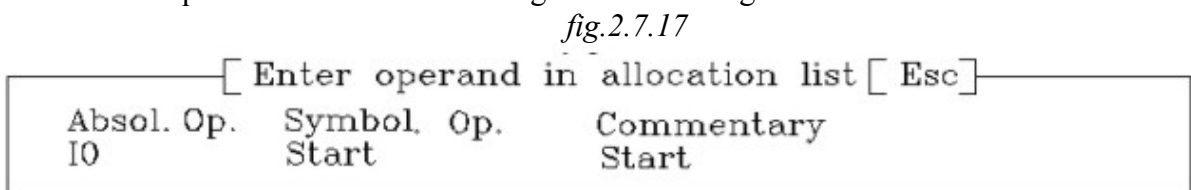


Si apre un riquadro nel quale vi sono le richieste di tre indicazioni:



- Viene richiesto l'operatore simbolico (*Synbol. Op.*). Questo è stato già immesso col nome di *Start* che compare sotto la richiesta del simbolo.
- Viene richiesto l'operatore assoluto, quello che il programma riconosce come ingresso, uscita flag, ecc.  
In questo caso lo *Start* viene associato all'ingresso *I0* (vedi tabella delle occupazioni).
- Viene richiesto un commento *Commentary*. In questo si indichi l'elemento associato al contatto. In questo caso lo *Start*.
- Scrivere *I0* sotto "*Absol. Op.*" e pigiare *INVIO*- L'evidenziatore si sposta sotto *Commentary*- Scrivere *Start*.

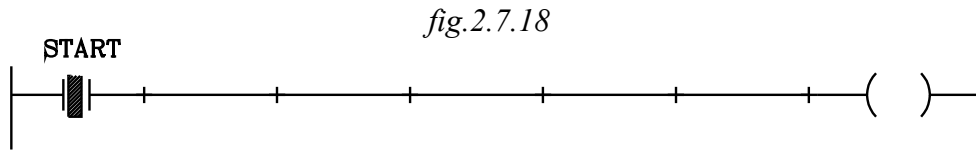
Si ottiene il riquadro con le indicazioni digitate come in figura:



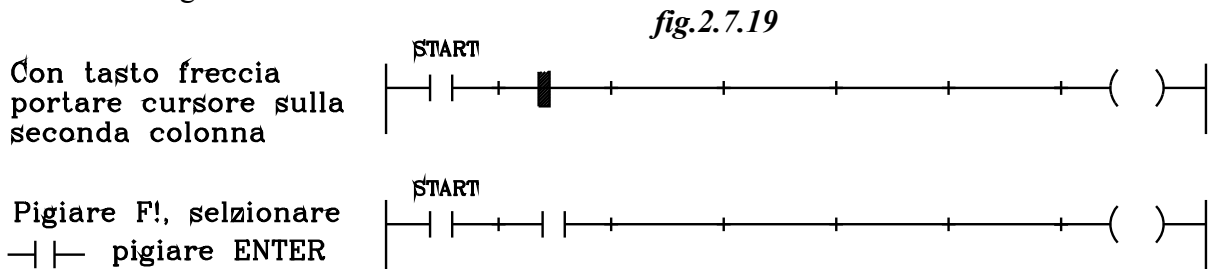
Alla fine occorre confermare le impostazioni pigiando il tasto funzione **F1**

- Pigiare il tasto funzione *F1* (enter).

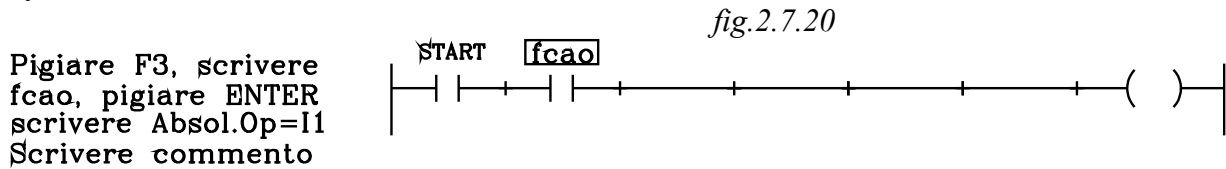
Si ottiene così il primo contatto NA contraddistinto dall'operatore simbolico "START" e associato all'ingresso, contraddistinto dall'operatore assoluto *I0*.



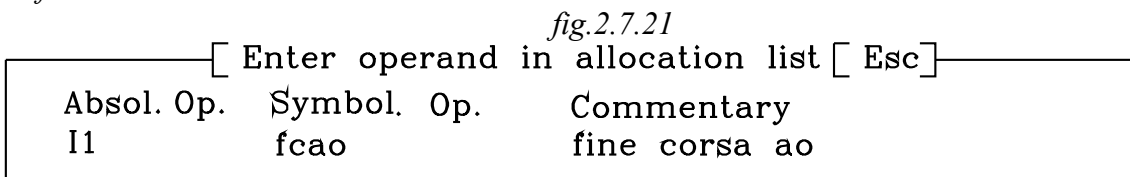
Operiamo alla stessa maniera per il contatto NA *fcao*, che si riferisce al finecorsa *ao*, associato all'ingresso *I1*.



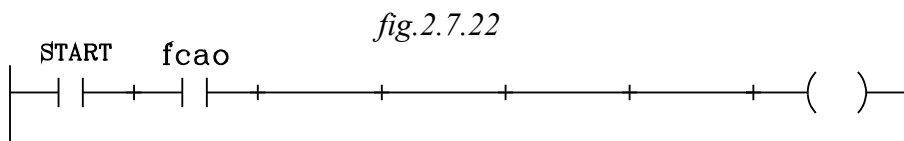
Si introduce, pigiando il tasto funzione *F3*, l'operatore simbolico del finecorsa *ao*, indicato con "*fcao*".



Introdotta l'operando simbolico *fcao* e pigiato *INVIO*, viene richiesto, come per il contatto precedente, il simbolo assoluto e il commento: associamo ad *fcao* l'ingresso *I1* e come commento "*fine corsa ao*".



Alla fine confermare con **F1**. Si ottiene:

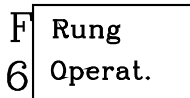


Occorre ora inserire un ramo parallelo. Le particolari operazioni di inserimento di rami di cancellazione ecc. sono previste nel sottomenu "*Rung Operat.*" Selezionare con il tasto funzione **F6** l'opzione:

**Rung operat.**



fig.2.7.23



- Con il mouse cliccare il riquadro contenete l'opzione "Rung Operat."

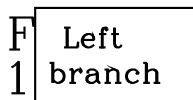
- Selezionato "Run operat" le sue opzioni compaiono nei riquadri di selezione nel bordo inferiore.

fig.2.7.24



Qui di seguito vengono riportate le opzioni di "Run operat." con accanto la loro utilizzazione.

fig.2.7.25



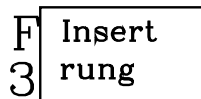
Per inserire a sinistra un ramo parallelo. Cliccato il riquadro, in esso compare il messaggio "Right branch" indicante l'inserimento a destra del ramo parallelo (vedi oltre nell'esempio).

fig.2.7.26



Per cancellare un ramo parallelo

fig.2.7.27



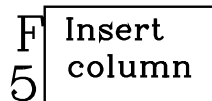
Per inserire una nuova linea di corrente

fig.2.7.28



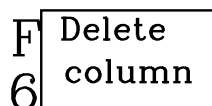
Per cancellare una intera linea di corrente

fig.2.7.29



Per inserire una colonna, e, quindi, avere la possibilità di porre sulla stesa riga contatti in più di quelli previsti de default: si aggiunge uno spazio.

fig.2.7.30



Per cancellare una colonna: si toglie uno spazio

fig.2.7.31

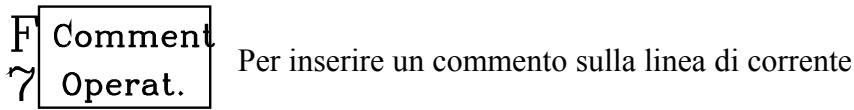
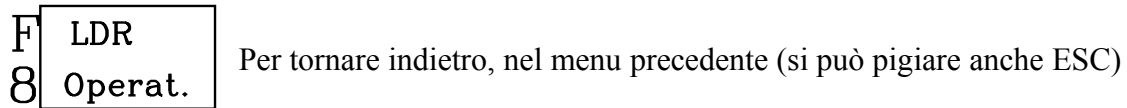


fig.2.7.32

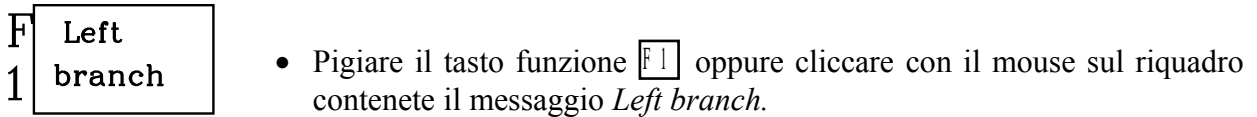


#### 2.7.4 Inserimento di un ramo parallelo

Come si è detto, occorre selezionare l'opzione "Rung operat." posta tra i riquadri di selezione principale di fig.4, ottenendo nel riquadro F1 l'opzione "Left branch"

Portare il cursore (con il tasto freccia o con il mouse) sulla prima colonna, a sinistra, contenente il contatto sul quale si vuole aprire il ramo parallelo: Il contatto START nel nostro caso (vedi schema di figura).

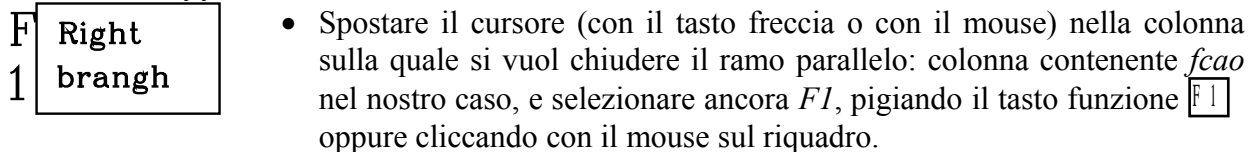
fig.2.7.33



Viene agganciato, a sinistra del contatto contenente *Start*, il ramo parallelo da digitare.

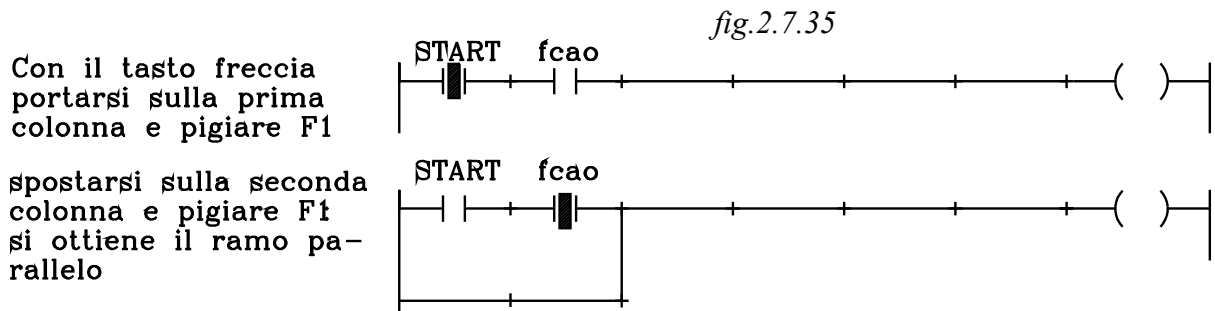
Osservare che si evidenzia l'aggancio (schema di figura) e il messaggio scritto nel riquadro di **F1** muta la sua indicazione in :

fig.2.7.34



Si ottiene il ramo parallelo.

Riassumendo:

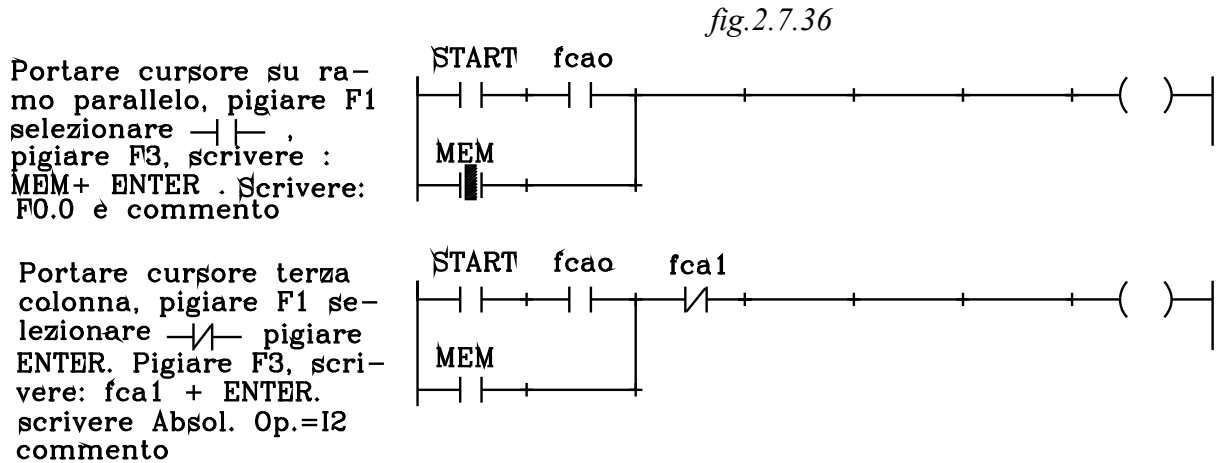


Occorre ora porre nel ramo parallelo il contatto normalmente aperto *MEM* di autoritenuta.

Per poter proseguire nella digitazione dei contatti e porre in essi i simboli che li rappresentano, occorre fare un passo indietro e ritornare nei precedenti menu principali.

- Per tornare da un sottomenu al menu principale pigiare sempre il tasto funzione F8.

Ora si prosegue nella digitazione degli altri contatti come riportato qui di seguito:



Occorre notare che il contatto di autoritenuta *MEM* si riferisce ad un *FLAG*. Questo è una cella di memoria appartenente ad una matrice a *m* linee ed *n* colonne. Viene contraddistinto da un operatore assoluto formato dal carattere *F* e da due numeri riferentesi alla riga e colonna di appartenenza: del tipo *F0.0 F0.1... F1.0 F1.1... ecc.*

Associamo al contatto di autoritenuta *MEM* il flag "*F0.0*".

Così, dopo aver introdotto il contatto normalmente aperto pigiando *F1* scritto MEM e pigiato l'INVIO, si apre il solito riquadro per l'immissione dell'operatore assoluto e il commento.

*fig.2.7.37*

[ Enter operand in allocation list [ Esc ]		
Absol. Op.	Symbol. Op.	Commentary
F0.0	MEM	Autoritenuta e comando Y1

Alla fine confermare con *F1*.

Per digitare il contatto normalmente chiuso *fca1* "Ä"/[Ä", riferentesi al finecorsa *a1*, si procede alla stessa maniera degli altri contatti:

- Pigiare il tasto funzione *F3* o cliccare sul riquadro.
- Selezionare il contatto *NC*.

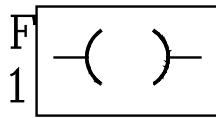
### 2.7.5 Scrittura nella zona di azione

Nella zona di azione vengono immessi gli attuatori, flag, temporizzatori, contatori.

Nel nostro caso occorre immettere l'espressione simbolica del flag *MEM*. Esso è stato già definito nell'introduzione del contatto riferentesi ad esso: non occorre quindi, di nuovo, associargli l'operatore assoluto e il commento: questi vengono ricordati dal programma.

- Portare, quindi, il cursore sulla zona di azione "Ä( )Ä".

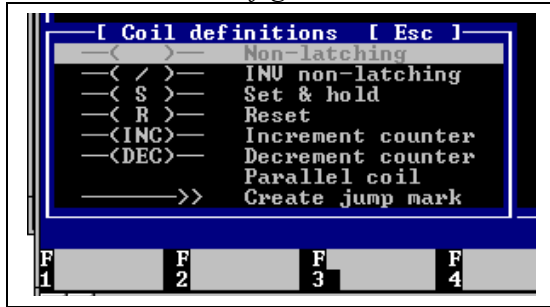
*fig.2.7.38*



Si noti che l'indicazione, nel riquadro di selezione del tasto funzione *F1*, posto in basso a sinistra, è cambiata: ora al posto del simbolo del contatto "Ä ] [Ä" compare "Ä( )Ä " che caratterizza il contattatore.

- Pigiare il tasto funzione *F1* o cliccare sul riquadro.

fig. 2.7.39



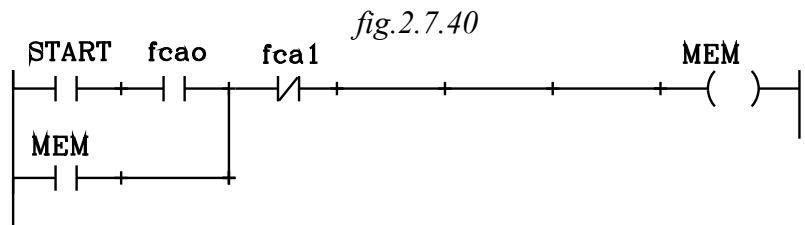
Si apre un riquadro di selezione, con l'indicazione del modo di attivazione del contattatore.

- Selezionare il modo non permanente Ä( )Ä
- A questo punto occorre immettere l'espressione simbolica *MEM*: pigiare il tasto funzione *F3*, scrivere *MEM* e pigiare il tasto *INVIO*.

Come già detto il programma non richiede altre specificazioni, avendo già definito l'operando *MEM*.

Riassumendo:

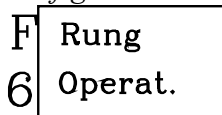
Portarsi nella zona di azione. Pigiare *F1*, selezionare ( ) +ENTER, pigiare *F3*, scrivere *MEM* confermare con *F1*



### 2.7.6 Commento della riga di corrente

Nello stampare una riga di corrente conviene che in essa compaia un commento di specificazione della operazione che verrà effettuata. In tal modo, si può più agevolmente verificare il programma passo passo una volta digitato.

fig.-2.7.41



- Nel menu principale, con il mouse cliccare il riquadro di selezione contenete l'opzione "*Rung Operat.*".

In basso compaiono nei riquadri le opzioni di "*Rung Operat.*".

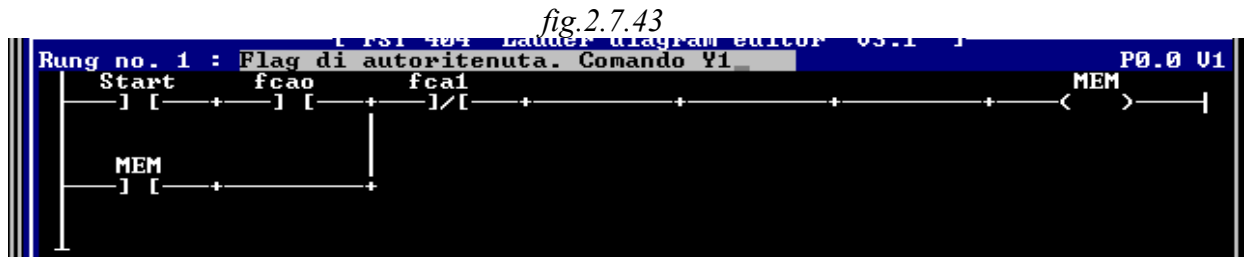
fig.2.7.42



- Selezionare l'opzione "*Comment*" : pigiare il tasto *F7* o cliccare con il mouse il riquadro

Sopra la linea di corrente si apre una barra di scrittura . Scriviamo in questa come commento:

"Flag di autoritenuta. Comando di Y1"

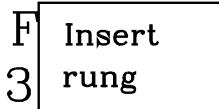


### 2.7.8 Inserimento di un'altra riga di corrente

Ultimata la riga di corrente del *Flag*, occorre inserire la riga successiva.

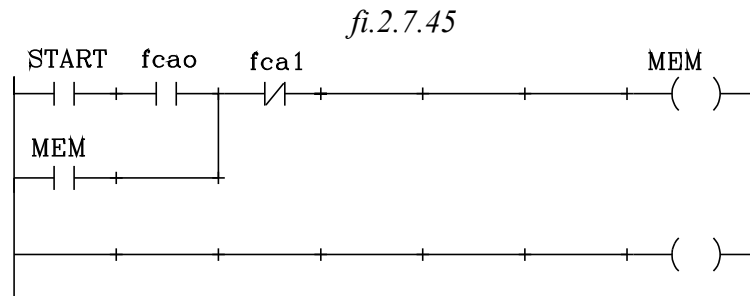
Nel sottomenu del "*Rung operat.*" (il medesimo nel quale si trova l'opzione *Comment* adoperato) vi è l'opzione "*Insert rung*".

*fig.2.7.44*



- Pigiare il tasto funzione *F3* o cliccare con il mouse il riquadro di selezione *Insert rung*

Sullo schermo si visualizza un'altra riga di corrente:

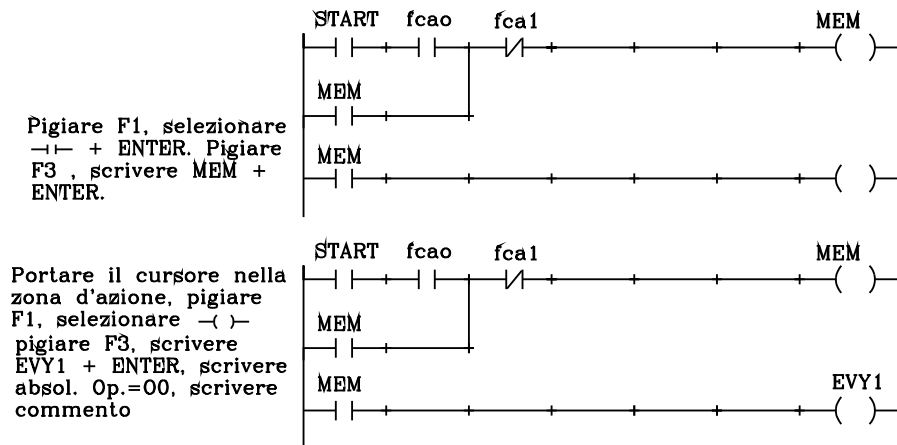


Occorre ora digitare la nuova linea di corrente, avente nella zona di test il contatto *MEM* e nella zona di azione l'uscita di attivazione della elettrovalvola *Y1*.

Nella tabella delle occupazioni si è posto come etichetta dell'elettrovalvola l'espressione simbolica *EVY1* e come operatore assoluto l'uscita *O0*.

Riassumendo le operazioni da effettuare si ha:

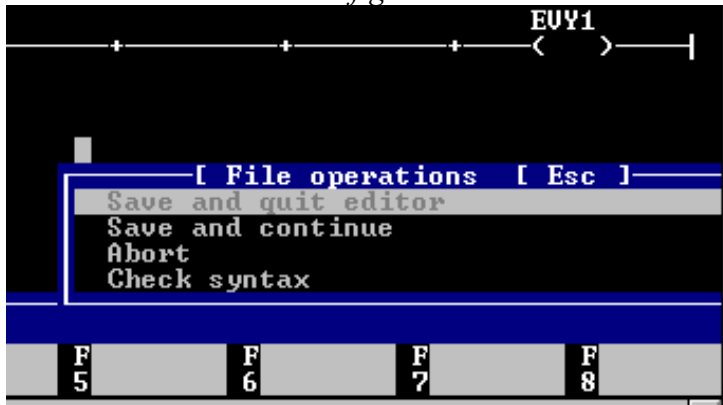
*fig.2.7.46*



- Ultimato lo schema a contatti pigiare il tasto funzione F8

NB Occorre ricordare che il tasto funzione F8 serve per retrocedere nel percorso dei menu. Così, se si è selezionata una opzione di un menu, per tornare indietro, nel menu principale si pigia F8

fig.2.7.47



Quando viene ultimata la digitazione dello schema a contatti e si pigia il tasto funzione F8, il programma fa apparire una finestra, in basso, a destra con diverse opzioni di selezione.

- Selezionare "Save and quit editor" (pigiare INVIO o clic con il mouse)  
 Si torna nel menu principale della prima schermata.

## 2.7.9 INTESTAZIONE PAGINA

La pagina di stampa può essere intestata.

Siamo ora nel menu principale rappresentato in fig.2.7.1

- Portarsi sulla opzione *Utilities*
- Selezionare l'opzione *Utilities* pigiando l'INVIO o cliccando su di essa con il mouse.

Si apre una finestra a tendina con diverse opzioni di selezione

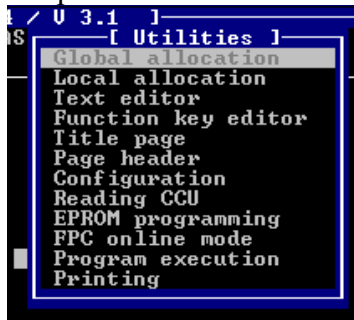
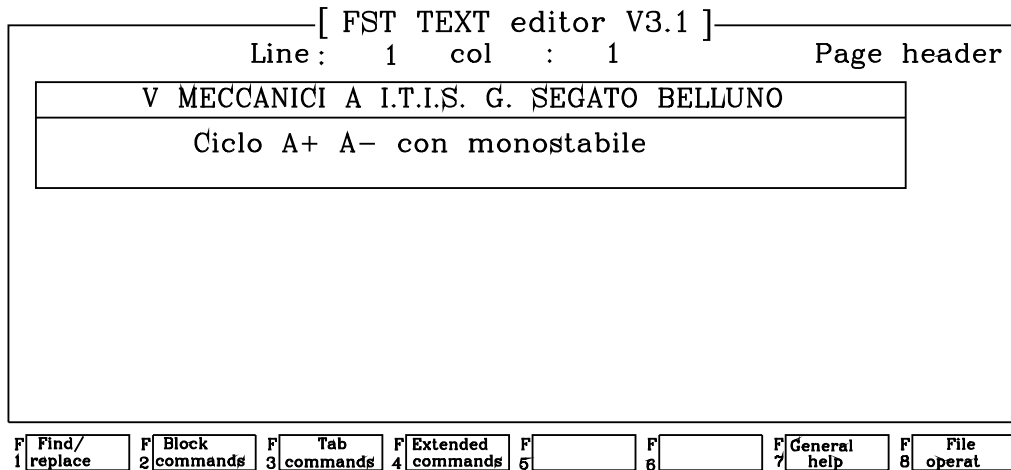


Fig.2.7.48

- Selezionare l'opzione "Pag heder" con il quale si può digitare una intestazione su ogni pagina che si vuole stampare.

- Selezionata l'opzione "Pag heder", si apre una finestra a tutto schermo di digitazione.

fig.2.7.49

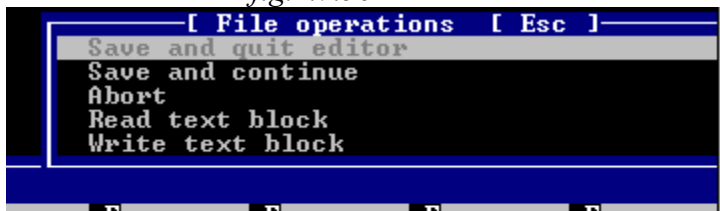


Nel riquadro vi è l'intestazione riferita alla FESTO. Si può scrivere sopra cambiando l'oggetto del messaggio. Di default si è in modo sovrascrittura, pigiando Ins si passa in modo inserimento.

Nell'esempio, entro l'area di squadratura, è scritta l'intestazione riferita all'I.T.I.S. G. SEGATO di BELLUNO, l'Istituto Tecnico nel quale insegna l'Autore di queste note; al disotto, è posta l'indicazione del programma: il semplice ciclo  $A^+$   $A^-$ , preso come modello per la digitazione.

- Come al solito, alla fine della digitazione pigiare il tasto funzione  $F8$ :

*fig.2.7.50*



Si apre un riquadro di selezioni con più opzioni.

- Selezionare *Save and quit editor*

## 2.7.10 Configurazione e stampa del programma

### Configurazione

Accertarsi che il computer sia collegato alla stampante

Dal menu principale *fig.2.7.1* scegliamo di nuovo l'opzione

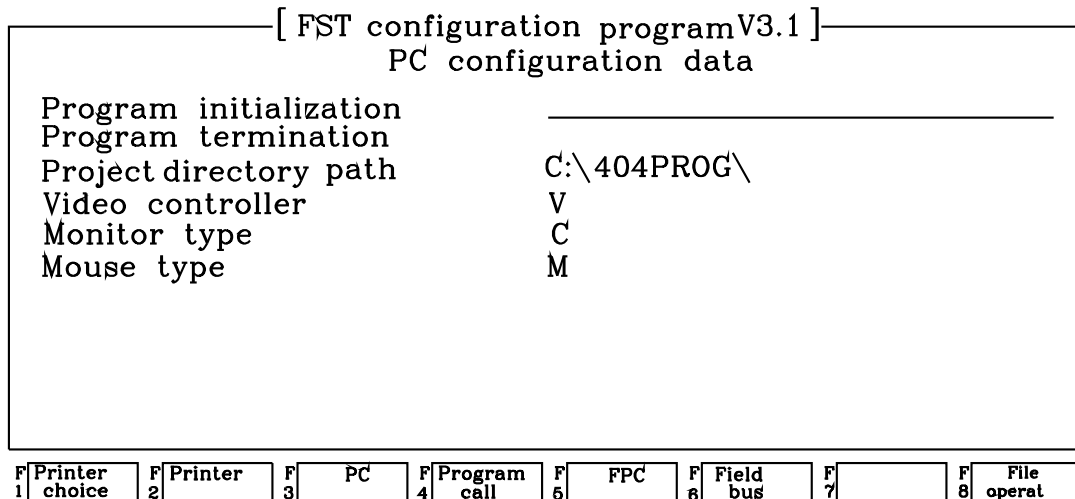
**Utilities**

Si apre il menu tendina di *fig.2.7.48* dal quale selezioniamo l'opzione:

**Configurations**

Si apre una finestra di dialogo.

*fig.2.7.51*

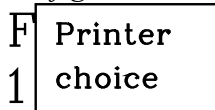


Nella finestra compaiono le opzioni di configurazione, impostate all'inizio, nella installazione del programma e precisamente:

Project directory path	C:\404\prog:	indica dove viene salvato il programma
Video controller	V	: scheda VGA
Monitor Type	C	: colori
Mouse	M	: mouse

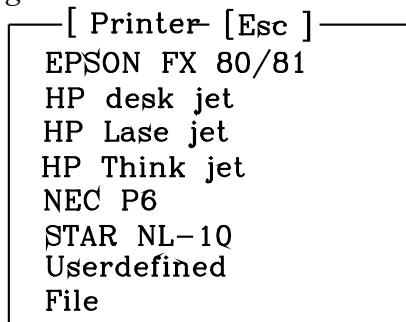
Nella riga inferiore compaiono altre opzioni.

fig.2.7.52



- Selezioniamo l'opzione Printer choice.

fig.2.7.53



Si apre un riquadro con le opzioni di selezione della stampante.

Selezionare tipo di stampante.  
Per esempio , la *NEC P6*.

- Pigiare il tasto funzione *F8* e quindi salvare.

fig.2.7.54



Dal menu sulla riga inferiore selezionare l'opzione *Printer*.

Si apre una finestra di dialogo



fig.2.7.55



Porre:

- Lines/pag 72
- Columns/line 80
- Pigiare il tasto funzione *F8* e salvare.

### Stampare il programma

- Torniamo al menu principale di *fig.2.7.1*
- Selezioniamo l'opzione:

**Ladder  
diagram**

- Dal menu a tendina di *fig.2.7.2* selezionare l'opzione *Print program*:

Lo schema a contatti digitato verrà stampato.

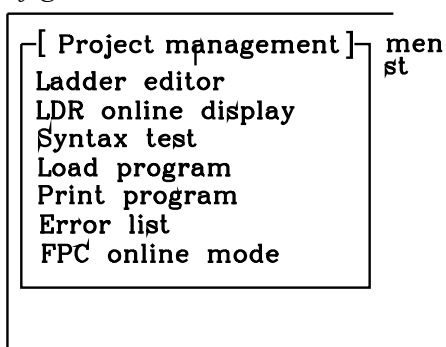
## 2.7.11 CARICAMENTO DEL PROGRAMMA NEL PLC

- Selezionare dal menu principale di *fig.2.7.1* l'opzione:

**Ladder  
diagram**

Si apre il menu a tendina di *fig.2.7.56*

fig.2.7.56

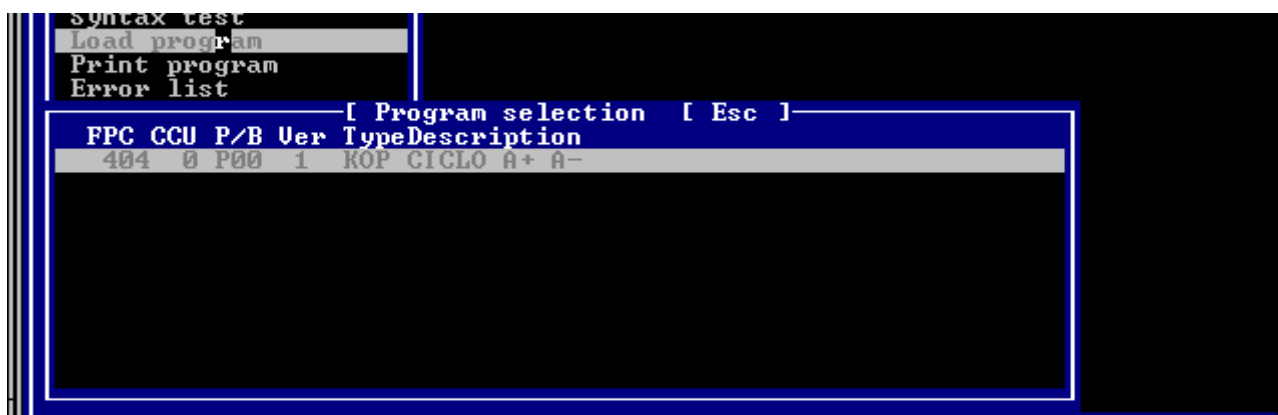


- Seleziona l'opzione *Load program*.

Si apre un riquadro nel quale sono indicate le versioni del programma selezionato (nel nostro caso una sola versione).

Viene evidenziata la riga indicante la versione selezionata (nel nostro caso un'unica riga).

fig.2.7.57



Se vi fosse un'altra versione che si volesse caricare , portarsi con tasto freccia su questa.

- Caricare quindi la versione voluta con *l'INVIO* (o clic con il mouse).

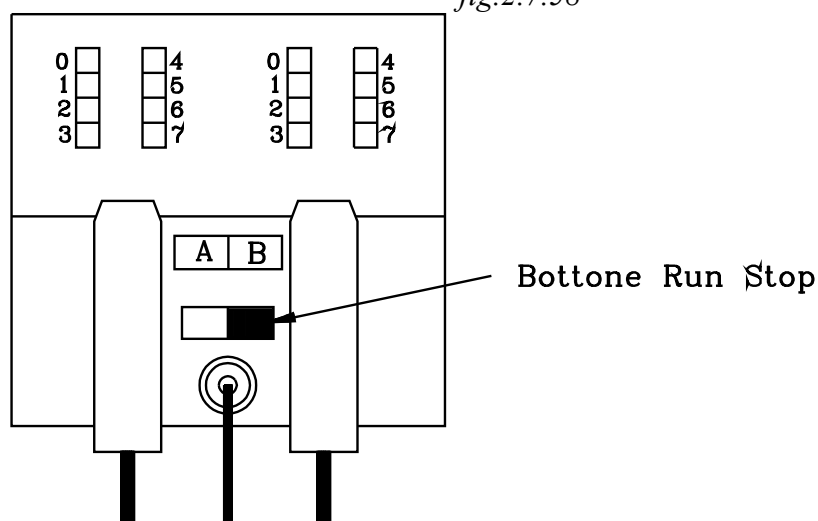
Aspettare che venga caricato il programma sul PLC.

Alla fine compare il messaggio:

*Download finished*

### 2.7.11.1 Operazione RUN/STOP

fig.2.7.58



Perché il programma possa essere eseguito occorre iniziarlo nel PLC.

Sulla parte superiore del modulo principale 0 sono posti: i led indicanti l'attivazione degli 8 ingressi e uscite, due spie *A* e *B* indicanti lo stato del sistema e una levetta di *RUN/STOP*.

In condizioni normali, prima di inizializzare il programma caricato nel PLC, risulta accesa la spia *A* di colore giallo, mentre *B* è spenta.

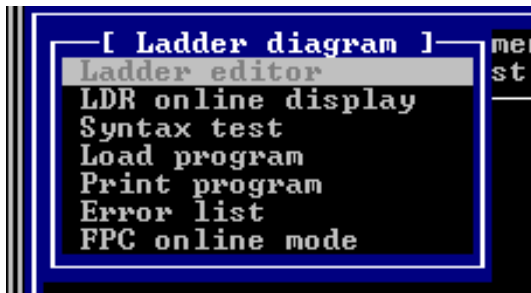
Per inizializzare il programma occorre effettuare l'operazione di *RUN/STOP*, consistente nello spostare la levetta, rappresentata in figura, prima a sinistra e poi riportarla nella posizione di destra " $\leftarrow \rightarrow$ ": Si deve accendere la spia *B* di luce verde.

A questo punto si può eseguire il programma

PIGIARE LO START

**2.7.12 FPC online mode**

*fig.2.7.59*



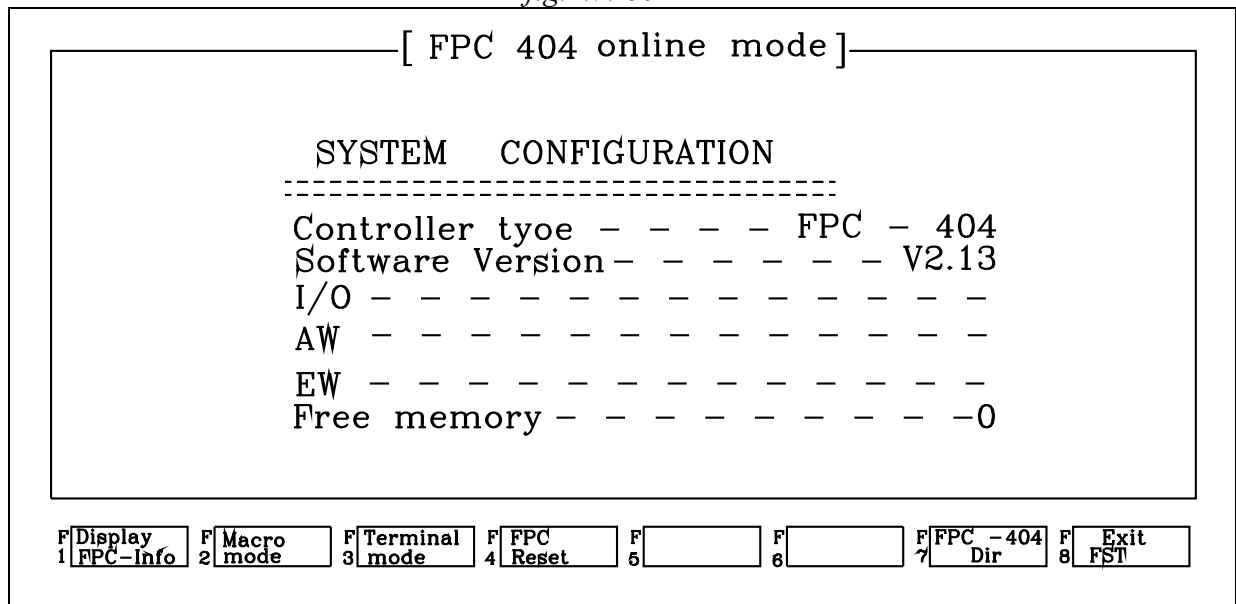
L'inizializzazione del programma può essere ottenuto con l'opzione *FPC online mode* di Ladder diagram.

- Dal menu a tendina Ladder diagram, selezionare l'opzione:

**FPC online mode**

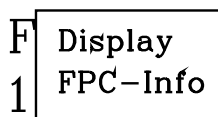
Si ottiene un altro schermo "SYSTEM CONFIGURATION", nel quale viene indicata la configurazione del sistema.

*fig.2.7.60*



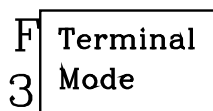
Nella riga inferiore compare un altro menu con diverse opzioni. Interessanti sono::

*fig.2.7.61*



Rende le informazioni sullo stato del sistema

*fig.2.62*



Accede al terminale. In tale ambiente si possono impartire ordini eseguibili: eliminare lo stato di errore, cancellare i programmi dalla memoria del PLC ecc.

### 2.7.12.1 PLC in stato di errore

**ATTENZIONE!** In condizioni di funzionamento normali la spia *B* sul modulo *0* o è spenta oppure è verde dopo il *Run/Stop*. Quando vi è un errore di sistema (connettori non allacciati ecc.) la spia diviene rossa (*può capitare che la batteria in tampono sia scarica*).

- Per togliere errori eliminabili, pigiare il tasto funzione *F3* per accadere nell'ambiente *Terminal mode*
- Al prompt ">" dei comandi digitare *mf*  
>*mf*

Viene segnalato il tipo di errore da consultare sul manuale

- Se lo stato di errore è eliminabile (per esempio è dovuto alla batteria scarica o al connettore che si è riallacciato) *digitare 0 e pigiare INVIO*

Ad esempio il messaggio, dopo la digitazione di *mf* può essere:

>*mf*=1,0,0,00:           digitare 0  
>*mf*=1,0,0,00: 0           pigiare ↵

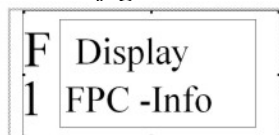
### 2.7.12.2 Per eliminare dalla memoria del PLC i programmi

- Digitare al prompt *y*.
- Pigiare ↵

### 2.7.12.3 Sottomenu dell'opzione Display FPC- info

Dal sottomenu *Display FPC -info* si hanno informazioni sullo stato del sistema e la possibilità di effettuare l'operazione di *RUN/STOP*.

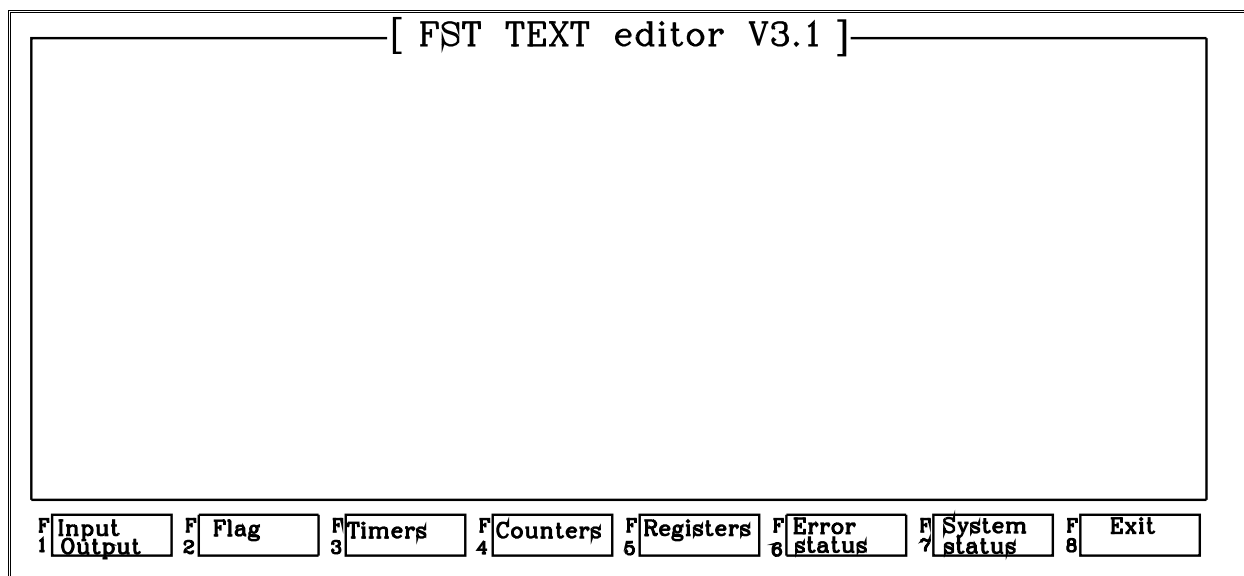
*fig.2.61*



- Pigiare il tasto funzione *F1* di selezione di *Display FPC -info*.

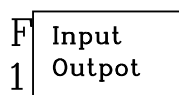
Si apre una nuova finestra. Nella riga inferiore compaiono nuove opzioni nei riquadri di selezione.

*fig.2.7.63*



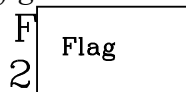
Nella riga inferiore compare un altro menu con le seguenti opzioni:

*fig.2.7.64*



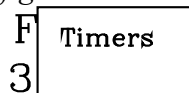
Vengono visualizzati gli ingressi e le uscite.

*fig.2.7.65*



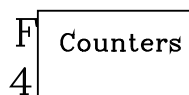
Vengono visualizzati i *flag*.

*fig.2.7.66*



Vengono visualizzati i *temporizzatori*.

*fig.2.7.67*



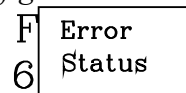
Vengono visualizzati i *contatori*.

*fig.2.7.68*

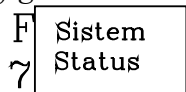


Vengono visualizzati i valori dei *registri*.

*fig.2.7.69*

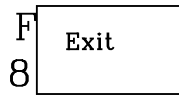


*fig.2.7.70*



Dà lo stato del sistema, che può essere allo stato *inattivo* o *attivo*.

*fig.2.7.71*



Uscita e ritorno al menu precedente.

- Per attivare ed eseguire il programma pigiare il tasto funzione *F7* ( System Status).

Compare:

Prog N.	Status	Proces	actual step
0	inactive	0	0

Con l'indicazione dello stato .

- Pigiare ancora il tasto funzione *F7* corrispondente a *RUN/*.

Si passa dallo stato inattivo all'attivo e compare la relativa indicazione nella colonna *Status*

Prog N.	Status	Proces	actual step
0	active	0	0

Il programma è pronto per essere eseguito :

*PIGIARE START.*

## 2.8 APPLICAZIONI - TEMPORIZZATORI - CONTATORI

### 2.8.1 Temporizzatori

Il PLC presenta la possibilità di programmare temporizzazioni.

Vi sono 16 temporizzatori indicati con  $T0 \div T15$ .

Per impostare una temporizzazione occorre:

- Attivare il temporizzatore attraverso la variazione del livello di un segnale attivante
- Immettere la durata di temporizzazione.

Una volta attivato, il temporizzatore emette un segnale il cui livello viene inviato all'Unità di Comando per verificare se è in atto o terminata la temporizzazione.

Iniziata la temporizzazione il segnale del temporizzatore si porta al livello logico 1:  $Ti=1$  (con  $i=0 \div 15$ ) per tutto il tempo prefissato, quindi, trascorso questo si porta al livello logico zero:  $Ti=0$ .

I temporizzatori possono essere interrogati allo stesso modo degli ingressi e le uscite.

L'unità minima di temporizzazione 0.02 sec. Il massimo valore che si può impostare è 10 min.

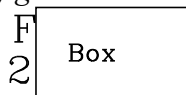
La durata viene immessa scrivendo prima il numero seguito dalla lettera *s* : per esempio " 3s" indica la durata di 3 secondi.

Nel programma *FST* sulla linea di corrente che attiva il temporizzatore vi sarà il contatto di attivazione: ad esempio uno *START*.

Per digitare la riga con l'attivazione del temporizzatore e l'impostazione del tempo effettuare i seguenti passi:

- Portare il cursore sulla zona di azione "Ä( )Ä" della riga di corrente.

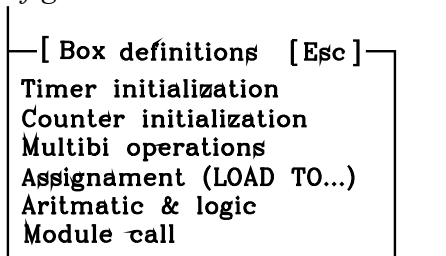
fig2.8.1



- Pigiare il tasto funzione *F2* di selezione del riquadro *Box* o fare clic su di esso con il mouse.

Si apre un riquadro di selezione.

fig.2.8.2



- Selezionare *Timer initialization*.

Sulla zona di azione si evidenzia una scatola con la parola *Timer* scritta nell'interno.

fig.2.8.3

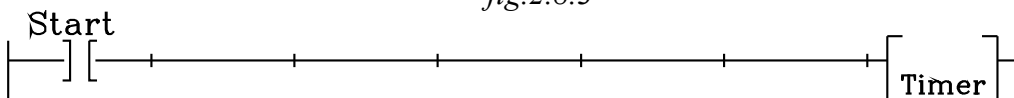
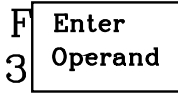
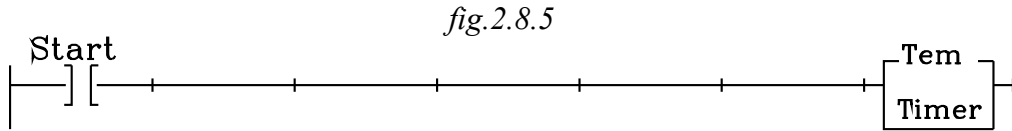


fig.2.8.4



- Pigiare il tasto funzione *F3* contenente il riquadro *Enter operand* per immettere l'espressione simbolica del temporizzatore.

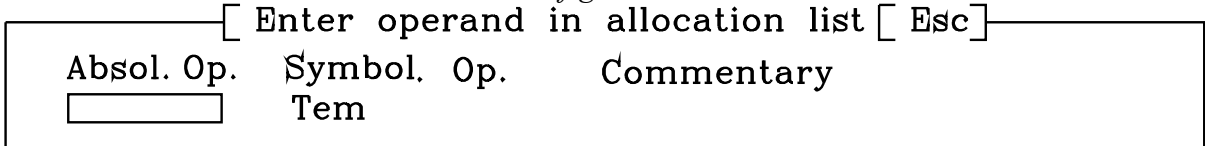
- Si evidenzia una striscia di scrittura ove porre il simbolo del temporizzatore: sia questi *Tem*, che verrà digitata in detta zona



- Scritto il simbolo *Tem* pigiare l'INVIO ↵

Si apre il solito riquadro di dialogo nel quale compare l'operatore simbolico *Tem* già introdotto e l'evidenziatore sotto l'operatore assoluto da immettere.

fig.2.8.6

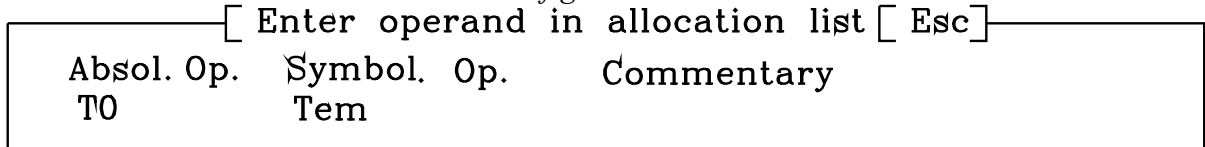


- Sia *T0* l'operatore assoluto indicante il temporizzatore: scriverlo nella zona evidenziata e pigiare l'INVIO ↵

L'evidenziatore si sposta sotto il commento.

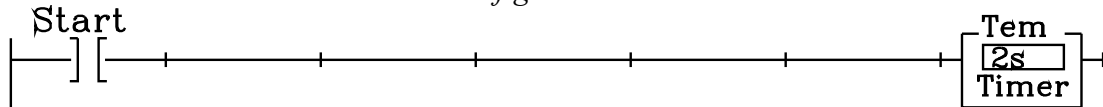
- Scrivere il commento: *Temporizzatore T0*

fig.2.8.7



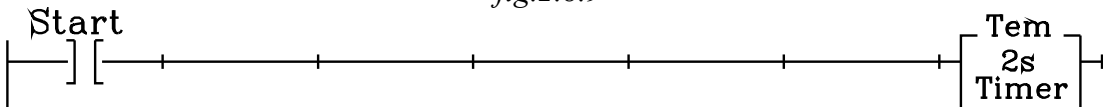
- Pigiare il tasto funzione *F1* per la conferma
- Si evidenzia nel centro un'altra striscia di digitazione ove scrivere in secondi "s" la durata di temporizzazione. Si voglia per esempio fissare la durata di 2 secondi: si scrive entro la zona di digitazione: *2s*

fig.2.8.8



- Pigiare l'INVIO ↵

fig.2.8.9





Nella riga di corrente digitata , il temporizzatore indicato con  $Tem$ , associato a  $T0$ , viene attivato dal contatto  $Start$  e rimane allo stato logico 1 ( $T0=1$ ) per 2 secondi.

Per fissare le idee riferiamoci ad una semplice applicazione pratica.

**ESEMPIO**

Si voglia effettuare il semplice ciclo  $A+ A-$  di un cilindro a doppio effetto, comandato da una valvola bistabile con un ritardo di 5 secondi nel rientro dello stelo: A- ritardato

Le equazioni logiche sono di semplice rilievo:

$$A^+ = Start \cdot a_0$$

$$T_0 = a_1 \quad T_0 = 2s$$

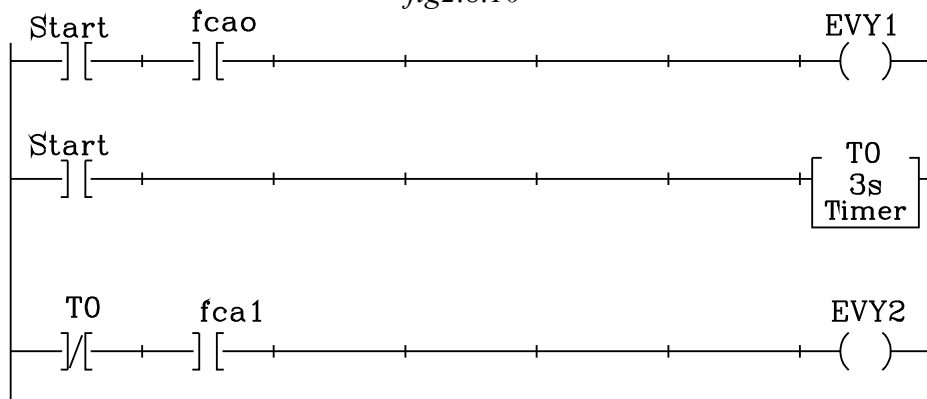
$$A^- = a_1 \cdot \overline{T_0}$$

La tabella delle occupazioni degli ingressi e uscite è la seguente.

TABELLA DELLE OCCUPAZIONI					
INGRESSI			USCITE		
Tipo	Simbolo	Operatore assoluto	Tipo	Simbolo	Operatore assoluto
Start	Start	I0	Elettrovalvola $Y_1$	EVY1	O0
finecorsa $a_0$	fcao	I1	Elettrovalv. Y2	EVY2	O1
finecorsa $a_1$	fca1	I2			

Dalle equazioni logiche si ricava facilmente il circuito a contatti.

fig.2.8.10

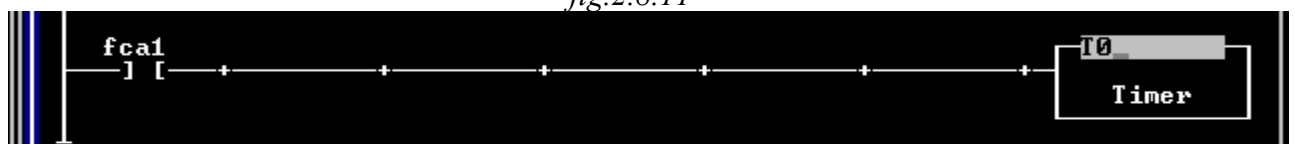


Occorre notare che come etichetta del temporizzatore è stato in questo caso adoperato l'indicazione dell'operatore assoluto  $T0$ .

In tal caso:

- Pigiato il tasto funzione  $F3$  per porre l'etichetta si è scritto direttamente l'operatore assoluto  $T0$ .

fig.2.8.11



- Pigiato *l'INVIO* si ottiene il solito riquadro di immissione degli operatori e del commento. In questo caso comparirà *T0* sotto l'operatore assoluto, mentre il cursore è sotto l'operatore simbolico.

fig.2.8.12



In questo caso si decida di indicare il temporizzatore con l'etichetta dell'operatore assoluto **T0** e di non associare ad esso alcun operatore simbolico e di non scrivere alcun commento.

- Senza scrivere altro, pigiare il tasto funzione *F1* di conferma

#### *Funzionamento dello schema a contatti*

*La elettrovalvola bistabile EVY1 viene eccitata dalla serie dello Start e del fincorsa  $a_0$  toccato.*

*Esce lo stelo del cilindro A, e, quando tocca il fincorsa  $a_1$ , allora viene attivato il temporizzatore T0.*

*Il temporizzatore T0 può essere interrogato come se fosse un contatto e può essere rappresentato con uno di questi NA o NC nella zone di Test.*

*Per il rientro del cilindro A devono passare 3 sec; perciò viene posto in serie al contatto di fincorsa  $a_1$  il contatto fittizio T0 negato, questi comandano l'elettrovalvola EVY2.*

*Così, toccato  $a_1$ , il temporizzatore T0 si pone al livello logico 1:  $T0=1$ .*

*Il negato di T0 ( $\overline{T_0}$ ), in serie ad  $a_1$  sulla linea di corrente, non permette l'attivazione dell'elettrovalvola Y2, fino a che, passati 3 secondi il temporizzatore T0 si pone allo stato logico 0:  $T=0$ .*

*Quando  $T0=0$ , risultando contemporaneamente  $\overline{T_0} = 1$  ed  $a_1 = 1$  perché toccato, allora viene attivata l'elettrovalvola EVY2 e lo stelo compie la corsa  $A^-$  di rientro.*

### **2.8.2 Contatori**

Con il PLC è possibile effettuare un conteggio di impulsi via software. Ogni impulso può essere interrogato, posto in ingresso, nella zona di test e rappresentato da un contatto NA o NC.

Il suo simbolo può rientrare nelle equazioni logiche di comando: può essere negato, posto in AND o in OR con altri contatti.

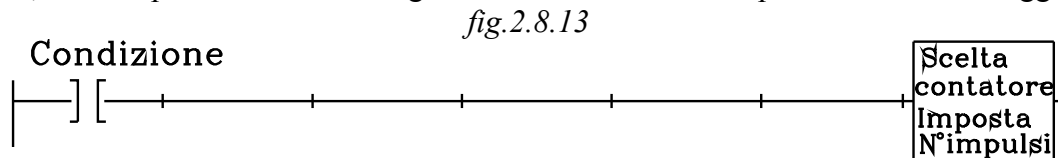
Nel PLC FESTO sono disponibili 16 contatori i cui operatori assoluti sono indicati con: C0, C1,..., C16

Il contatore viene impostato nella zona d'azione e la rappresentazione simbolica deve essere associata all'operatore assoluto

Se si adopera come indicazione del contatore direttamente l'operatore assoluto non occorre associare ad esso alcun'altra etichetta.

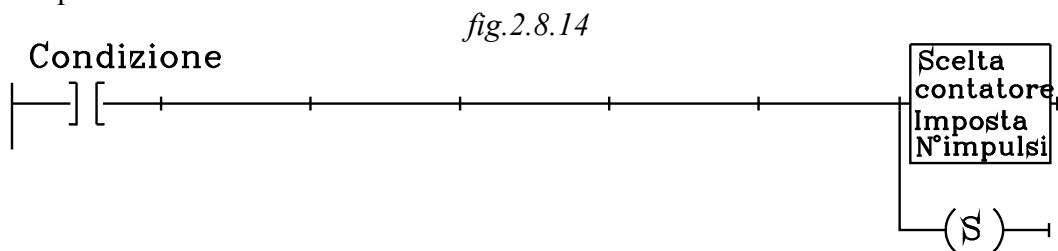
L'impostazione, l'attivazione e l'implementazione del conteggio si compongono di più fasi.

- La prima operazione è di scegliere il tipo di contatore tra i 16 e di impostare il valore del conteggio desiderato. Ciò avviene interrogando una determinata condizione, posta nella zona di test, e che si presenta come un ingresso in serie al Box di impostazione del conteggio.



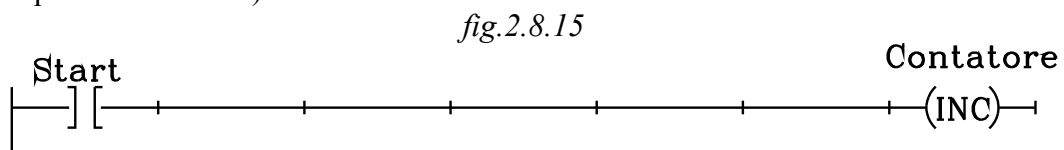
Tale condizione può essere un finecorsa, uno START, l'interrogazione di una uscita, un Flag...

- Una volta impostato il tipo di contatore, il valore del conteggio, occorre settare il contatore: tale segnale di Set può essere lo stesso che ha impostato il conteggio. In tal caso il Set si pone in parallelo al Box del contatore



Settato il contatore questo si pone nello stato logico 1:  $Contatore=1$

- A questo punto necessita un segnale che effettui l'incremento del conteggio. Questo si deve presentare come un contatto in serie all'uscita, nella quale viene posto il contatore di tipo *INC* (incremento) contrassegnato con l'etichetta del contatore prescelto (in forma simbolica o con l'operatore assoluto)



Per chiarire le idee e rendersi conto delle operazioni da effettuare, consideriamo il semplice esempio del ciclo  $A^+ A^-$ , nel quale, pigiato lo *Start*, si deve ripetere per 5 volte la fuoriuscita e il rientro dello stelo del cilindro *A*.

La scelta del tipo di contatore, l'impostazione del N° di impulsi e il suo settaggio viene effettuato dallo *START*.

Siano 5 il numero degli impulsi ed il contatore scelto sia *C0*.

$C0=START$   
 $N^\circ \text{ impulsi } =5$   
 $SET C0= START$

Riguardo agli ingressi ed uscite scegliamo le occupazioni già utilizzate nei precedenti cicli.

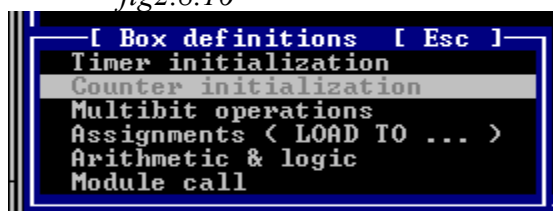
Aprire un nuovo File con nome, per esempio: *AACONTA*, seguendo la solita procedura già descritta:

- Selezionare dal menu principale *Project management*.
- Selezionare l'opzione *Create project*.
- Nel riquadro che si apre scrivere il nome del file: *AACONTA* e il commento " *A+ A-*" ripetuto 5 volte".
- Pigiare il tasto funzione *F1* per conferma.
- Dal menu principale selezionare *Leader diagram*.
- Selezionare l'opzione *Leader editor*.
- Scrivere nel riquadro sull'ultima riga il commento: " *A+ A-*" ripetuto 5 volte".
- Pigiare il tasto funzione *F1*.

Si entra nella finestra di editor dove compare la prima riga di corrente.

- Con la procedura già descritta, digitare nella zona di test il contatto normalmente aperto *Start*
- Portare il cursore nella zona di azione "*Ä( )Ä*".
- Per scegliere il tipo di contatore e impostare il N° di impulsi, pigiare il tasto funzione *F2* riferendosi al riquadro *Box*.

fig2.8.16

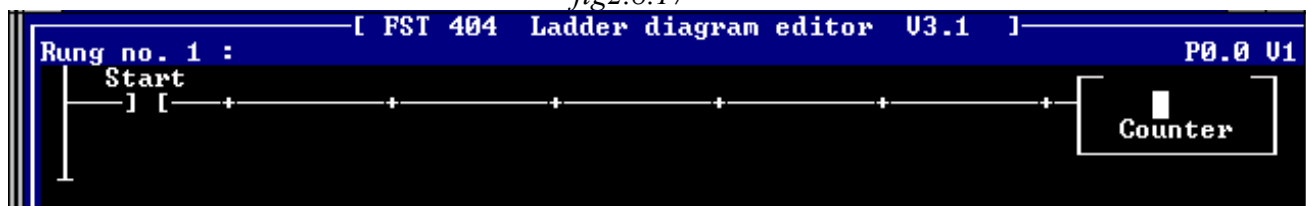


Si apre il riquadro di selezione.

- Selezionare *Counter initialization*.

Nella zona d'azione compare il Box di immissione.

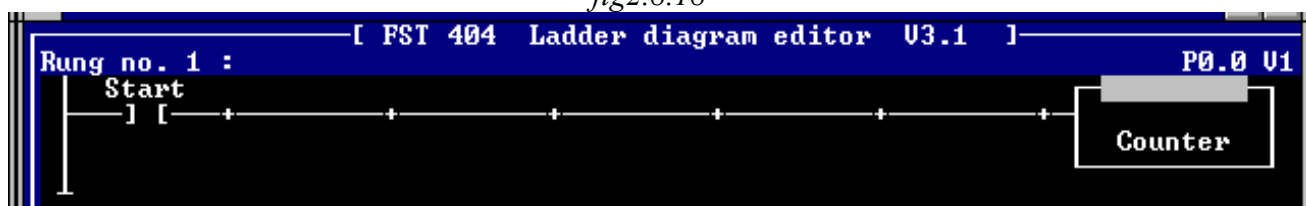
fig2.8.17



- Per scegliere il tipo di contatore e scrivere la sua etichetta pigiare il tasto funzione *F3* riferendosi al riquadro *Enter operand*.

Si evidenzia una zona di digitazione.

fig2.8.18



- Come etichetta scegliamo l'operatore assoluto del contatore *C0*. Scrivere *C0* e pigiare l'INVIO ↵

Si apre il solito riquadro per l'immissione delle etichette e il commento.

fig2.8.19



In questo caso si è immesso come etichetta l'operatore assoluto *C0* e questo compare sotto "Absol. Op."

Decidiamo di non associare all'operatore assoluto nessuna etichetta simbolica e né di scrivere alcun commento.

- Pigiare il tasto funzione *F1* riferentesi a *Enter*.

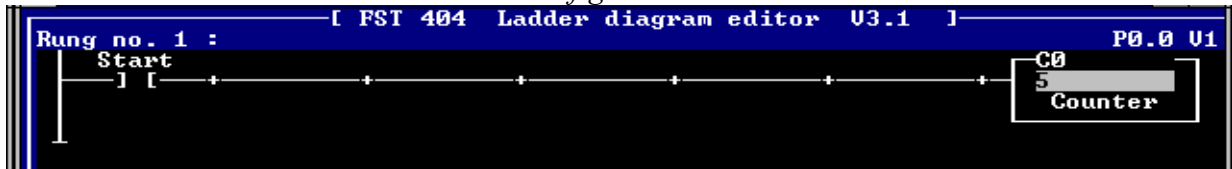
Pigiato *F1* compare al centro del Box la striscia evidenziata per l'immissione del N° di impulsi.

fig2.8.20



- Scrivere il numero di impulsi: ad esempio 5.

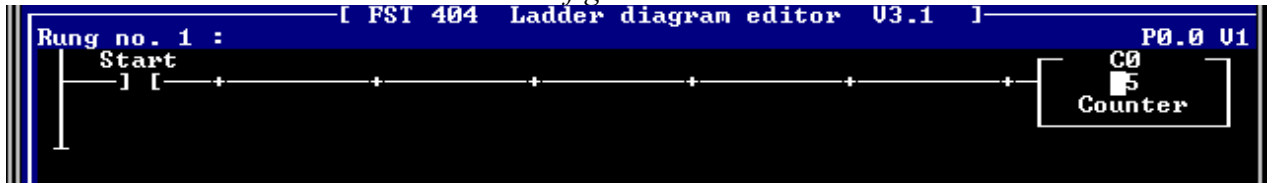
fig2.8.21



- Pigiare l'INVIO ↵

Si ottiene:

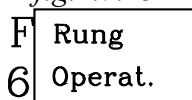
fig2.8.22



Con la procedura effettuata si è scelto ed inizializzato il contatore *C0* e, inoltre, si è impostato il n° di impulsi: 5.

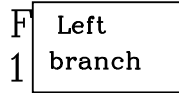
Occorre ora settare il contatore impostato. Questo deve essere attivato dallo stesso Start che l'ha inizializzato; quindi l'attivazione può essere effettuata da un *contattore di settaggio* posto su un braccio parallelo al Box di impostazione del contatore: simbolo "Ä( S )Ä".

fig.2.7.23



- Pigiare il tasto funzione *F6* riferentesi al riquadro *Rung Operat.*

fig.2.7.33



- Pigiare il tasto funzione *F1* riferentesi al riquadro *Parallel coil.*

Si ottiene il ramo parallelo al Box con il simbolo del contattore:  $\overline{A}(\quad)\overline{A}$

fig.2.8.23



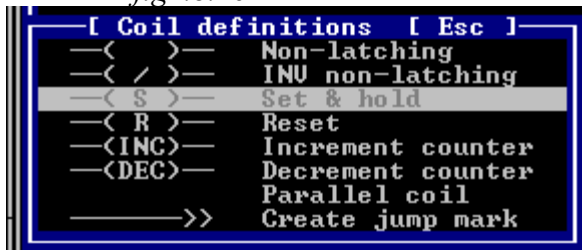
- Occorre tornare indietro nel menu precedente per porre l'etichetta al contattore: pigiare il tasto funzione *F8* (oppure Esc).
- Portare il cursore sul contattore " $\overline{A}(\quad)\overline{A}$ ", inserito in parallelo al box.

fig2.8.24



- Pigiare il tasto *F1* di selezione del contattore " $\overline{A}(\quad)\overline{A}$ "

fig2.8.26

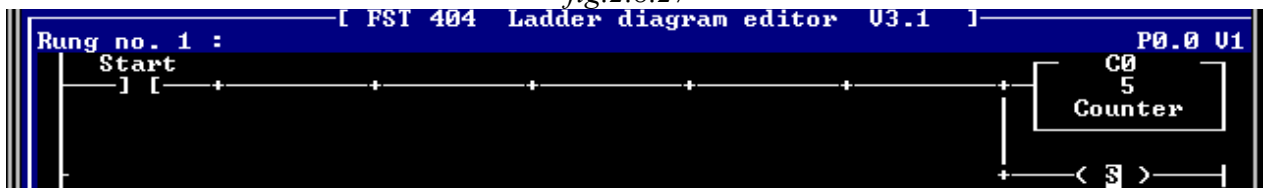


Si apre il riquadro di selezione di diversi tipi di contattori.

- Selezionare *Set & hold.*

Si ottiene:

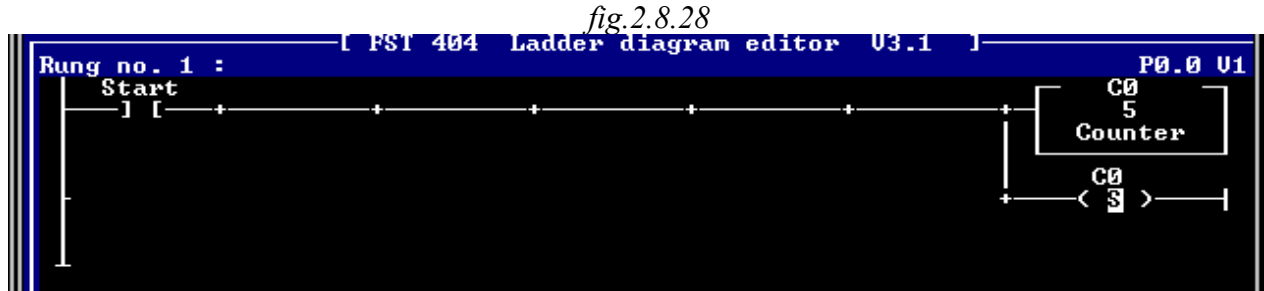
fig.2.8.27



Occorre ora associare al contattore "Ä( S )Ä" l'etichetta del contatore da settare: in questo caso "C0".

- Pigiare il tasto funzione F3.
- Scrivere C0.
- Pigiare INVIO ↵

Si ottiene:



Determiniamo le altre equazioni logiche.

L'elettrovalvola *EVY1* viene attivata dal finecorsa  $a_0$  in AND con il segnale emesso dal contatore *C0*:

$$EVY1 = a_0 \cdot C0$$

Effettuata la corsa  $A^+$ , viene toccato il finecorsa  $a_1$ . Questo viene posto come segnale che incrementa, fino al 5°, l'impulso del contatore; dopo di che esso si porta allo stato logico zero.

Il finecorsa  $a_1$ , inoltre determina il rientro dello stelo di *A*.

$$INC C0 = a_1$$

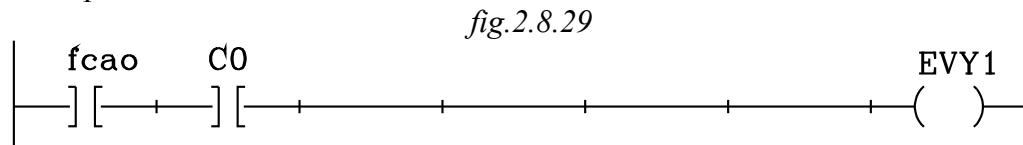
$$EVY2 = a_1$$

Le tre equazioni logiche si traducono nelle rispettive linee del circuito a contatti.

- Pigiando prima il tasto funzione F6 e successivamente nel sottomenu F3 si seleziona una nuova linea di corrente. Alla fine pigiare F8 per tornare al menu principale.

Equazione:  $EVY1 = a_0 \cdot C0$

Con la nota procedura si ottiene:



Occorre notare solamente che quando si introduce l'etichetta *C0* con il tasto funzione F3, non viene fatta dal programma alcuna altra richiesta: tutto è stato già definito nella riga di impostazione del contatore.

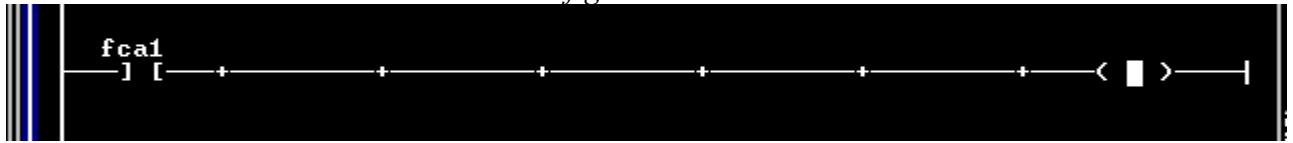
- Pigiando prima il tasto funzione F6 e successivamente nel sottomenu F3 si seleziona una nuova linea di corrente. Alla fine pigiare F8 per tornare la menu principale

Equazione  $INC C0 = a_1$

Seguiamo più in dettaglio la digitazione della linea di corrente che si riferisce all'incremento degli impulsi.

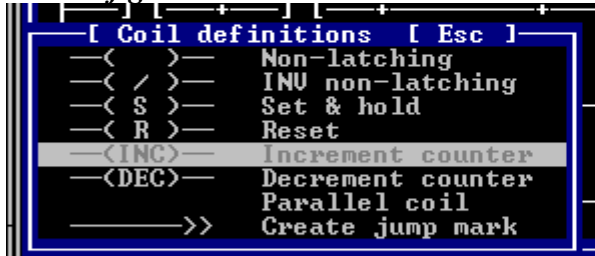
- Nella zona di test digitare, con la nota procedura, il contatto del finecorsa  $a_1$  avente come etichetta simbolica *fca1*.
- Portare poi il cursore nella zona d'azione "Ä( )Ä".

fig.2.8.30



- Selezionare il tipo di contattore pigiando il tasto funzione *F1*.

fig.2.8.31



Si apre il riquadro di selezione dei diversi tipi di contattori.

- Selezionare *Increment counter*.

Si ottiene:

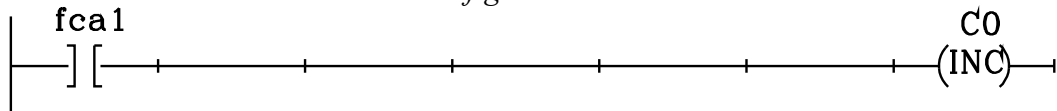
fig.2.8.32



- Selezionare con *F3* "Enter Operand" per introdurre l'operando.
- Scrivere *C0*.

Si ottiene:

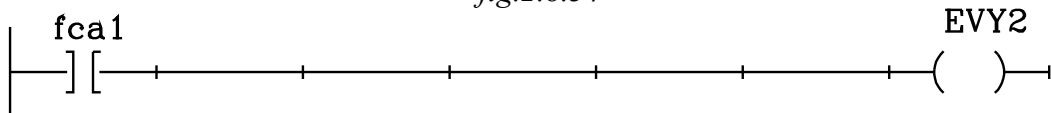
fig2.8.33



Equazione:  $EVY2 = a_1$

Con la nota procedura digitare la linea di corrente.

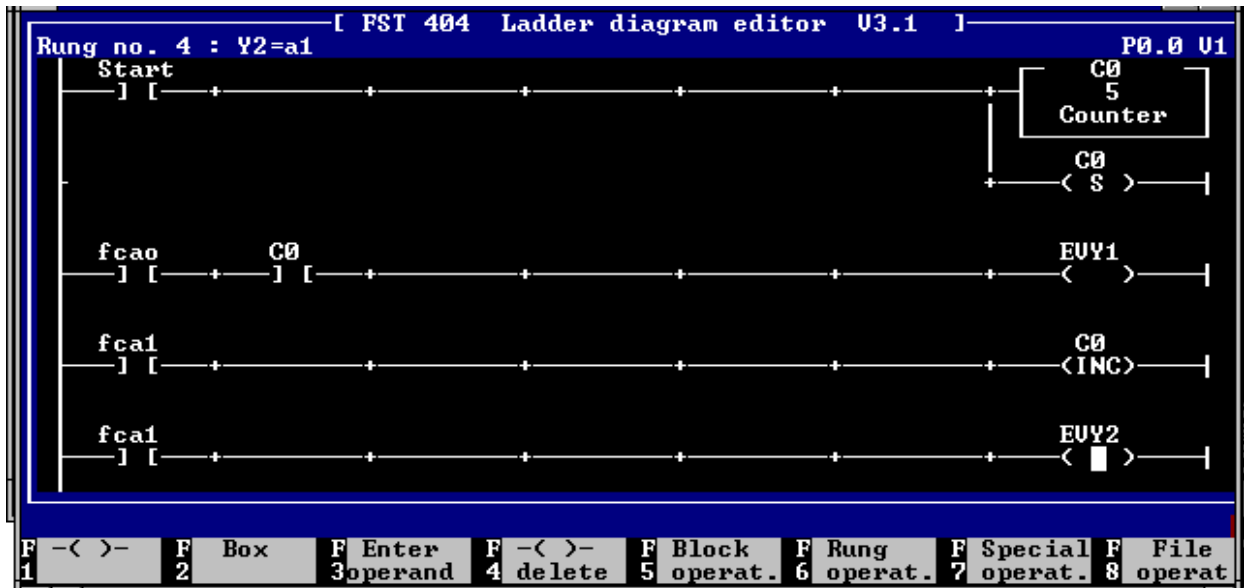
fig.2.8.34



Il programma completo è riportato in figura.

fig.2.8.35





### 2.8.3 PARTICOLARI APPLICAZIONI In laboratorio

*Circuito clock*

Si vuole effettuare un circuito che generi un'onda quadra della durata di  $\tau$  secondi.

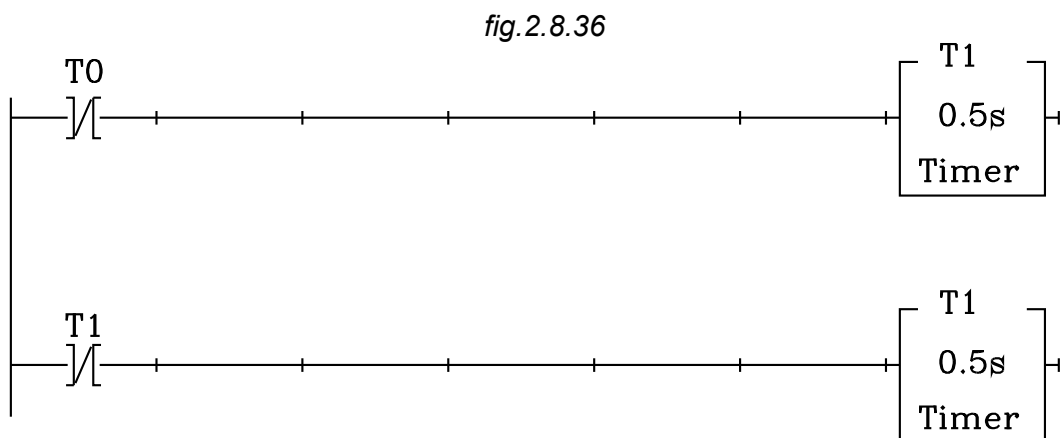
Ciò è possibile adoperando due temporizzatori, ciascuno dei quali viene eccitato dall'uscita negata dell'altro.

$$T1 = \overline{T0}$$

$$T0 = \overline{T1}$$

Supponiamo che l'onda quadra debba avere una durata 0.5 s

Le linee di corrente dei due temporizzatori sono:



Nelle condizioni iniziali risulta:

$$T0 = T1 = 0$$

di conseguenza si ha che:

$$\overline{T1} = \overline{T2} = 1$$

Nelle condizioni iniziali uno dei due temporizzatori verrà settato per primo: supponiamo che sia T1. Allora:

$$T1 = 1 \text{ per } 0.5 \text{ sec}$$

mentre, in tale tempo, il temporizzatore T0 non è settato, essendo questo comandato dalla condizione logica  $\overline{T1}$ :

$$\overline{T1} = \overline{1} = 0$$

Dopo 0.5 secondi il temporizzatore T1 va allo stato logico 0

$$T1 = 0$$

per cui viene attivato il temporizzatore T0 comandato dallo stato logico  $\overline{T1}$

$$\overline{T1} = \overline{0} = 1$$

Ora attivato il temporizzatore T0, lo stato logico  $\overline{T0} = \overline{1} = 0$  disattiva il temporizzatore T1, fino a che, trascorsi 0.5 sec cade T0 allo stato logico 0 e così via...

#### Esercizio

Si voglia con il clock far lampeggiare una lampada per un fissato numero n di volte. L'intervallo del lampeggio sia di 0.5 sec. ed n=5.

Occorre adoperare un contatore, che viene impostato ad un valore 5 e settato da uno Start

$$C0 = \text{Start}$$

$$N^{\circ} \text{ impulsi} = 5$$

$$\text{Set } C0 = \text{Start}$$

L'incremento al contatore può essere dato o da T1 o da T0: scegliamo T0

$$\text{INC } C0 = T0$$

La lampada viene attivata da T1 in AND con C0: quindi lampeggerà per n=5 volte

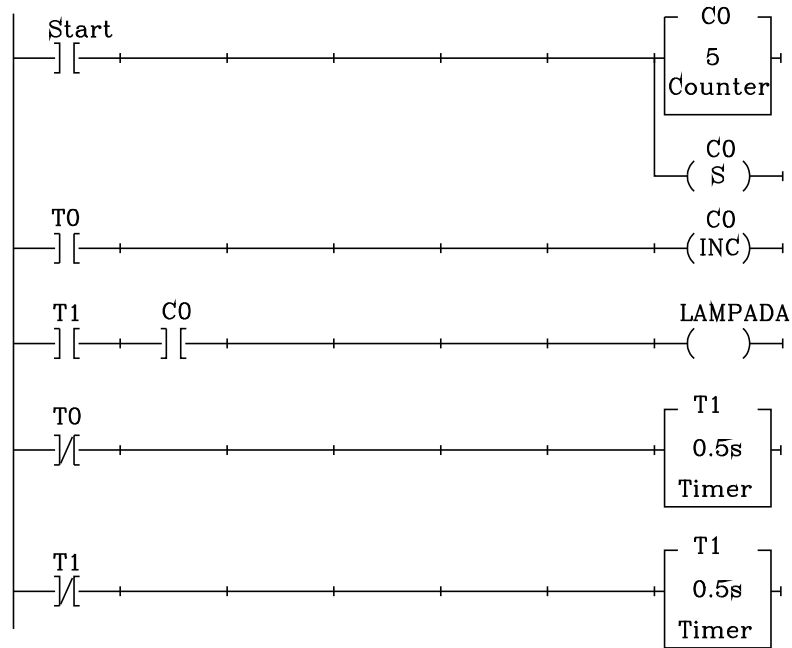
$$\text{Lampada} = T1 \cdot C0$$

Le equazioni logiche del clock sono:

$$T1 = \overline{T0}$$

$$T0 = \overline{T1}$$

fig.2.8.37.



**USO DEL GRAFCET NEL PLC**

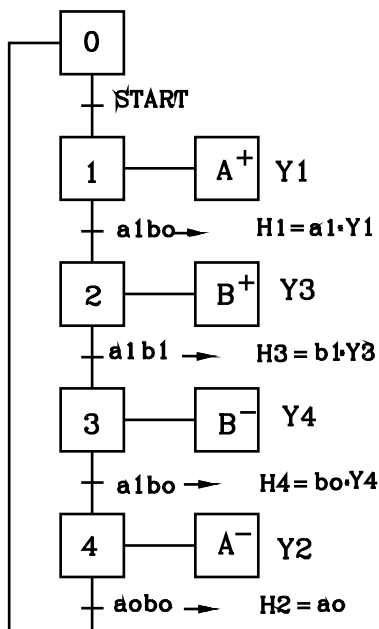
Il grafcet trova una appropriata applicazione nel PLC.

Nella zona di test possono essere interrogate anche le uscite, i cui segnali danno dei condizionali utilizzabili.

In tal modo si può formare una memorie con precedenza alla cancellazione, che comanda direttamente l'uscita e il cui segnale di autoritenuta è dato da quello Y fornito dall'uscita stessa.

$$Y = (Set + Y) \cdot \overline{Reset}$$

fig.2.8.38



In tal modo i circuiti effettuati nell'elettropneumatica mediante l'uso dei moduli del sequenziatore vengono modificati, eliminando le bobine ausiliarie K1,K2,K3... e i loro segnali.

Consideriamo così il ciclo  $A^+ B^+ B^- A^-$ , più volte analizzato, e si debba realizzare con il PLC nel linguaggio Ladder, utilizzando il grafcet per la soluzione.

fig.2.8.39

Vicino al blocco di comando, si è posto l'elettrovalvola corrispondente.

Se si vuole eliminare la necessità di porre il segnale di settaggio dell'ultimo modulo e resettaggio di tutti gli altri, occorre modificare l'ultimo comando di transizione H2.

Precedentemente, nell'elettropneumatica, si è trasformata a logica del sequenziatore pneumatico in quella di un circuito elettropneumatico.

Per funzionare occorre che, nelle condizioni iniziali venga settato l'ultimo modulo, e siano resettati tutti i moduli precedenti.

Settato l'ultimo modulo allora la bobina ausiliaria H2 è eccitata e può fornire il segnale di comando in AND con lo Start per iniziare il ciclo.

Si può modificare H2 facendolo dipendere solamente dalla condizione dei finecorsa e non dall'ultimo comando Y2. Ciò è possibile, in quanto non vi è, in precedenza, nessuna transizione di comando dei moduli che ha la stessa combinazione  $a_0 \cdot b_0$  di fine ciclo.

In tal caso essendo all'inizio del ciclo (come alla fine) i due finecorsa  $a_0, b_0$  attivati, esso può iniziare senza bisogno di un segnale di *preset* dell'ultimo modulo

È evidente che se l'ultima transizione avesse una combinazione dei finecorsa che si fosse già presentata precedentemente nel ciclo, in tal caso, occorrerebbe farla dipendere anche dal segnale di attivazione dell'ultimo modulo: si dovrebbe aggiungere un segnale di *preset* che lo attivi all'inizio del ciclo.

Adoperando solamente le variabili attive, l'ultima transizione è data dal finecorsa  $a_0$ . Questa è posta in AND con lo Start per formare la transizione che comanda l'inizio del ciclo:

$$H2 = a_0 \cdot Start$$

TABELLA DELLE OCCUPAZIONI					
INGRESSI			USCITE		
Tipo	Simbolo	Operatore assoluto	Tipo	Simbolo	Operatore assoluto
Start	Start	I0	Elettrovalvola Y <sub>1</sub>	EVY1	O0
finecorsa $a_0$	fcao	I1	Elettrovalv. Y2	EVY2	O1
finecorsa $a_1$	fca1	I2	Elettrovalv. Y3	EVY3	O2
finecorsa $b_0$	fcbo	I3	Elettrovalv. Y4	EVY4	O3
finecorsa $b_1$	fc b1	I4	Flah H1	FLH1	F0.0
			Flah H2	FLH2	F0.1
			Flah H3	FLH3	F0.2
			Flah H4	FLH4	F0.3

Si possono stendere le equazioni già note del sequenziatore, omettendo i segnali di *preset* dell'ultimo modulo e di *Set* di tutti gli altri

$$H2 = a_0 \cdot Start$$

$$Y1 = (Start \cdot H2 + Y1) \cdot \overline{Y3}$$

$$H1 = a_1 \cdot Y1$$

$$Y3 = (H1 + Y3) \cdot \overline{Y4}$$

$$H3 = b_1 \cdot Y3$$

Transizione fine ciclo in AND con Start

Comando A<sup>+</sup>

Transizione attuato stato A<sup>+</sup>

Comando B<sup>+</sup>

Transizione attuato stato B<sup>+</sup>

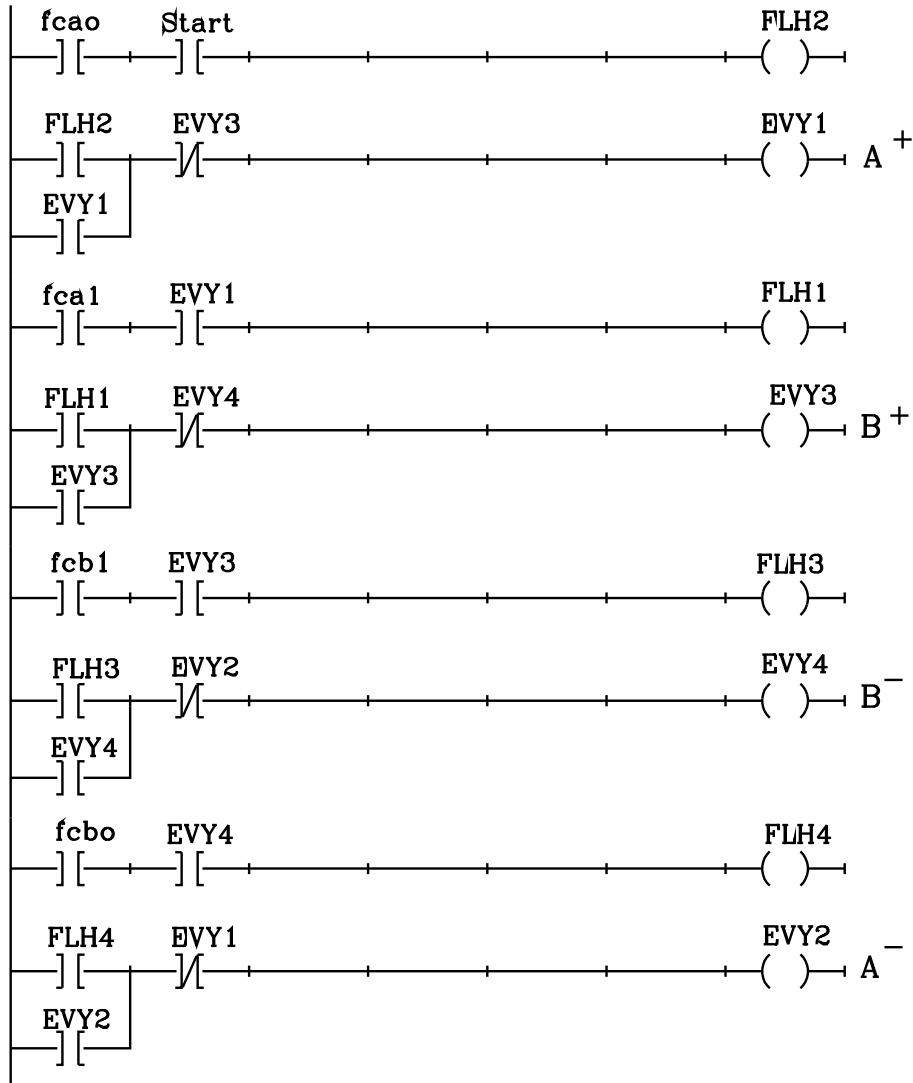
$$Y4 = (H3 + Y4) \cdot \overline{Y2}$$

$$H4 = b_0 \cdot Y4$$

$$Y2 = (H4 + Y2) \cdot \overline{Y1}$$

Comando B<sup>-</sup>Transizione attuato stato B<sup>-</sup>Comando A<sup>-</sup>

fig.2.8.40



Occorre notare nel PLC FEST 404 della FESTO le etichette debbono contenere non meno di 3 caratteri, così ai simboli dei segnali di transizione H1, H2, H3, H4 sono stati aggiunti i due caratteri FL. Quindi H1 → FLH1, H2 → FLH2 ecc.

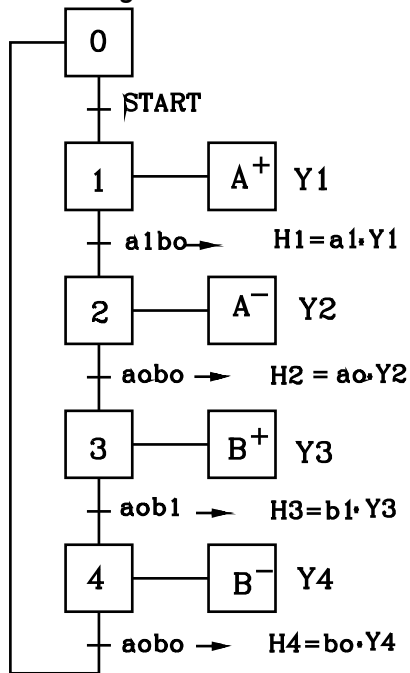
Si consideri ora un ciclo, nel quale la combinazione dei finecorsa di fine ciclo risulti uguale ad un'altra che comanda una fase precedente.

In tal caso occorre che anche l'ultimo segnale di transizione del finecorsa, che comanda la prima fase del ciclo, deve essere posto in AND con il segnale dell'elettrovalvola dell'ultimo modulo.

CICLO  $A^+ A^- B^+ B^-$ 

## GRAFCECT

fig.2.8.41



Si debbano effettuare più cicli consecutivi lasciando pigiato lo Start.

In questo caso non è possibile semplificare il circuito eliminando nell'ultimo segnale di transazione  $L'AND$  con quello dell'elettrovalvola dell'ultimo modulo  $Y4$ , in quanto la combinazione  $a_0 \cdot b_0$  dei finecorsa che si ha alla fine del ciclo si è già verificata nella transizione che comanda  $B^+$  e il ciclo si blocca

Infatti nelle due transizioni sono attivati entrambi i finecorsa  $a_0, b_0$  e quindi, rimanendo pigiato lo  $START$ , nel ciclo vi sono due segnali di comando che contemporaneamente vengono attivi:  $b_0 \cdot Start$  al rientro dello stelo di  $B$  che comanda un nuovo ciclo e  $a_0 \cdot Y_2$  al rientro dello stelo di  $A$  che comanda la fuoriuscita di  $B$ .

Per cui, rimanendo pigiato lo  $START$ , quando rientra il cilindro  $B$ , allora fuoriesce  $A$  ( $b_0 \cdot Start$  è attivato), ma fuoriesce anche  $B$  ( $a_0 \cdot Y_2$  è attivato) e il ciclo si blocca.

Occorre differenziare il segnale di eccitazione di  $Y1$  ponendo in  $AND$  alla combinazione " $b_0 \cdot Start$ " il segnale  $Y4$ . In tal caso, come si è detto nei sequenzatori occorre aggiungere il segnale di  $PRESET$  da porre in  $AND$  negato nelle linee di corrente che comandano l'elettrovalvole salvo l'ultima nella quale detto segnale viene posto in  $OR$ .

Le equazioni logiche sono:

$$\begin{aligned}
 PR &= IN \\
 H4 &= b_0 \cdot Y4 \cdot Start \\
 Y1 &= (H4 + Y1) \cdot \overline{Y2} \cdot PR \\
 H1 &= a_1 \cdot Y1 \\
 Y2 &= (H1 + Y2) \cdot \overline{Y3} \cdot PR \\
 H2 &= a_0 \cdot Y2 \\
 Y3 &= (H2 + Y3) \cdot \overline{Y4} \cdot PR \\
 H3 &= b_1 \cdot Y3 \\
 Y4 &= (H3 + Y4 + PR) \cdot \overline{Y1}
 \end{aligned}$$

fig.2.8.42

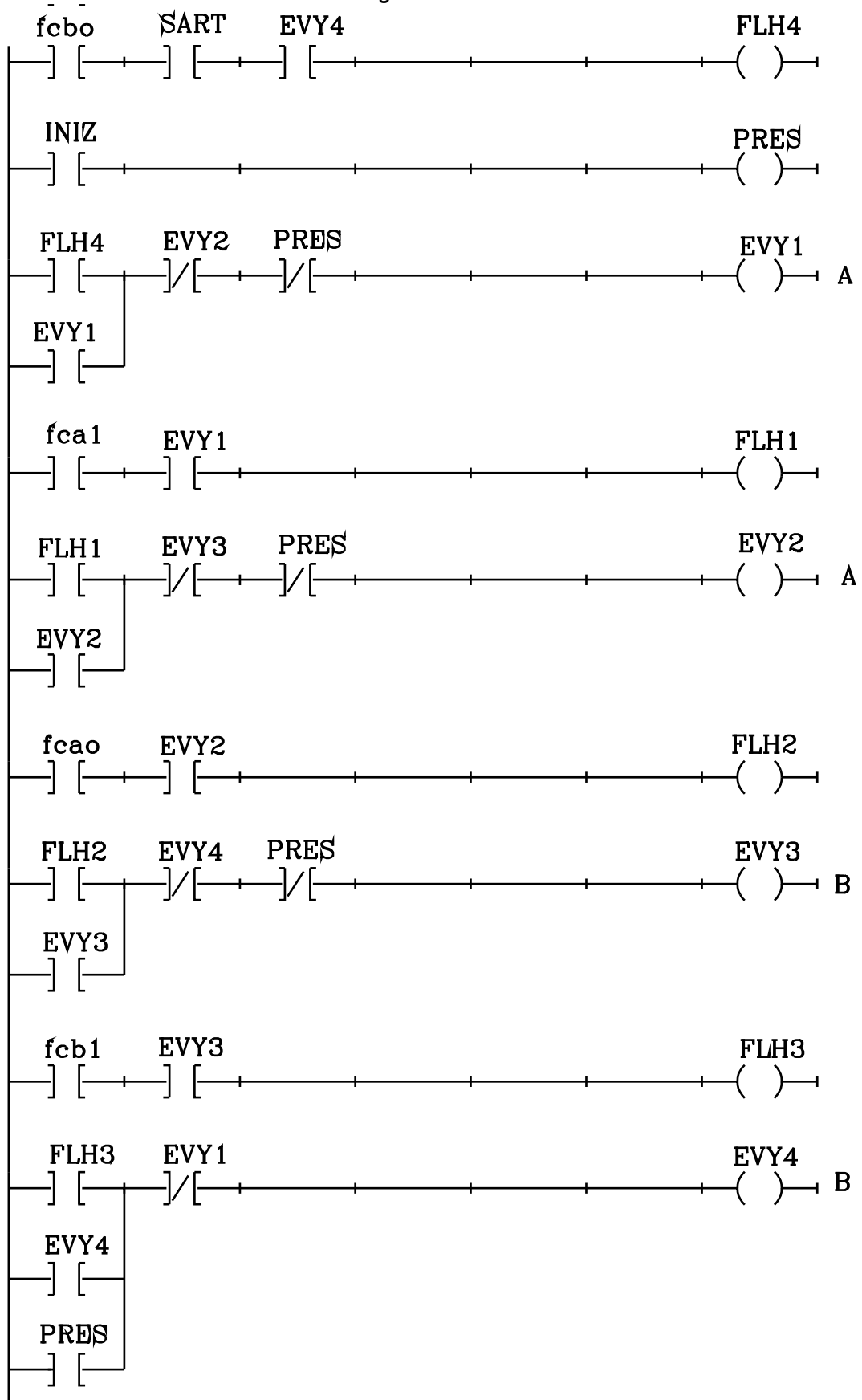


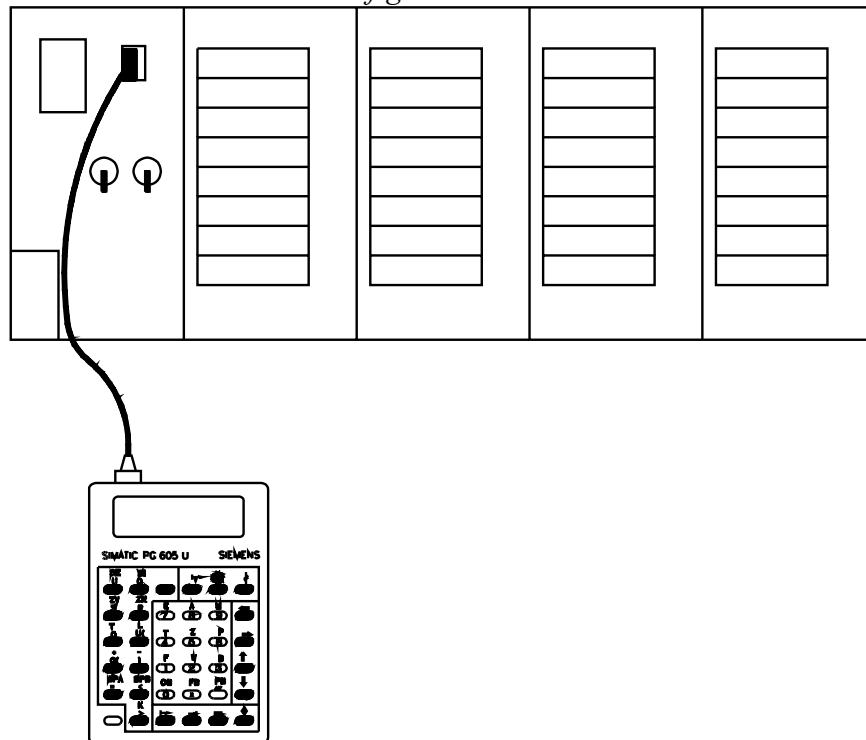
TABELLA DELLE OCCUPAZIONI					
INGRESSI			USCITE		
Tipo	Simbolo	Operatore assoluto	Tipo	Simbolo	Operatore assoluto
Pulsante	INIZ	I5	Flag	PRES	F0.4
Start	Start	I0	Elettrovalvola Y <sub>1</sub>	EVY1	O0
finecorsa a <sub>0</sub>	fcao	I1	Elettrovalv. Y2	EVY2	O1
finecorsa a <sub>1</sub>	fca1	I2	Elettrovalv. Y3	EVY3	O2
finecorsa b <sub>0</sub>	fcbo	I3	Elettrovalv. Y4	EVY4	O3
finecorsa b <sub>1</sub>	fcbl	I4	Flah H1	FLH1	F0.0
			Flah H2	FLH2	F0.1
			Flah H3	FLH3	F0.2
			Flah H4	FLH4	F0.3



## 2.10 PLC CON TASTIERA PER LISTA DI ISTRUZIONI

Vi sono in commercio diversi PLC portatili con tastiera dedicata alla digitazione di programmi in lista di istruzioni.

fig.2.9.33



Questi PLC sono maneggevoli e di costo contenuto; di contro, rispetto ai sistemi forniti di software con CAD per la rappresentazione di contatti, detti PLC presentano l'inconveniente di dover tradurre lo schema a contatti o funzionale in lista di istruzioni da digitare sulla tastiera.

Ciò comporta che il tecnico non deve solamente rivolgere l'attenzione sulla soluzione del problema di automazione da risolvere, ma anche nella stesura della lista di istruzioni, che presuppone la conoscenza del particolare linguaggio di programmazione, diverso a seconda della casa costruttrice.

Se dal punto di vista industriale la stesura del programma in lista di istruzioni rappresenta un certo aggravio, dal punto di vista didattico riveste invece un particolare interesse, in quanto le istruzioni digitate fanno capire la logica del PLC.

È evidente che con un software comprendente un CAD con il quale vengono visualizzati i contatti o i blocchi funzionali con scelta da un menu, l'attenzione è tutta rivolta alla stesura degli schemi, disinteressandosi completamente dalle operazioni logiche che deve effettuare il PLC.

In questo caso occorre acquisire una certa pratica nella selezione delle opzioni del menu. Ciò risulta molto agevole e di facile comprensione.

Proprio per avvicinarsi alla logica di operazioni che il PLC deve effettuare viene qui di seguito brevemente descritto un sistema di programmazione in lista di istruzioni.

Si fa riferimento al *PLC SIMATIC 605 U* della *SIEMENS*. Apparecchio particolarmente affidabile di costo contenuto, destinato a settori bassi e medi. Il PLC può essere interfacciato con PC per i linguaggi di programmazione in schema a contatti (*KOP*) e in schema logico (*FUP*).

I PLC della *SIEMENS* trovano largo impiego nelle industrie.

### 2.10.1 Moduli di ingresso e di uscita

Come tutti i PLC vi sono moduli di ingresso e di uscita. I moduli di ingresso vengono contrassegnati con la lettera "E", quelli di uscita con la lettera "A".

I moduli sono indicati da numeri progressivi, partendo dal principale contrassegnato con il numero "0".

Ogni modulo contiene 8 canali contrassegnati da numeri progressivi da 0 fino a 7.

Come già si è fatto per i precedenti PLC spiegati, per contraddistinguere un ingresso si scrive prima la lettera E, seguita dal numero indicante il modulo e poi quello del canale (il particolare ingresso di quel modulo).

Così:

*E0.2*

rappresenta il canale 2 del modulo 0 di ingresso (E) .

Alla stessa maniera per indicare un canale di uscita si scrive in ordine: - la lettera A - Il numero indicante il modulo - il numero indicante il canale di quel modulo.

Così:

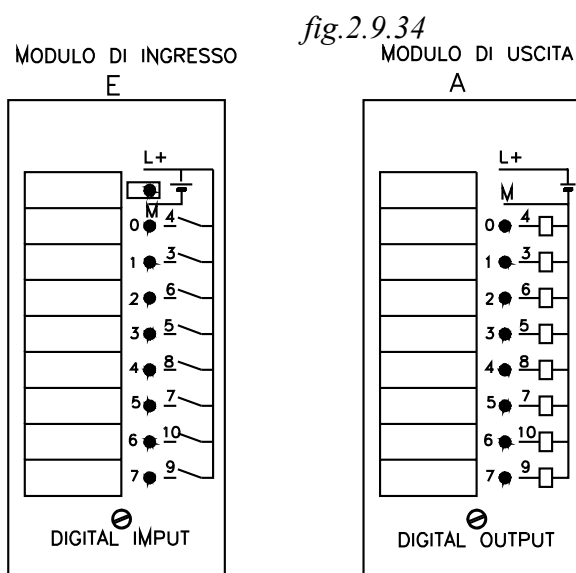
*A1.3*

rappresenta il canale 3 del modulo 1 di uscita (A).

Ora ciascuna canale è associato con una coppia di morsetti di una morsettiera sulla quale vengono collegati i conduttori recanti i segnali di ingresso, provenienti dai sensori, finecorsa, Start ecc. Segnali che provengono dal processo controllato.

La morsettiera contiene 10 morsetti sui quali possono essere collegati dei conduttori.

La rispondenza tra il numero che contraddistingue il canale e il numero del morsetto corrispondente è riportato sulla piastra frontale del modulo utilizzato.



Come riportato schematicamente in *fig.9.34*, sul frontale del modulo di ingresso, in corrispondenza dei canali di ingresso, sono riportati i numeri dei morsetti a quali sono associati.

Anche sul frontale del modulo di uscita, in corrispondenza dei canali di uscita, sono riportati i numeri dei morsetti a quali sono associati.

Come si può constatare si hanno le seguenti associazione:

Associazione can. ingresso - morsetto		Associazione can. uscita - morsetto	
Canale	Morsetto	Canale	Morsetto
0	4	0	4
1	3	1	3
2	6	2	6
3	5	3	5
4	8	4	8
5	7	5	7
6	10	6	10
7	9	7	9

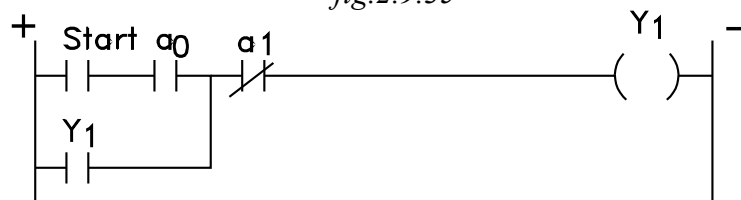
### 10.1.1 Allacciamento degli ingressi alla morsettiera

Nella morsettiera vi sono 10 morsetti, 8 dei quali sono associati ai canali di ingresso come riportato nella *tabella 10.1*. Il morsetto 2 è a massa mentre il morsetto 1 è collegato al potenziale positivo.

Un finecorsa (un sensore...) posto sul processo da controllare va collegato tra il conduttore comune posto sul morsetto 1 e il morsetto associato al canale al quale detto finecorsa invia il comando.

Consideriamo come esempio lo schema a contatti di *fig.9.35*, riferentesi al ciclo  $A^+$ ,  $A^-$  di comando di un cilindro pneumatico con l'impiego di una elettrovalvola monostabile.

*fig.2.9.35*



Gli ingressi sono dati dallo *Start* e dai finecorsa  $a_0$ ,  $a_1$ .

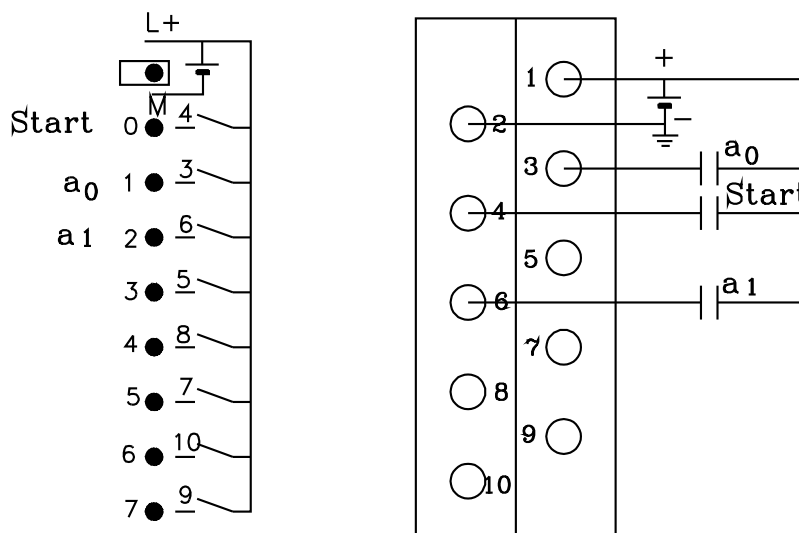
Si scelgano le seguenti associazioni tra i segnali di ingresso, i canali e i morsetti ad essi associati:

Elemento	Simbolo	Canale	Morsetto
Start	<i>Start</i>	<i>E0.0</i>	4
Finecorsa	$a_0$	<i>E0.1</i>	3
Finecorsa	$a_1$	<i>E0.2</i>	6

È stato utilizzato il modulo 0.

Nella figura *fig.9.36* sono scritti i finecorsa e lo *Start* accanto ai canali. Così in corrispondenza del canale 0 è scritto lo *Start*, vicino al canale 1 il finecorsa  $a_0$  e accanto al canale 2 il finecorsa  $a_1$ .

*fig.2.9.36*



Considerando le associazioni dei canali con i morsetti si ha:

- Lo *Start* che comanda il canale  $E0.0$  associato al morsetto 4, viene collegato tra questo e il comune 1.
- Il finecorsa  $a_0$ , che comanda il canale  $E0.1$  associato al morsetto 3, viene collegato tra questo e il comune 1.
- Il finecorsa  $a_1$ , che comanda il canale  $E0.1$  associato al morsetto 6, viene collegato tra questo e il comune 1.

### 10.1.2 Allacciamento delle uscite alla morsettiera

Nella morsettiera vi sono 10 morsetti, 8 dei quali sono associati ai canali di uscita come riportato nella *tabella 10.1*. Il morsetto 2 è a massa mentre il morsetto 1 è collegato al potenziale positivo. Il morsetto 1 è collegato al potenziale positivo, mentre il morsetto 2, collegato al negativo, è a massa.

Una bobina, che eccitata comanda un attuatore, va collegata tra il conduttore comune, posto sul morsetto 2, e il morsetto associato al canale dal quale riceve il comando.

Il comando dell'attuatore può essere diretto se l'assorbimento di corrente è compatibile con le uscite del PLC, altrimenti detto comando va impartito ad una bobina di un relè con basso assorbimento che chiude un contatto normalmente aperto, ponendo così in comunicazione l'attuatore con una sorgente di alimentazione di elevata potenza.

Nel proseguo della trattazione si suppone di lavorare in quest'ultima ipotesi.

Per meglio comprendere quanto detto ci si riferisca ad un semplice esempio.

Il circuito di comando sia rappresentato da quello a contatti di figura *fig.2.9.37*, riferentesi al ciclo  $A^+$ ,  $A^-$  di comando di un cilindro pneumatico con l'impiego di una elettrovalvola bistabile.

*fig.2.9.37*

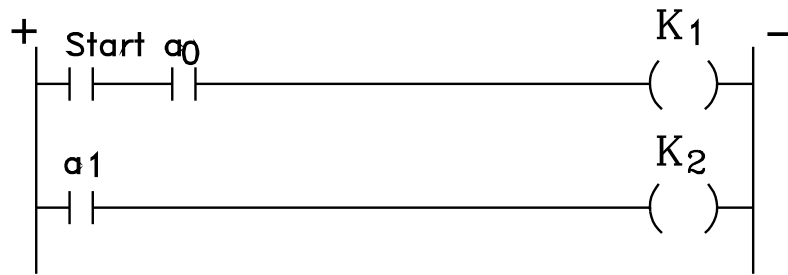
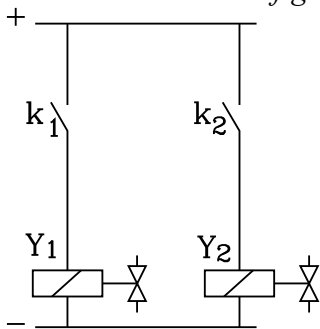


fig.2.9.38



Si suppone che le uscite non comandino direttamente le bobine  $Y_1, Y_2$  dell'elettrovalvola bistabile, ma due bobine ausiliarie  $K_1, K_2$  che eccitano chiudano i contatti normalmente aperti posti rispettivamente in serie alla alimentazione delle  $Y_1, Y_2$ , come rappresentato in fig.2.8.38.

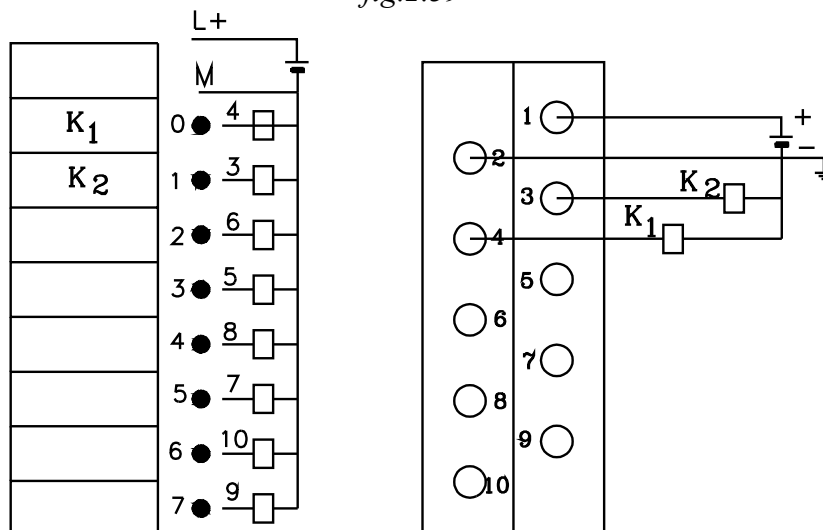
Si scelgano le seguenti associazioni tra i segnali di uscita, i canali e i morsetti ad essi associati:

Elemento	Simbolo	Canale	Morsetto
Bobina	$K_1$	A2.0	4
Bobina	$K_2$	A2.1	3

È stato utilizzato il modulo 2.

Nella figura fig.9.39 sono scritte le bobine ausiliarie accanto ai canali. Così in corrispondenza del canale 0 è scritta l'indicazione della bobina ausiliaria  $K_1$ , vicino al canale 1 l'indicazione della bobina ausiliaria  $K_2$ ,

fig.2.39



Considerando le associazioni dei canali con i morsetti si ha:

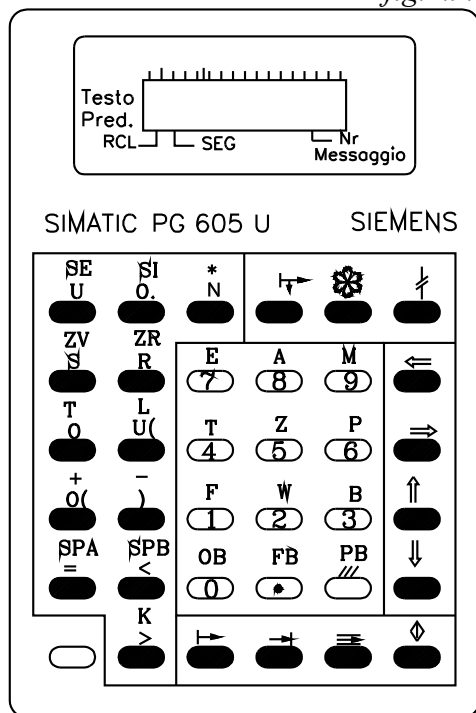
- La bobina ausiliaria  $K_1$  comandata dal canale A2.0 associato al morsetto 4, viene collegata tra questo e il comune 2.

- La bobina ausiliaria  $K_2$  comandata dal canale  $A2.10$  associato al morsetto 3, viene collegata tra questo e il comune 2.

## 10.2 Stesura del programma

La serie *SIMATIC* prevede diversi tipi di linguaggi, quello che qui viene considerato è la rappresentazione in lista di istruzioni (*AWL*).

fig.2.9.40



Nella figura è riportata la tastiera con le indicazioni delle funzioni e dei dati in prossimità dei tasti.

L'immissione delle istruzioni si ottiene pigiando i tasti contenenti le indicazioni delle funzioni e dati che si vogliono porre nel programma.

Alcuni tasti hanno una doppia indicazione, corrispondenti a funzioni diverse. La seconda indicazione in alto, stampigliata in bianco (*compare in nero nella figura schematica fig.2.9.40*), viene attivata dal tasto bianco, posto nell'estremità in basso a sinistra della tastiera priva di indicazioni: *tasto di predisposizione*.

Così per selezionare la funzione *PB* occorre pigiare prima il tasto di predisposizione e poi quello con l'indicazione *PB*.

Le funzioni indicate in nero vengono immesse semplicemente pigiando il tasto corrispondente.

### 10.2.1 Composizione di una istruzione

L'argomento è stato già trattato nella parte generale.

Ricordiamo che una istruzione è composta di tre parti essenziali: *Indirizzo - Codice operativo - Operando*.

**Indirizzo** È un numero naturale che va da 0 fino a 255. Esso rappresenta l'indirizzo della istruzione nella memoria di programma.

**Codice operativo** Rappresenta la operazione che deve essere eseguita sugli operandi.

Le operazioni logiche sono indicate con particolari caratteri:

Ad esempio l'operazione *AND* è indicata con *U*, l'operazione *OR* con *O*, ecc. (vedi oltre).

**Operando** È la parte dell'istruzione che contiene l'informazione sul dato su cui deve essere eseguita la operazione.

L'operando a sua volta è contraddistinto da due elementi fondamentali:

**Tipo** Contraddistingue il tipo di operando contraddistinto da una sigla.

Esso indica se si tratta:

Di un ingresso: Simbolo *E*

Di una uscita Simbolo *A*

Ecc.

*Elemento* Vi sono più elementi dello stesso tipo. Così gli ingressi sono composti da più moduli e ciascuno di essi è costituito da un certo numero di elementi.

L'elemento specifico può essere indicato da due numeri separati da un punto, che rappresentano, rispettivamente, il modulo e l'ingresso utilizzato (*canale*). Esempio:

*E2.4*

Rappresenta il canale 4 del modulo di ingresso 2.

$$Istruzione \left\{ \begin{array}{l} \text{Codice Operativo} \\ \text{Operando} \left\{ \begin{array}{l} \text{Tipo} \\ \text{Elemento} \end{array} \right. \end{array} \right.$$

Si consideri così l'istruzione:

*U E2.4*

Dove *U* indica l'operazione *AND*

Essa è composta dalle seguenti parti

$$Istruzione \ U \ E2.4 \left\{ \begin{array}{l} \text{Codice Operativo:} \quad \text{Operazione } AND \ (U) \\ \text{Operando} \left\{ \begin{array}{l} \text{Tipo:} \quad \text{Ingresso } E \\ \text{Elemento} \quad \text{Modulo } 2 - \text{Canale } 4 \end{array} \right. \end{array} \right.$$

### 10.3 Il linguaggio AWL della SIMAC S5 605U

Qui di seguito si riportano i principali codici degli operandi del linguaggio di programmazione STEP 5.

*E* Codice degli ingressi - Interfaccia tra il processo esterno ed il PLC.

*A* Codice delle uscite - Interfaccia tra il PLC ed il processo esterno.

*M* Merker (Flag) - Sono le memorie intermedie; corrispondenti alle bobine ausiliarie dei circuiti cablati.

*T* Memorie per l'ottenimento delle memorizzazioni.

*Z* Memorie per l'ottenimento dei conteggi.

#### *Impostazione e digitazione di semplici istruzioni in lista di istruzione*

Come primo approccio alla programmazione conviene riferirsi a una semplice applicazione.

#### 10.3.1 Comando di un attuatore con istruzione AND

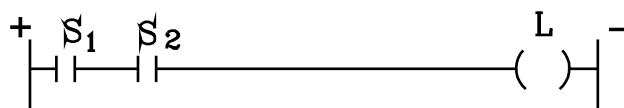
Si debba accendere una lampada *L* pigiando contemporaneamente i due tasti *S<sub>1</sub>*, *S<sub>2</sub>*.

Si tratta della semplice operazione *AND*:

$$L = S_1 \cdot S_2 \quad (10.3.1.1)$$

Lo schema a contatti è riportato nella figura *fig.2.9.41*

*fig.2.9.41*



Si scelgano le occupazioni riportate nella tabella.

Elemento	Simbolo	Canale	Morsetto
Pulsante	$S_1$	$E1.0$	4
Pulsante	$S_2$	$E1.1$	3
Lampada	L	$A2.0$	4

L'operazione logica espressa dalla (10.3.1.1) consiste nel porre in *AND* l'ingresso  $E1.0$  con  $E1.1$  e il risultato logico assegnarlo all'uscita  $A2.0$ .

Il linguaggio in lista di istruzione si svolge scrivendo una istruzione alla volta in corrispondenza di un indirizzo e inviandola alla memoria di programma.

Il programma proposto si svolge nei seguenti passi:

1. Nel blocco del primo indirizzo si scrive l'operando  $E1.0$  preceduto dal codice operativo  $U$  che immette l'operando nell'ingresso di una porta logica *AND* .  
 $U E1.0$
2. Nel secondo indirizzo si scrive l'operando  $E1.1$  preceduto dal codice operativo  $U$  che immette l'operando in una altro ingresso della porta logica *AND*.  
 $U E1.1$
3. Nel terzo blocco si scrive "=" che effettua l'operazione precedentemente impostata e il valore logico viene assegnato all'uscita  $A2.0$   
 $=A2.0$

Il programma sarà così composto dai seguenti blocchi di istruzione:

$U E1.0$   
 $U E1.1$   
 $=A2.0$

#### *Digitazione del programma sulla tastiera*

Durante la digitazione porre l'AG (PLC) in *Stop* con l'apposita levetta.

Per l'introduzione del programma si eseguano operazioni di digitazione elencate qui di seguito.

- Liberare la memoria e ripristinare lo stato originario. Si pigia prima il *tasto bianco di predisposizione*  $\bigcirc$ , posto nell'estremità inferiore a sinistra della tastiera e quindi il tasto con l'indicazione *XI*.
- Pigiare il tasto di introduzione  $\boxed{\rightarrow}$
- Pigiare il tasto di *predisposizione*  $\bigcirc$  e poi il tasto con l'indicazione *PB*. Questa indicazione compare sul display
- Scrivere il numero di blocco che contraddistingue il programma. Immettere un numero da 0 a 255: pigiare per esempio il tasto con il numero  $I$ .
- Pigiare il tasto di introduzione con il simbolo  $\blacklozenge$



Alla fine di ogni riga digitata occorre pigiare detto tasto di introduzione.

- Iniziare la digitazione delle righe di istruzione. Alla fine di ogni pigiare il tasto di introduzione.
- 000      *UE1.0*      ◊
- 002      *UE1.1*      ◊
- 004      *=A2.0*      ◊

Gli indirizzi li pone automaticamente il *PG*.

- Ultimate le righe di programma pigiare di nuovo il tasto di introduzione ◊ . Sul display compare il messaggio che richiede se il programma deve essere inviato nell'AG. Per conferma premere ancora il tasto ◊ .
- Porre l'AG in *Run* manovrando l'apposita levetta.
- Eseguire il programma: pigiando i due start  $S_1, S_2$  si accende la lampada.

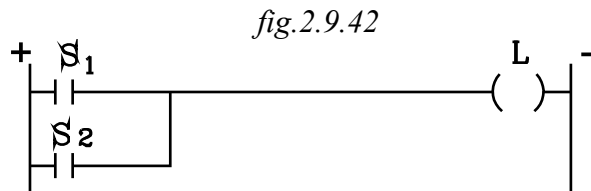
### 10.3.2 Comando di un attuatore con istruzione OR

Si debba accendere una lampada  $L$  pigiando uno dei due tasti  $S_1, S_2$ .

Si tratta della semplice operazione *OR*:

$$L = S_1 + S_2 \quad (10.3.2.1)$$

Lo schema a contatti è riportato nella figura *fig.2.9.42*



Si scelgano le occupazioni riportate nella tabella.

Elemento	Simbolo	Canale	Morsetto
Pulsante	$S_1$	<i>E1.0</i>	4
Pulsante	$S_2$	<i>E1.1</i>	3
Lampada	L	<i>A2.0</i>	4

L'operazione logica espressa dalla (10.3.2.1) consiste nel porre in *OR* l'ingresso *E1.0* con *E1.1* e il risultato logico assegnarlo all'uscita *A2.0*.

Il programma proposto si svolge nei seguenti passi:

1. Nel blocco del primo indirizzo si scrive l'operando *E1.0* preceduto dal codice operativo *O* che immette l'operando nell'ingresso di una porta logica *OR*.

$$O E1.0$$

2. Nel secondo indirizzo si scrive l'operando *E1.1* preceduto dal codice operativo *O* che immette l'operando in un altro ingresso della porta logica *OR*.

*O E1.1*

3. Nel terzo blocco si scrive "=" che effettua l'operazione precedentemente impostata e il valore logico viene assegnato all'uscita *A2.0*

*=A2.0*

Il programma sarà così composto dai seguenti blocchi di istruzione:

*O E1.0*

*O E1.1*

*=A2.0*

Per la digitazione del programma si proceda come nell'esercizio precedente.

### 10.3.3 Comando di un attuatore con combinazione di istruzioni AND ed OR

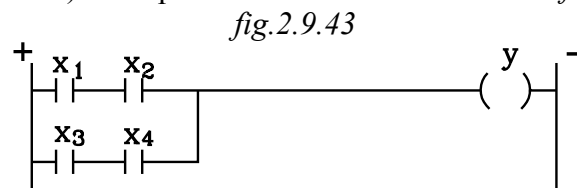
In un insieme di operazioni logiche l'operazione *AND* prevale sull'*OR*. Si ha la stessa proprietà che si riscontra nelle espressioni matematiche nelle quali la moltiplicazione prevale sulla somma: vengono effettuate prima le moltiplicazioni e poi si sommano i risultati.

Così consideriamo l'operazione logica:

$$y = x_1 \cdot x_2 + x_3 \cdot x_4 \quad (10.3.3.1)$$

Viene prima effettuata l'operazione  $x_1 \cdot x_2$  e il risultato viene sommato al prodotto logico  $x_3 \cdot x_4$ .

L'operazione logica (10.3.3.1) corrisponde allo schema a contatti di *fig.2.9.43*



Le variabili  $x_1, x_2, x_3, x_4$  siano dei pulsanti che comandano l'accensione della lampada  $y$ .

Si scelgano le occupazioni riportate nella tabella.

Elemento	Simbolo	Canale	Morsetto
Pulsante	$x_1$	<i>E1.0</i>	4
Pulsante	$x_2$	<i>E1.1</i>	3
Pulsante	$x_3$	<i>E1.3</i>	6
Pulsante	$x_4$	<i>E1.4</i>	5
Lampada	$y$	<i>A2.0</i>	4

Il programma sarà così composto dai seguenti blocchi di istruzione:

*U E1.0*

*U E1.1*

*O*

*U E1.3*

*U E1.4*

*=A2.0*

Porre il PLC in Run e provare il programma. Pigiando contemporaneamente  $x_1, x_2$  oppure  $x_3, x_4$  si deve accendere la lampada

Consideriamo come esempio una semplice applicazione.

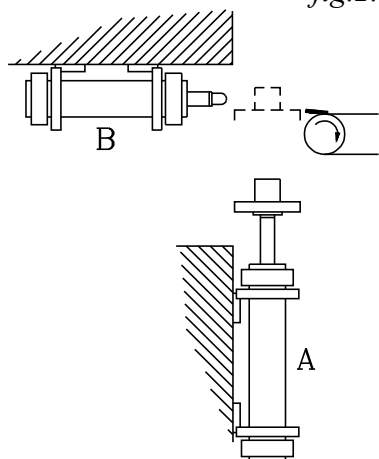
Ciclo di lavoro  $A^+ B^+ A^- B^-$ 

## 10.3.3.1 Tema

Dei cestelli devono essere trasportati da un piano inferiore ad uno superiore e, in questo, spinti su un nastro trasportatore. Il ciclo di lavoro è azionato da uno Start.

Si determini il circuito pneumatico che effettui il ciclo di lavoro, non considerando altri eventuali segnali di sicurezza.

fig.2.9.44



Il problema è stato già precedentemente risolto nel volume 2° con circuito pneumatico.

Per brevità si tralascia l'indicazione dei componenti utilizzati, il diagramma corsa passo ecc. (vedi volume 2°); si concentra l'attenzione nella traduzione delle equazioni logiche nel programma in lista di istruzioni.

Per un controllo totale del ciclo vengono utilizzate le equazioni logiche complete. Ciò d'altra parte non richiede alcun aggravio circuitale ma semplicemente un allungamento delle istruzioni digitate.

Le equazioni logiche complete sono riportate nella

seguente tabella.

Equazioni logiche complete

$$A^+ = a_0 \cdot b_0 \cdot Start$$

$$B^+ = a_1 \cdot b_0$$

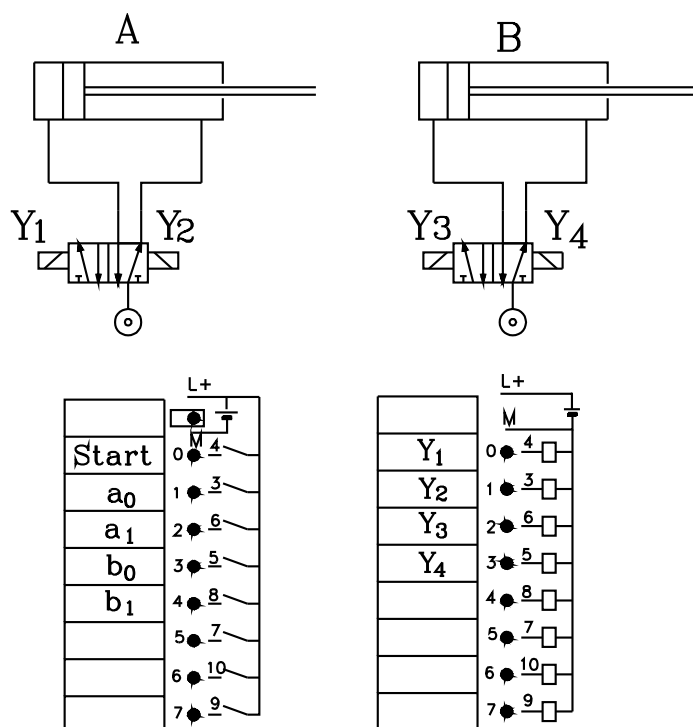
$$A^- = a_1 \cdot b_1$$

$$B^- = a_0 \cdot b_1$$

## Occupazioni

Elemento	Simbolo	Canale	Morsetto
Start	Start	E1.0	4
finecorsa	$a_0$	E1.1	3
finecorsa	$a_1$	E1.2	6
finecorsa	$b_0$	E1.3	5
finecorsa	$b_1$	E1.4	8
Elettrovalvola	$Y_1$	A2.0	1
Elettrovalvola	$Y_2$	A2.1	3
Elettrovalvola	$Y_3$	A2.2	6
Elettrovalvola	$Y_4$	A2.3	5

fig.2.9.45



Dalle equazioni logiche si può ricavare direttamente il programma in lista di istruzioni.

Equazione logica	Lista di istruzioni
$A^+ = a_0 \cdot b_0 \cdot Start$	UE1.1 UE1.3 UE1.0 =A2.0
$B^+ = a_1 \cdot b_0$	UE1.2 UE1.3 =A2.2
$A^- = a_1 \cdot b_1$	UE1.2 UE1.4 =A2.1
$B^- = a_0 \cdot b_1$	UE1.1 UE1.4 =A2.3

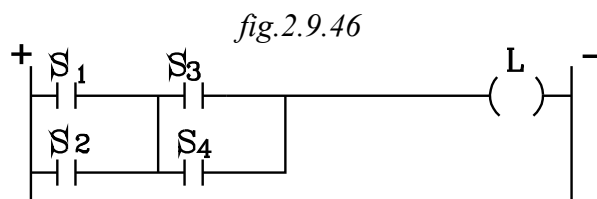
Inviare il programma nel PG, porlo in Run e provare il funzionamento.

Occorre fare attenzione all'assorbimento di corrente delle elettrovalvole. Se è elevato, si inviano i segnali di uscita del PLC a 4 bobine ausiliarie  $K_1, K_2, K_3, K_4$  le quali, con contatti normalmente aperti comanderanno rispettivamente le bobine delle elettrovalvole  $Y_1, Y_2, Y_3, Y_4$ .

Ai canali di uscita A2.0, A2.1, A2.2, A2.3 si collegheranno le bobine ausiliarie  $K_1, K_2, K_3, K_4$ .

### 10.3.4 Lista di istruzioni con serie di paralleli

Le istruzioni di comandi combinatori finora considerati non richiedevano l'uso di parentesi. Si consideri ora il circuito a contatti di *fig.2.9.46*



L'equazione logica del circuito è:

$$L = (S_1 + S_2) \cdot (S_3 + S_4) \quad (10.3.4.1)$$

Si scelgano le occupazioni riportate nella tabella.

Elemento	Simbolo	Canale	Morsetto
Pulsante	$S_1$	<i>E1.0</i>	4
Pulsante	$S_2$	<i>E1.1</i>	3
Pulsante	$S_3$	<i>E1.3</i>	6
Pulsante	$S_4$	<i>E1.4</i>	5
Lampada	$L$	<i>A2.0</i>	4

Per la stesura dell'equazione logica (10.3.4.1) occorre l'impiego di parentesi.

Occorre osservare che l'operazione posta entro la prima parentesi è in *AND* (*U*) con quella posta entro la seconda parentesi.

Occorre quindi preordinare l'operazione *AND* avanti alla prima parentesi. L'istruzione è composta dal carattere *U* seguito dalla parentesi aperta "*(*":

*U*(

Sulla tastiera, nella zona a sinistra, vi è il tasto con l'indicazione "*U*(".

Entro la parentesi vi è l'operazione *OR* tra i due segnali inviati dai pulsanti  $S_1, S_2$  e quindi, considerata l'assegnazione, tra i due operandi *E1.0* e *E1.1*.

*L'operazione OR, quando è compresa entro una parentesi, è ottenuta pigiando il tasto posto in corrispondenza della istruzione, composta dal carattere O seguito da un punto: "O."*

Così si hanno le seguenti due istruzioni che si riferiscono all'operazione *OR* tra i due operandi *E1.0* ed *E1.1*:

*O. E1.0*

*O. E1.1*

Alla fine delle due istruzioni dell'operazione *OR* tra *E1.0* ed *E1.1* si deve chiudere la parentesi, che conclude il primo termine dell'operazione *AND*, iniziata con "*U*(".

La chiusura della parentesi va posta su una riga di programma:

)

Inizia ora il secondo termine dell'operazione *AND*, effettuata sull'operazione *OR* tra i due operandi *E1.2* ed *E1.3*, posti entro la parentesi.

Si procede così alla stessa maniera eseguita per la prima parentesi. In ordine:

- Si preme il tasto:

*U*(

- Si digitano le due istruzioni:

*O. E1.2*

*O. E2.3*

- Si preme il tasto di chiusura parentesi

)  
 Alla fine si assegna il valore logico ottenuto dalle operazioni effettuate sugli operandi all'uscita  $A2.0$ :

$=A2.0$

Riassumendo in ordine le digitazioni da effettuare si ha il seguente schema delle operazioni:

○	XI	◇	Cancellazione AG.
→			Tasto di introduzione.
○	PB 1	◇	Impostazione.
000	U(		Preordina operazione <i>AND</i> avanti alla parentesi.
002	O. EI.0		Immissione dell'operando <i>EI.0</i> nella porta <i>OR</i> .
004	O. EI.1		Immissione dell'operando <i>EI.0</i> nella porta <i>OR</i> .
006	)		Conclusione dell'operazione posta entro parentesi che andrà in <i>AND</i> con la successiva operazione.
008	U(		Preordina operazione <i>AND</i> avanti alla parentesi.
00A	O. EI.0		Immissione dell'operando <i>EI.0</i> nella porta <i>OR</i> .
00B	O. EI.1		Immissione dell'operando <i>EI.0</i> nella porta <i>OR</i> .
00C	)		Conclusione dell'operazione posta entro parentesi.
00D	=A2.0		Assegnazione delle operazioni effettuate all'uscita <i>A2.0</i>
00E	◇		Conclusione del programma. Viene richiesto se si deve trasferire il programma dal <i>PG</i> nell' <i>AG</i> : →   — — > AG ?
	◇		Conferma.

Porre il PLC in Run e verificare l'esattezza di esecuzione del programma.

Pigiare contemporaneamente i due pulsanti  $S_1, S_3$  e si deve accendere la lampada.

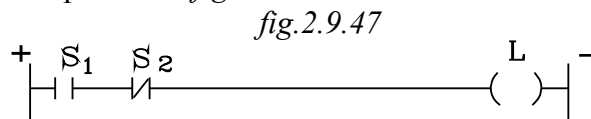
### 10.3.5 Comando di un attuatore con la negazione di una operazione su un operando

La negazione si ottiene con il simbolo  $N$  attivato dall'apposito tasto della tastiera.

Si consideri la semplice funzione binaria:

$$L = S_1 \cdot \overline{S_2} \quad (10.3.5.1)$$

Il cui schema a contatti è riportato in *fig.2.9.47*



Si scelgano le occupazioni riportate nella tabella.

Elemento	Simbolo	Canale	Morsetto
Pulsante	$S_1$	<i>EI.0</i>	4
Pulsante	$S_2$	<i>EI.1</i>	3
Lampada	L	<i>A2.0</i>	4

Il programma in lista di istruzioni è il seguente:

000  $U E1.0$   
 002  $U N E1.1$   
 004  $= A2.0$

Trasferito il programma nell'AG e posto questo in Run verificare l'esattezza dell'esecuzione del programma. Pigiato solamente il tasto  $S_I$  si accende la lampada; pigiati entrambi i tasti la lampada si spegne.

### 10.3.6 Circuito di memoria con autoritenuta

Nei circuiti sequenziali sono state frequentemente utilizzate le memorie con prevalenza alla cancellazione. Un circuito di questo tipo si vuole tradurre nel programma in lista di istruzioni.

Per la comprensione conviene riferirsi ad un esempio pratico da verificare in laboratorio.

---

### IN LABORATORIO

---

*Si debba effettuare il semplice ciclo  $A^+ A^-$  con un cilindro a semplice effetto, utilizzando una valvola monostabile. Si utilizzino solamente le variabili attive. Il ciclo è controllato da finecorsa e attivato da uno Start.*

Qui si tralasciano le indicazioni dei componenti utilizzati, il diagramma corsa passo ecc.

Si concentra l'attenzione sulla stesura del programma in lista di istruzioni.

*Equazioni logiche con variabili attive*

$$A^+ = a_0 \cdot Start$$

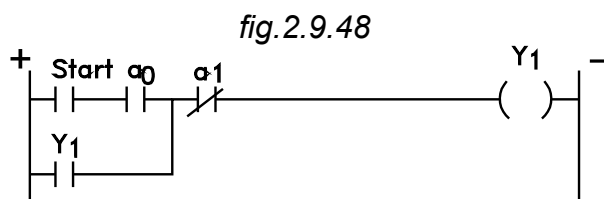
$$A^- = a_1$$

$$\text{Comando elettrovalvola monostabile } Y_1 \begin{cases} \text{Set} & A^+ = a_0 \cdot Start \\ \text{Re set} & A^- = a_1 \end{cases} \quad (10.3.6.1)$$

Dalle espressioni (10.3.6.1) si ricava la funzione di comando di  $Y_1$  :

$$Y_1 = (a_0 \cdot Start + Y_1) \cdot \overline{a_1} \quad (10.3.6.2)$$

Dalla funzione di trasmissione (10.3.6.2) si ricava il circuito a contatti è riportato in figura *fig.2.9.48*.



Le occupazioni sono riportate nella seguente tabella.

Elemento	Simbolo	Canale	Morsetto
Start	$Start$	$E1.0$	4
Finecorsa	$a_0$	$E1.1$	3
Finecorsa	$a_1$	$E1.2$	6
Elettrovalvola	$Y_1$	$A2.0$	4

Per comprendere la lista di istruzioni sostituiamo nella (10.3.6.2) ai simboli dei componenti quelli dei rispettivi canali occupati, ponendo al secondo membro l'attuatore.

$$((E1.1 \cdot E1.0) + A2.0) \cdot \overline{E1.2} = A2.0 \quad (10.3.6.3)$$

Occorre osservare che nella espressione vi sono due parentesi: l'esterna, pone in *AND* l'operazione scritta nel suo interno con  $\overline{E1.2}$ ; l'altra interna, pone in *OR* l'operazione scritta nel suo interno con *A2.0*.

Occorre all'inizio delle parentesi preordinare l'operazione che si effettuerà tra la espressione entro parentesi e l'operando seguente.

Così per la prima parentesi esterna si dovrà preordinare l'operazione *AND* tra l'espressione posta nel suo interno e  $\overline{E1.2}$ : *U*(

Per la seconda parentesi interna, invece, occorre preordinare l'operazione *OR* tra l'espressione interna e *A2.0*: *O*(

Occorre osservare poi che l'operando *E1.2* deve essere prima negato e poi immesso in una porta *AND*: *U N E1.2*

La espressione (10.3.6.3) conviene scriverla per comodità ,momentaneamente, nella seguente forma, utile per la trasformazione in lista di istruzioni, nella quale il prodotto logico tra due termini viene sostituito con il simbolo *U* su ognuno di essi l'operazione *OR* con il simbolo *O*. L'operazione *OR* entro la parentesi è ottenuta con "O.":

$$U(O(UE1.1 UE1.0) O. A2.0) U N E1.2 = A2.0 \quad (10.3.6.4)$$

Le istruzioni (10.3.6.4) scritte su un'unica riga per comodità di traduzione in lista istruzioni dalla espressione logica (10.3.6.2), vanno introdotte una alla volta nel *PG*, come riportato qui di seguito.

```

000  U(
002  O(
004  UE1.1
006  UE1.0
008  )
00A  O. A2.0
00B  )
00C  U N E1.2
00D  = A2.0

```

Gli indirizzi 000 , 002... vengono posti automaticamente del *PG*

Trasferire il programma dal *PG* nell'*AG*, porre questo in *Run* e pigiare lo *Start*.

### 10.3.7 Circuito con memorizzazioni provvisorie - Merker (Flag)

Nei circuiti con segnali bloccanti si sono utilizzate delle memorie intermedie che memorizzavano stati logici provvisori da porre in *AND* con i segnali primari provenienti dall'esterno.

Nell'elettropneumatica a tale scopo si sono utilizzate bobine ausiliarie. Queste nel *PLC* sono sostituite da memorie provvisorie dette *Merker*.

Queste memorie sono contraddistinte dalla lettera *M*, seguita da due numeri separati dal punto ".", indicanti il *parametro* (*i Merker costituiscono una matrice di m righe ed n colonne*).

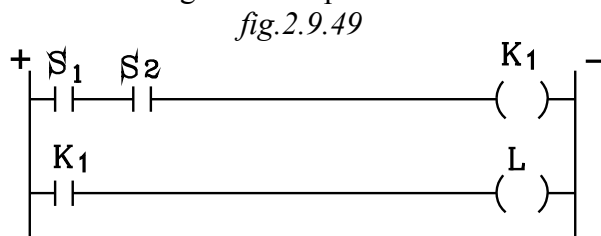
Così un *Merker* può essere indicato con "*MI.0*".



I *Merker* costituiscono una uscita provvisoria e quindi va trattata come tale, assegnandogli il valore logico proveniente da una operazione: per esempio  $=MI.0$

Il valore logico del *Merker* potrà poi essere interrogato per costituire un operando di una operazione logica con altri ingressi od uscite.

Per chiarire le idee consideriamo il seguente semplice circuito a contatti di *fig.2.9.49*



In un circuito cablato lo schema di *fig.2.9.49* risolverebbe il seguente problema: i due pulsanti  $S_1, S_2$  debbono eccitare la bobina  $K_1$  la quale con un contatto normalmente aperto comanda l'accensione della lampada  $L$ .

Nel PLC la bobina ausiliaria  $K_1$  è sostituita da un *Merker (Flag)* che con il suo stato logico interrogato comanda la lampada  $L$ .

Si considerino le seguenti occupazioni poste in tabella. In questa il *Merker* è considerato come un'uscita, ma non occupa, in effetti, nessun morsetto d'uscita.

Elemento	Simbolo	Canale	Morsetto
Pulsante	$S_1$	$E1.0$	4
Pulsante	$S_2$	$E1.1$	3
<i>Merker</i> $K_1$		$MI.0$	
Lampada	L	$A2.0$	4

Tenendo conto delle precedenti precisazioni il circuito a contatti di figura si può tradurre nel seguente programma in lista di istruzioni:

```

000  U E1.0
002  U E1.1
004  =MI.0
006  U MI.0
008  =A2.0

```

Trasferito il programma nell'AG e posto questo in Run, si verifica l'esatta esecuzione: pigiando entrambi i pulsanti si accende la lampada.

Si noti che quando nella zona di Test vi è un solo contatto ( $MI.0$  nella quarta riga del programma) l'istruzione si compone dell'operando preceduto dalla  $U$  (AND ad un solo ingresso).

Per esercitazione di programmazione in lista di istruzioni contenenti memorie di autoritenuta e *Merker*, si consideri il ciclo ad L più volte proposto e risolto con diverse tecniche.

---

 IN LABORATORIO
 

---

## 10.3.8 Tema

Su una serie di pezzi deve essere effettuata una stampigliatura.

Ciascun pezzo viene posto a mano su un apposito sistema di bloccaggio. Azionati due Start viene bloccato il pezzo attraverso l'azione di un cilindro pneumatico a doppio effetto. Successivamente un altro cilindro effettua la doppia corsa di stampigliatura. Ultimata questa si apre il sistema di bloccaggio e il pezzo viene tolto a mano. Si utilizzino valvole bistabili

Come per i precedenti esercizi qui viene concentrata l'attenzione solamente sulla traduzione delle equazioni logiche in lista di istruzioni.

Nel presente tema proposto il ciclo dei due cilindri è:

$$A^+ B^+ B^- A^-$$

Il ciclo è stato impostato e risolto nel 2° Volume con un circuito elettropneumatico in logica cablata (punto 8.4).

Dalla mappa di Karnaugh si ottengono le equazioni logiche:

$$\begin{aligned} A^+ &= \bar{x} \cdot S_1 \cdot S_2 \\ B^+ &= a_1 \cdot \bar{x} \\ X_1 &= b_1 \\ B^- &= x \\ A^- &= b_0 \cdot x \\ X_0 &= a_0 \end{aligned} \quad (10.3.8.1)$$

La memoria X è comandata dai segnali:

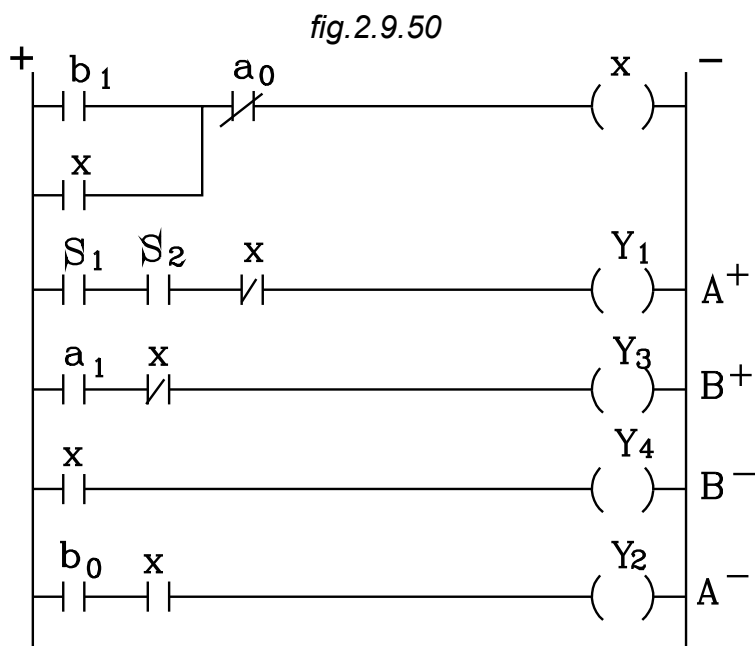
$$\text{Memoria } X \begin{cases} \text{Set} & X_1 = b_1 \\ \text{Reset} & X_0 = a_0 \end{cases}$$

L'equazione logica della memoria è:  $X = (b_1 + x) \cdot \bar{a}_0$  (10.3.8.2)

Si assumano le occupazioni riportate nella tabella . In questa è anche indicata la memoria temporanea associata al Merker M1.0

Elemento	Simbolo	Canale	Morsetto
Start	$S_1$	E1.0	4
Start	$S_2$	E1.1	3
finecorsa	$a_0$	E1.2	6
finecorsa	$a_1$	E1.3	5
finecorsa	$b_0$	E1.4	8
finecorsa	$b_1$	E1.5	7
Elettrovalvola	$Y_1$	A2.0	1
Elettrovalvola	$Y_2$	A2.1	3
Elettrovalvola	$Y_3$	A2.2	6
Elettrovalvola	$Y_4$	A2.3	5
Merker X	M1.0		

Le equazioni logiche (10.3.8.1) (10.3.8.2) si traducono nello schema a contatti di *fig. 2.9.50*



Dalle equazioni logiche o dallo schema a contatti si può ricavare il programma in lista di istruzioni

Equazione logica	Lista istruzioni
$X = (b_1 + x) \cdot \bar{a}_0$	U(
	O. E1.5
	O. M1.0
	)
	UNE1.2
	= M1.0
$A^+ = Y_1 = S_1 \cdot S_2 \cdot \bar{x}$	UE1.0
	UE1.1
	UN
	M1.0
$B^+ = a_1 \cdot \bar{x}$	= A2.0
	UE1.3
	UN
$B^- = x$	M1.0
	= A2.2
	UM1.0
$A^- = b_0 \cdot x$	= A2.3
	UE1.4
	UM1.0
	= A2.1

---



---

*IN LABORATORIO*

---

### 10.3.9 Tema

*Risolvere il tema precedente utilizzando valvole monostabili*

Si impiegano due valvole monostabili con ritorno a molla.

Come al solito indichiamo con  $Y_1$  la bobina della valvola monostabile che comanda il cilindro A, e con  $Y_3$  quella che comanda il cilindro B.

Le equazioni logiche dei comandi della memoria temporanea e di fuoriuscita e rientro steli sono quelle del tema precedente.

$$A^+ = \bar{x} \cdot S_1 \cdot S_2$$

$$B^+ = a_1 \cdot \bar{x}$$

$$X_1 = b_1$$

$$B^- = x$$

$$A^- = b_0 \cdot x$$

$$X_0 = a_0$$

Si può associare ad ogni valvola un circuito di memoria con autoritenuta.

$$A^+ \quad A^- \rightarrow \text{Valvola } Y_1 \quad \begin{cases} \text{Set} & A^+ = S_1 \cdot S_2 \cdot \bar{x} \\ & A^- = b_0 \cdot x \end{cases}$$

Equazione logica di  $Y_1$ :

$$Y_1 = (S_1 \cdot S_2 \cdot \bar{x} + Y_1) \cdot \overline{b_0 \cdot x} \quad (10.3.9.1)$$

Si pone:

$$\boxed{H_1 = b_0 \cdot x} \quad (8.3.9.2)$$

L'equazione logica (8.3.9.1) si può scrivere nella forma:

$$\boxed{Y_1 = (S_1 \cdot S_2 \cdot \bar{x} + Y_1) \cdot \overline{H_1}} \quad (10.3.9.3)$$

$$B^+ \quad B^- \rightarrow \text{Valvola } Y_3 \quad \begin{cases} \text{Set} & B^+ = a_1 \cdot \bar{x} \\ & B^- = x \end{cases}$$

Equazione logica di  $Y_3$  sarà:

$$Y_3 = (a_1 \cdot \bar{x} + Y_3) \cdot \bar{x}$$

Svolgendo l'operazione si ottiene:

$$Y_3 = a_1 \cdot \bar{x} + Y_3 \cdot \bar{x}$$

Raccogliendo  $\bar{x}$  si ottiene:

$$Y_3 = (a_1 + Y_3) \cdot \bar{x} \quad (10.3.9.4)$$

La memoria temporanea X ha l'equazione logica ricavata nel tema precedente:

$$X = (b_1 + x) \cdot \overline{a_0} \quad (10.3.9.5)$$

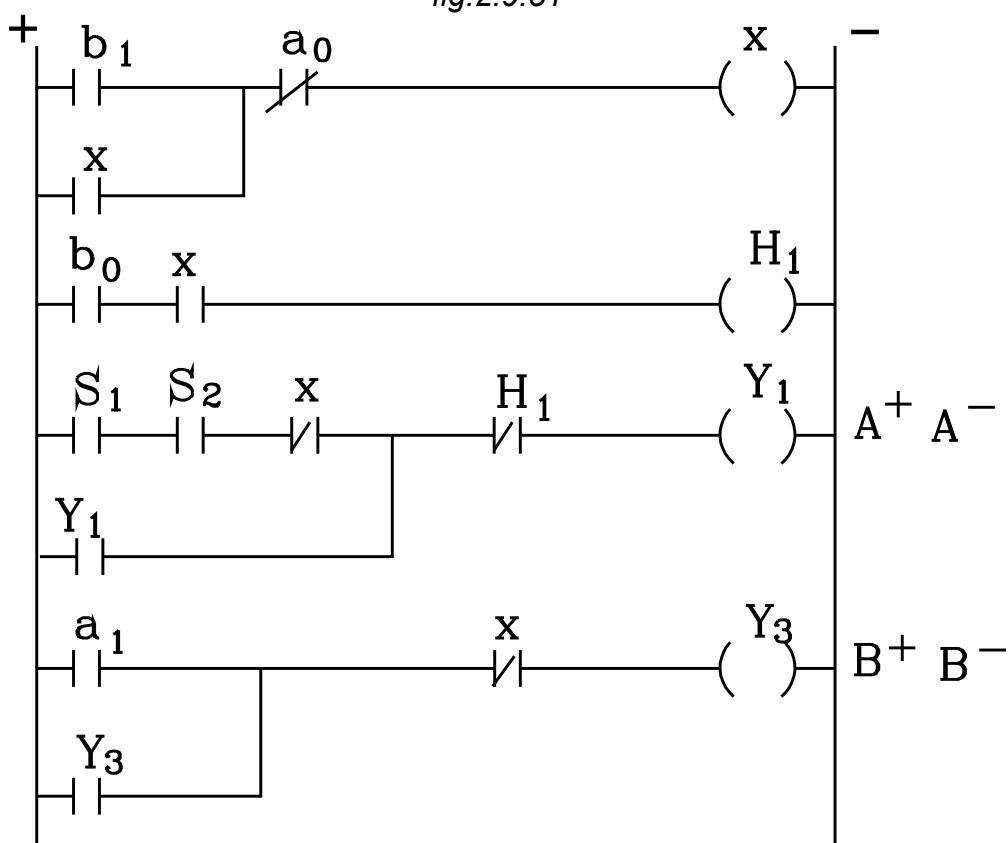
## Occupazioni

Nella tabella sono riportate le occupazioni scelte per gli ingressi e le uscite; inoltre sono indicate le due memorie temporanee associate ai segnali  $X$  ed  $H_1$

Elemento	Simbolo	Canale	Morsetto
Start	$S_1$	$E1.0$	4
Start	$S_2$	$E1.1$	3
finecorsa	$a_0$	$E1.2$	6
finecorsa	$a_1$	$E1.3$	5
finecorsa	$b_0$	$E1.4$	8
finecorsa	$b_1$	$E1.5$	7
Elettrovalvola	$Y_1$	$A2.0$	1
Elettrovalvola	$Y_3$	$A2.1$	6
Merker $X$		$M1.0$	
Merker $H_1$		$M1.1$	

Le equazioni logiche (10.3.9.2), (10.3.9.3), (10.3.9.4), (10.3.9.5) si traducono nel circuito a contatti di *fig.2.9.51*

fig.2.9.51



Dalle equazioni logiche o dallo schema a contatti si può ricavare il programma in lista di istruzioni

Equazione logica	Lista istruzioni
$X = (b_1 + x) \cdot \overline{a_0}$	U(
	O. E1.5
	O. M1.0
	)
	U N E1.2
	= M1.0
$H_1 = b_0 \cdot x$	U E1.4
	U M1.0
	= M 1.1
$Y_1 = ((S_1 \cdot S_2 \cdot \overline{x}) + Y_1) \cdot \overline{H_1}$	U(
	O(
	U E1.0
	U E1.1
	U N
	M1.0
	)
	O. A2.0
	)
	U N
M1.1	
=A2.0	
$Y_3 = (a_1 + Y_3) \cdot \overline{x}$	U(
	O. E1.3
	O. A2.1
	)
	U N
	M1.0
= A2.1	

#### 10.4 Operazione di temporizzazione

Possono essere impostati diversi tipi di temporizzatori sia come impulso di una certa durata o come ritardo all'inserzione.

In tutti i casi occorre che sia caricato un valore del tempo o per la durata dell'impulso o per il ritardo all'inserzione.

L'istruzione si compone come al solito di un codice operativo ed di un operando, contenente il simbolo e i parametri indicanti il tempo.

L'istruzione del caricamento di un tempo è così composta:

*Codice operativo* L

*Operando* È composto dal simbolo *KT* seguito dai parametri indicanti il valore del tempo

*Parametri* I parametri sono costituiti da un numero da 0 a 999 seguito da un punto e da un altro numero da 0 a 3, indicante la base dei tempi.

La base dei tempi indica l'unità di misura con la quale si intende moltiplicato il primo numero espresso nei parametri.

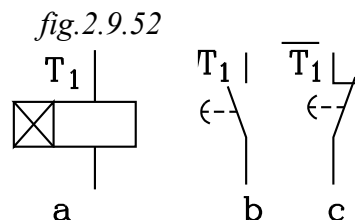
<i>Base</i>	0	1	2	3
<i>Unità</i>	0,01 s	0,1	1	10

Così per esempio se si vuole caricare un tempo di 4 s, l'istruzione può porsi nelle seguenti forme equivalenti tra loro:

$L \text{ KT } 400.0$

$L \text{ KT } 40.1$

$L \text{ KT } 4.2$



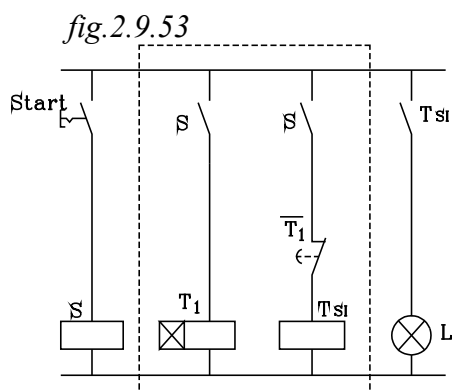
Nei circuiti cablati un relè temporizzatore con ritardo alla eccitazione viene indicato dal simbolo di fig.2.9.52 a. Il relè comanda dei contatti normalmente aperti (fig.2.9.52 b) o dei contatti normalmente chiusi (fig.2.9.52 c).

Per comprendere le istruzioni di temporizzazione nel linguaggio in lista di istruzioni conviene riferirsi all'equivalente circuito a contatti che effettua la stessa operazione.

#### 10.4.1 Temporizzazione di un impulso di durata $\Delta t$ - Temporizzazione SI

Si vuole effettuare la seguente temporizzazione.

Inviato il segnale di attivazione al temporizzatore si deve attivare l'uscita e questa rimane allo stato logico 1 per il tempo  $\Delta t$  preordinato, e se continua, durante questo intervallo, ad essere attivo il segnale di attivazione. Se prima che passi il tempo  $\Delta t$  il segnale di attivazione si porta allo stato 0, anche l'uscita si porta in tale stato.



Per comprendere l'impostazione della temporizzazione si consideri il circuito di fig.2.9.53.

Il circuito temporizzatore è la parte compresa entro il riquadro tratteggiato.

Esso è composto da un relè  $T_1$  temporizzatore con ritardo alla eccitazione, comandato da un contatto normalmente aperto  $S$ , rappresentante il segnale di ingresso, proveniente nell'esempio in esame dalla bobina  $S$ , comandata dallo *Start* con aggancio meccanico.

La bobina ausiliaria  $T_{SI}$  è comandata dal contatto normalmente aperto  $S$  e dal normalmente chiuso  $\overline{T_1}$  del relè temporizzatore.

Il segnale che giunge sulla bobina ausiliaria  $T_{SI}$  è ritardato rispetto a quello di ingresso  $S$  del tempo  $\Delta t$  impostato sul relè temporizzatore.

La lampada  $L$  è comandata dal contatto normalmente aperto della bobina del relè ausiliario  $T_{SI}$ , e quindi si accende con un ritardo  $\Delta t$  rispetto al segnale di ingresso  $S$ .

Comunque, in mancanza del segnale  $S$  la bobina,  $T_{SI}$  si diseccita e la lampada si spegne.

Si è indicato con " $\Delta t$ " l'intervallo di ritardo all'eccitazione della bobina  $T_1$  del relè temporizzatore.

Immeso il segnale  $S$ , la bobina  $T_1$  non si eccita fino a che non è trascorso l'intervallo  $\Delta t$ : l'eccitazione si ha quando è presente il segnale di ingresso  $S$  e non si è nell'intervallo  $\Delta t$ . Si può scrivere quindi in forma logica:

$$T_1 = S \cdot \overline{\Delta t} \quad (10.4.1.1)$$

Il segnale che comanda la bobina ausiliaria  $T_{SI}$  è dato dalla operazione *AND*:

$$T_{SI} = S \cdot \overline{T_1} \quad (10.4.1.2)$$

Sostituendo la (10.4.1.1) si ha:

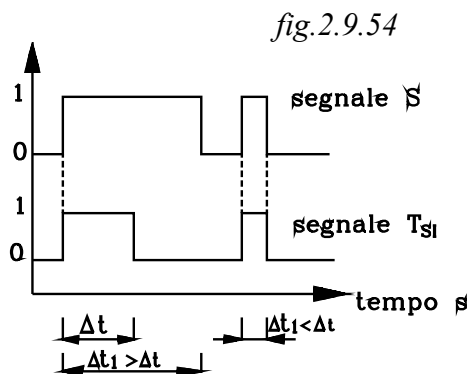
$$T_{SI} = S \cdot \overline{S \cdot \overline{\Delta t}} \quad (10.4.1.3)$$

Per il teorema di De Morgan si ha:

$$T_{SI} = S \cdot (\bar{S} + \Delta t)$$

$$T_{SI} = S \cdot \Delta t \quad (10.4.1.4)$$

La bobina  $T_{SI}$  si eccita, e quindi si ha il segnale di uscita dal temporizzatore, quando è presente il segnale di ingresso  $S$  e si è nell'intervallo di tempo  $\Delta t$ , impostato nel relè temporizzatore con ritardo alla cancellazione.



Nella figura fig.2.9.54 viene riportato il comportamento del segnale di uscita  $T_{SI}$  rispetto a quello di ingresso  $S$ .

In presenza del segnale di ingresso  $S$  si ha subito l'attivazione dell'uscita  $T_{SI}$ .

Se il segnale di ingresso  $S$  permane allo stato logico 1 per un tempo  $\Delta t_1$  superiore a  $\Delta t$  impostato nel relè temporizzatore, allora l'uscita  $T_{SI}$ , trascorso detto intervallo  $\Delta t$  si porta dallo stato logico 1 allo stato 0.

Se il segnale di ingresso  $S$  permane nello stato logico 1 per un tempo  $\Delta t_1$  inferiore a  $\Delta t$ , allora il segnale di uscita si annulla all'annullarsi del segnale di ingresso, come è espresso dalla (10.4.1.4).

Il tempo  $\Delta t$  viene fornito da un temporizzatore interno al PLC indicato dal simbolo  $T$  seguito da un numero:  $T1$ ,  $T2$ , ecc.

L'espressione (10.4.1.4) si potrà esprimere indicando il temporizzatore scelto:

$$T_I = S \cdot \Delta t$$

Lo stato logico di  $T_I$  viene poi assegnato al Merker  $T_{SI}$ .

Agli elementi del circuito di figura fig.2.9.53 si associano nel PLC gli operandi indicati nella seguente tabella:

Elemento	Simbolo	Operando
Start	$S$	$E0.1$
Relè tempor.	$T_I$	$T1$
Bobina relè	$T_{SI}$	$M1.0$
Lampada	$L$	$A2.0$

L'uscita del segnale temporizzato (impulso di durata  $\Delta t$ ) viene inviato o in una uscita o assegnato ad un Merker  $T_{SI}$  come nel caso considerato. Questa memoria temporanea può essere interrogata nel proseguo del programma.

Il tipo di temporizzazione di un impulso di durata  $\Delta t$  viene immesso con la funzione  $SI$ .

Il programma in lista di istruzioni della temporizzazione si compone dei seguenti passi:

- Si introduce il segnale di attivazione dato dallo start  $S$ :

$U E0.1$

- Si carica il tempo di durata dell'impulso. Si supponga di caricare  $t=5$  s.

$L KT 50.1$

- Si imposta il tipo di temporizzazione ( $SI$ ) e si attiva un temporizzatore : si scelga  $T1$ .

$SI T1$

- Si interroga ora lo stato logico del temporizzatore:

$U T1$

Si osservi che lo stato logico del temporizzatore è 1 per il tempo  $\Delta t$  impostato. Lo stato logico è quello poi assegnato all'uscita o ad un Merker.



- Si assegna lo stato logico del temporizzatore al *Merker*  $M1.0$  rappresentante la bobina ausiliaria:  
 $=M1.0$
- Si interroga lo stato logico del *Merker*:  
 $UM1.0$
- Si assegna lo stato logico del *Merker*  $UM1.0$  all'uscita  $A2.0$  corrispondente alla lampada  $L$ :  
 $=A2.0$

Riassumendo il programma in lista di istruzioni:

```

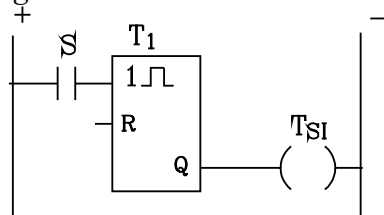
UE0.1
L KT 50.1
SI T1
UT1
=M 1.0
UM1.0
=A2.0

```

È evidente che lo stato logico del temporizzatore si poteva assegnare subito all'uscita  $A2.0$ ; lo si è assegnato ad un *Merker* soltanto per esercitazione.

Lo schema di *fig.2.9.53* è stato adoperato come riferimento per la comprensione del funzionamento.

*fig.2.9.55*



Nello schema a contatti lo si indicherà con un solo blocco avente in ingresso il segnale di eccitazione e in uscita il comando dell'attuatore o l'assegnazione ad un *Merker* ( $T_{SI}$ ).

Si effettui a laboratorio una semplice applicazione di temporizzazione.

## IN LABORATORIO

### 10.4.2 Tema

Da un caricatore automatico escono dei dischetti che debbono subire un controllo dimensionale. Caduto il dischetto su un apposito porta pezzo, questo viene spinto in posizione di misurazione. Dopo un tempo  $\Delta t=70$  s, occorrente per la misura e per togliere il dischetto dal porta pezzi, questo deve automaticamente rientrare nella posizione iniziale per raccogliere un altro dischetto.

Deve essere utilizzato una valvola di potenza monostabile e il ciclo è avviato da uno Start.

Qui vengono tralasciati i punti riguardanti la scelta dei componenti, il diagramma corsa passo ecc. concentrando l'attenzione sulla stesura del programma in lista di istruzioni.

È evidente che le relazioni da effettuare dopo l'esercitazione in laboratorio va completata con le parti tralasciate.

#### Equazioni logiche di comando

Il ciclo di lavoro è  $A^+ A^-$ . La corsa  $A^+$  di fuoriuscita stelo si ottiene solamente quando è toccato il finecorsa  $a_0$  (stelo rientrato) e si pigia lo Start. Il comando  $A_1$  di fuoriuscita stelo sarà quindi dato dall'AND dei due segnali.

$$A^+ = a_0 \cdot Start$$

Il rientro stelo si ottiene solamente quando: è toccato il finecorsa  $a_1$ , ed è trascorso il tempo  $\Delta t$  prescritto per misurare il pezzo e toglierlo dal portapezzi.

Il comando  $A_0$  di rientro stelo è dato dall'AND tra il finecorsa  $a_1$  e il temporizzatore.

Il segnale viene emesso quando è toccato  $a_1$  e *non si è* nell'intervallo  $\Delta t$ .

$$A^- = a_1 \cdot \overline{\Delta t}$$

Dovendo utilizzare una valvola monostabile, per il comando della sua bobina  $Y_1$ , occorre un circuito di memorizzazione con autoritenuta .

$$\text{Comando } A^+ \ A^- \rightarrow Y_1 \begin{cases} \text{Set} & A^+ = a_0 \cdot \text{Start} \\ \text{Reset} & A^- = a_1 \cdot \overline{\Delta t} \end{cases}$$

L'equazione logica di comando della valvola risulta:

$$Y_1 = (\text{Start} \cdot a_0 + Y_1) \cdot \overline{a_1 \cdot \Delta t}$$

Si pone:

$$H_1 = a_1 \cdot \overline{\Delta t} \quad (10.4.2.1)$$

Risulta quindi:

$$Y_1 = (\text{Start} \cdot a_0 + Y_1) \cdot \overline{H_1} \quad (10.4.2.2)$$

Scegliamo il temporizzatore  $T_1$  e il tipo di temporizzazione  $SI$ .  $T_1$  viene attivato dal finecorsa  $a_1$  e rimane attivo per il tempo  $\Delta t$ .  $T_1$  è dato dalla espressione logica:

$$T_1 = a_1 \cdot \Delta t$$

Si assegna lo stato logico di  $T_1$  al *Merker*  $T_{SI}$

$$T_{SI} = T_1 = a_1 \cdot \Delta t \quad (10.4.2.3)$$

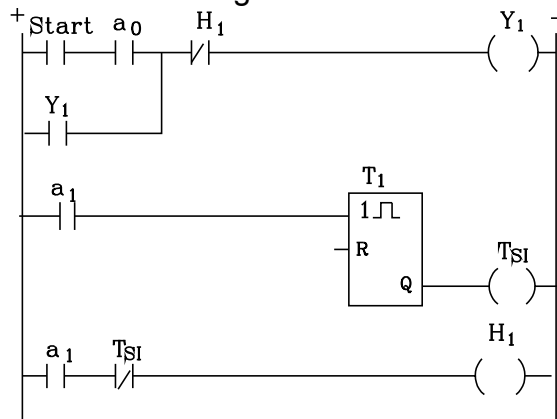
$T_{SI}$  nello stato logico 1 implica la contemporaneità del segnale del finecorsa  $a_1$  e dell'intervallo  $\Delta t$ . Queste condizioni sono espresse anche nella (10.4.2.1). Si può sostituire in questa il segnale  $T_{SI}$  al posto di  $\Delta t$ . Infatti:

$$a_1 \cdot \overline{\Delta t} = a_1 \cdot \overline{a_1 \cdot \Delta t} = a_1(\overline{a_1 + \Delta t}) = a_1 \cdot \overline{\Delta t} = H_1 \quad \text{quindi:}$$

$$\boxed{H_1 = a_1 \cdot \overline{T_{SI}}} \quad (10.4.2.4)$$

Dalle (10.4.2.2), (10.4.2.3), (10.4.2.4) si ottiene il circuito a contatti:

fig.2.9.56



Il circuito ha il seguente funzionamento:

- *Pigiato lo Start si eccita  $Y_1$  e fuoriesce lo stelo di A.*
- *Attivato il finecorsa  $a_1$  si attiva il temporizzatore  $T_1$  che fornisce l'impulso di durata  $\Delta t$*

- Lo stato logico di  $T_I$  è assegnato al Merker  $T_{SI}$  il quale apre il contatto NC posto in serie ad  $a_1$  nella terza linea; cosicché il Merker  $H_I$  durante l'intervallo  $\Delta t$  non è attivato.
- Trascorso l'intervallo di tempo  $\Delta t$  si disattiva  $T_{SI}$ , e si chiude il relativo contatto normalmente chiuso.
- Si eccita il Merker  $H_I$  che apre il relativo contatto sulla prima linea e si diseccita  $Y$  facendo rientrare lo stelo del cilindro  $A$

Occupazioni

Elemento	Simbolo	Canale	Morsetto
Start	<i>Start</i>	<i>E1.0</i>	4
Finecorsa	$a_0$	<i>E1.1</i>	3
Finecorsa	$a_1$	<i>E1.2</i>	6
Merker $T_{SI}$		<i>M1.0</i>	
Merker $H_I$		<i>M1.1</i>	
Elettrovalvola	$Y_I$	<i>A2.0</i>	4

Il tempo di temporizzazione è di 70 s

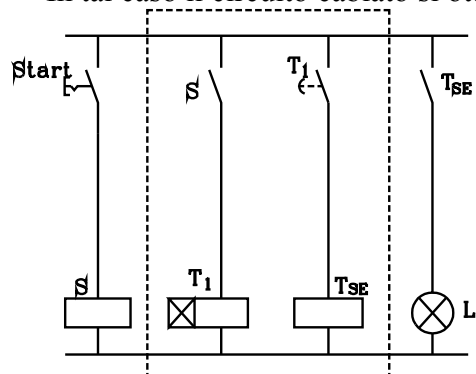
Il programma in lista di istruzioni è il seguente

Equazione logica	Lista istruzioni
$Y_I = ((Start \cdot a_0) + Y_I) \cdot \overline{H_I}$	<i>U(</i>
	<i>O(</i>
	<i>U E1.0</i>
	<i>U E1.1</i>
	<i>)</i>
	<i>O. A2.0</i>
	<i>)</i>
	<i>U N</i>
	<i>M1.1</i>
	<i>= A2.0</i>
Impostazione temporizzatore T1	<i>U E1.2</i>
	<i>L KT</i>
	<i>70.2</i>
	<i>SI T1</i>
	<i>U T1</i>
$H_I = a_1 \cdot \overline{T_{SI}}$	<i>= M1.0</i>
	<i>U E1.2</i>
	<i>U N</i>
	<i>M1.0</i>
	<i>= M1.1</i>

### 10.4.3 Temporizzazione di un segnale con ritardo alla inserzione - Tipo SE

In tal caso il circuito cablato si ottiene mediante lo schema di figura *fig.2.9.57*

*fig.2.9.57*



Il relè con ritardo alla eccitazione comanda con un contatto normalmente aperto la bobina ausiliaria  $T_{SE}$  alla quale viene assegnato il segnale  $T_I$  di uscita dal temporizzatore.

Questo ha lo stato logico  $I$  quando è presente il segnale di ingresso  $S$  di eccitazione e dopo che è trascorso il tempo di ritardo alla eccitazione  $\Delta t$ : quando si è fuori dell'intervallo  $\Delta t$ .

$$T_I = S \cdot \overline{\Delta t}$$

Viene poi assegnato alla bobina ausiliaria  $T_{SE}$  lo stato logico di  $T_I$

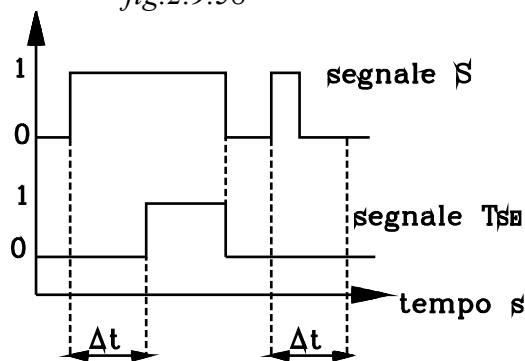
$$T_{SE} = T_I$$

La lampada  $L$  è comandata da  $T_{SE}$

$$L = T_{SE}$$

È evidente che la lampada  $L$  può essere inserita al posto della bobina ausiliaria  $T_{SE}$  ed interrogare direttamente lo stato logico di  $T_I$ .

fig.2.9.58



Nella figura fig.2.9.58 è rappresentato il comportamento dell'uscita del temporizzatore di tipo  $SE$ . Se il segnale di ingresso ha una durata superiore all'intervallo di ritardo  $\Delta t$  impostato, allora l'uscita  $T_{SE}$  si pone allo stato logico  $I$  dopo detto intervallo e cade allo stato  $0$  all'annullarsi del segnale di ingresso.

Se il segnale di ingresso ha una durata inferiore all'intervallo di ritardo  $\Delta t$  impostato, allora non può attivarsi alcun segnale di uscita dal temporizzatore.

Agli elementi del circuito di figura fig.2.9.57 si associano nel PLC gli operandi indicati nella seguente tabella:

Elemento	Simbolo	Operando
Start	$S$	$E0.1$
Relè temp.	$T_I$	$T1$
Bobina relè	$T_{SI}$	$M1.0$
Lampada	$L$	$A2.0$

Il tipo di temporizzazione del segnale con ritardo  $\Delta t$  alla inserzione viene immesso con la funzione  $SE$

Il programma in lista di istruzioni della temporizzazione si compone dei seguenti passi:

- Si introduce il segnale di attivazione, dato dallo start  $S$ :  
 $U E0.1$
- Si carica il tempo di durata dell'impulso. Si supponga di caricare  $t=5$ .  
 $L KT 50.1$
- Si imposta il tipo di temporizzazione ( $SE$ ) e si attiva un temporizzatore: si scelga  $T1$   
 $SE T1$
- Si interroga ora lo stato logico del temporizzatore:  
 $U T1$

Si osservi che lo stato logico del temporizzatore è  $I$  dopo l'intervallo di tempo  $\Delta t$  impostato.

Lo stato logico è quello poi assegnato all'uscita o ad un *Merker*.

- Si assegna lo stato logico del temporizzatore al *Merker*  $M1.0$  rappresentante la bobina ausiliaria:

$$=M 1.0$$

- Si interroga lo stato logico del *Merker*:

$$U M1.0$$

- Si assegna lo stato logico del *Merker*  $U M1.0$  all'uscita  $A2.0$  corrispondente alla lampada  $L$ :

$$=A2.0$$

Riassumendo il programma in lista di istruzioni:

```

U E0.1
L KT 50.1
SE T1
U T1
=M 1.0
U M1.0
=A2.0

```

Provare il programma. Attivando lo *Start* dopo 5 s si accende la lampada.

### 10.5 Operazione di conteggio

Il PLC permette operazioni di conteggio.

Viene qui descritta l'operazione di conteggio in decremento.

Sono disponibili 16 contatori interni, indicati con i simboli da *Z0* .. *Z15*.

Per impostare un conteggio occorre dare le seguenti istruzioni:

- Immettere il segnale di ingresso che dia avvio al conteggio.  
Se per esempio l'avvio al conteggio è dato dall'ingresso *E0.1*, si scriverà:  
*U E0.1*

- Caricare il valore del conteggio.  
Questo si ottiene come per il caricamento del temporizzatore ponendo in ordine:  
L        Codice operativo di caricamento  
KZ      Operando che determina il caricamento di un valore costante di conteggio.  
Numero   È il parametro corrispondente al valore di conteggio  
Esempio:

```
L KZ 8
```

Viene caricato il valore di conteggio uguale a 8.

- Con la funzione *S* si setta un contatore sul valore iniziale ( 8 nell'esempio ). Così se il contatore scelto è *Z1* si scrive:

```
S Z1
```

- Si deve settare l'ingresso che determina ad ogni sua attivazione un decremento del contatore. Sia ad esempio l'ingresso *E 0.21*, si scriverà:  
*U E0.2*

- Si scrive l'istruzione *ZR* di decremento di una unità del valore impostato e trasferito al contatore *Z1*:

```
ZR Z1
```

- Si interroga lo stato di *Z1*, il quale è allo stato logico *1* fino a che il numero di decrementi è pari al valore impostato: 8 nell'esempio.

```
U Z1
```

- Si assegna il valore logico del contatore *Z1* ad una uscita o ad un *Merker*. per esempio sia assegnato all'uscita *A2.0*:

```
=A2.0
```

Per provare il programma impostato si colleghi l'ingresso  $E0.1$  ad uno *Start* con aggancio meccanico, l'ingresso  $E0.2$  ad un pulsante e l'uscita  $A2.0$  ad una lampada.

Nella tabella si riassumono le occupazioni con l'indicazione del contatore utilizzato.

Elemento	Simbolo	Operando
Start	$S_1$	$E0.1$
Pulsante	$S_2$	$E0.2$
Contatore		$Z1$
<i>Merker</i>	$H_1$	$M1.0$
Lampada	$L$	$A2.0$

Riassumendo il programma in lista di istruzioni:

$U E0.1$   
 $L KZ 8$   
 $S Z1$   
 $U E0.2$   
 $ZR Z1$   
 $U Z1$   
 $=A2.0$

Portato il programma dal *PG* all'*AG* e posto questo in Run si verifichi il funzionamento.

Attivato lo Start  $S_1$  si accende la lampada. Si preme 8 volte il pulsante  $S_2$ , all'ottavo impulso la lampada si spegne.

## IN LABORATORIO

### Tema

*Una piattaforma porta pezzi viene comandata da un cilindro a doppio effetto. Pigiato uno Start la piattaforma si solleva portandosi ad un livello controllato da un finecorsa. Nella posizione raggiunta, vengono posizionati sulla piattaforma dei pezzi e ad ogni posizionamento esatto viene pigiato un pulsante  $P$  di accettazione.*

*Dopo il quarto azionamento del pulsante  $P$ , automaticamente la piattaforma si riporta al livello inferiore controllato da un finecorsa, dove verranno smontati i pezzi.*

Il cilindro a doppio effetto  $A$  viene comandato da una valvola di potenza monostabile. Viene impiegato un contatore  $Z_1$  che dopo quattro impulsi effettua il rientro dello stelo del cilindro  $A$  doppio effetto

Il contatore  $Z_1$  viene settato e portato allo stato logico 1 dall'azionamento del finecorsa  $a_1$

$$Z_1 = a_1$$

Il decremento del contatore è determinato dal pulsante di accettazione  $P$ .

$$Dec = P$$

Con queste precisazioni il comando della valvola monostabile è data:

$$A^+ = a_0 \cdot Start$$

$$A^- = \overline{Z_1} \cdot a_1$$

Il comando  $Y_1$  alla bobina dell'elettrovalvola sarà:

$$Y_1 = (a_0 \cdot Start + Y_1) \cdot a_1 \cdot \overline{Z_1}$$

Si pone:

$$H_1 = \overline{Z_1} \cdot a_1$$

Per cui:

$$Y_1 = (a_0 \cdot Start + Y_1) \cdot \overline{H_1}$$

Lo stato logico del contatore  $Z_1$  viene assegnato ad un Mercher  $C$

$$C = Z_1$$

Riunendo i segnali di comando:

$$Y_1 = (a_0 \cdot Start + Y_1) \cdot \overline{H_1}$$

$$Z_1 = a_1$$

$$Dec = P$$

$$C = Z_1$$

$$H_1 = \overline{Z_1} \cdot a_1$$

Occupazioni

Elemento	Simbolo	Canale	Morsetto
Start	<i>Start</i>	<i>E1.0</i>	4
Finecorsa	$a_0$	<i>E1.1</i>	3
Finecorsa	$a_1$	<i>E1.2</i>	6
Merker $H_1$		<i>M1.0</i>	
Merker		<i>M1.1</i>	
Elettrovalvola	$Y_1$	<i>A2.0</i>	4

Il programma in lista di istruzioni è il seguente

Equazione logica	Lista istruzioni
$Y_1 = ((Start \cdot a_0) + Y_1) \cdot \overline{H_1}$	<i>U(</i>
	<i>O(</i>
	<i>U E1.0</i>
	<i>U E1.1</i>
	<i>)</i>
	<i>O. A2.0</i>
	<i>)</i>
	<i>U N</i>
	<i>M1.0</i>
	<i>= A2.0</i>
contatore $Z_1 = a_1$ $Dec = P$ $C = Z_1$	<i>U E1.2</i>
	<i>LK Z1</i>
	<i>U E1.3</i>
	<i>ZR Z1</i>
	<i>U Z1</i>
$H_1 = \overline{Z_1} \cdot a_1$	<i>= M1.1</i>
	<i>U E1.2</i>
	<i>U N</i>
	<i>M1.1</i>
	<i>= M1.0</i>

Trasferire il programma nell'AG è provare. Pigiato lo Start fuoriesce lo stelo. Si pigi più volte il tasto  $P$ : alla quarta attivazione rientra lo stelo.

*Temi proposti da effettuare con i PLC con i diversi programmi in dotazione del laboratorio.*

1°

*Una serie di pezzi debbono subire una marchiatura. Essi vengono prelevati da un raccogliitore e spinti nel sistema di bloccaggio.*

*Bloccato il pezzo viene effettuata la marchiatura. Eseguita questa, si sblocca il pezzo che viene tolto a mano dal sistema di bloccaggio.*

2°

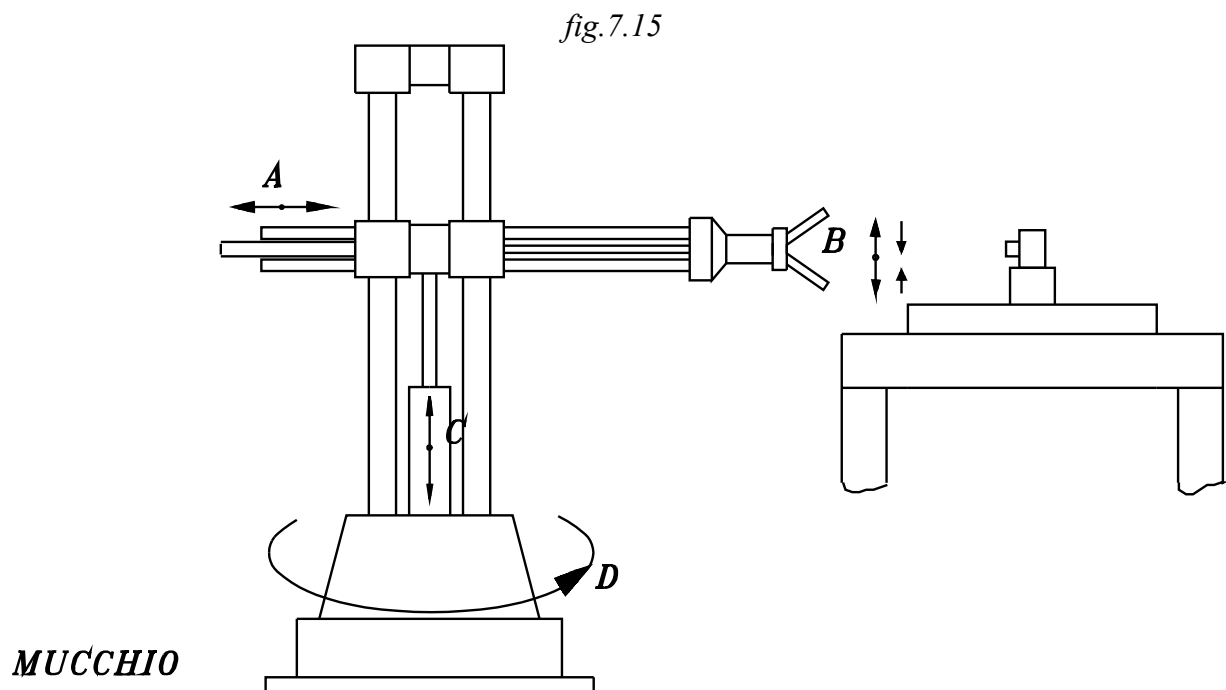
*Su ogni dischetto di una serie si deve eseguire una punzonatura.*

*I dischetti vengono montati a mano sull'apposito porta pezzo e vengono bloccati attraverso la corsa di un cilindro A. Un cilindro B, sul cui stelo è montato il punzone, effettua la doppia corsa di punzonatura. Eseguita questa, il cilindro A effettua la corsa di ritorno liberando il dischetto dal bloccaggio. Alla fine un cilindro C effettua una doppia corsa per espellere il dischetto dal portapezzo.*

3°

*Risolvere con il PLC il tema svolto nel 2° volume riguardante il robot sequenziale*

*Il robot deve prendere dei pezzi dal nastro trasportatore, trasportarli e lasciarli cadere nel mucchio.*



*La movimentazione avviene con le seguenti fasi:*

1°

*Il braccio è al livello del pezzo.*

*Quando questo, trasportato dal nastro, si trova in posizione di presa, si ferma il nastro trasportatore e viene dato il segnale S di inizio del ciclo. Il cilindro A fa*



*fuoriuscire il braccio ( $A^+$ ), portando la pinza in posizione di presa. Un sensore di prossimità  $u$  meccanico rileva la corretta posizione del pezzo.*

- 2° *Quando il braccio ha raggiunto il finecorsa  $a_1$  e si ha il segnale di corretta posizione del pezzo:  $u=1$ , allora viene chiusa la pinza attraverso il cilindro B ( $B^+$ ).*
- 3° *Toccato il finecorsa  $b_1$  viene comandato il cilindro C, il quale fa fuoriuscire lo stelo ( $C^+$ ), alzando il pezzo dal nastro trasportatore.*
- 4° *Toccato il finecorsa  $c_1$ , viene comandato il cilindro D, il quale fa fuoriuscire lo stelo ( $D^+$ ). Il moto rettilineo dello stelo viene tramutato in moto rotatorio del montante del robot attraverso un meccanismo di rocchetto dentiera.*
- 5° *Effettuata una rotazione di  $180^\circ$  e toccato il finecorsa  $d_1$ , viene comandato il cilindro C ad abbassarsi in posizione di deposizione del pezzo nel pallet ( $C^-$ ).*
- 6° *Toccato il finecorsa  $c_1$  e rilevata la giusta posizione del pezzo attraverso un sensore di prossimità  $w$ , si apre la pinza con rientro del cilindro B ( $B^-$ ).*
- 7° *Toccato  $b_0$  si ritrae il braccio ( $A^-$ )*
- 8° *Toccato  $a_0$  il robot ruota di  $180^\circ$  in senso inverso al precedente, rientra il cilindro D ( $D^-$ ) fino a toccare il finecorsa  $d_0$ .*

4° *Un foglio di lamiera deve essere profilato con deformazione a freddo. Il foglio viene posto a mano sull'apposita controstampo. Pigiati due Start il foglio viene profilato dall'apposito stampo comandato da un cilindro pneumatico. Effettuato il ciclo di profilatura un espulsore comandato da un altro cilindro espelle il foglio profilato.*

*Condizioni al contorno*

*Debbono essere impiegati due start per impegnare entrambi le mani . Deve essere previsto un pulsante Em di emergenza, attivando il quale i cilindri debbono tornare nella posizione di rientro e non risultano attivi gli Start.*

5° *Per effettuare la piegatura del foglio di cui al tema 4°, l'impianto deve essere modificato come segue.*

- *Il controstampo su cui viene depositato il foglio di lamiera, nelle condizioni iniziali è al di fuori di una zona protetta non accessibile dalla mani dell'operatore.*
- *Posto il foglio sul controstampo e attivati i due start, attraverso la corsa di un cilindro il controstampo si porta nella posizione di piegatura posta nella zona protetta.*
- *Viene quindi effettuata la profilatura del foglio attraverso uno stampo comandato da un altro cilindro pneumatico.*
- *Effettuata la piegatura il controstampo si porta di nuovo nella posizione iniziale ove un espulsore al disotto del foglio profilato lo espelle.*



Avanti...

[Clic qui per proseguire](#)



Indietro...

[clic per precedente](#)



Indietro...

[Clic qui per la pagina iniziale](#)