

PROTEUS ES

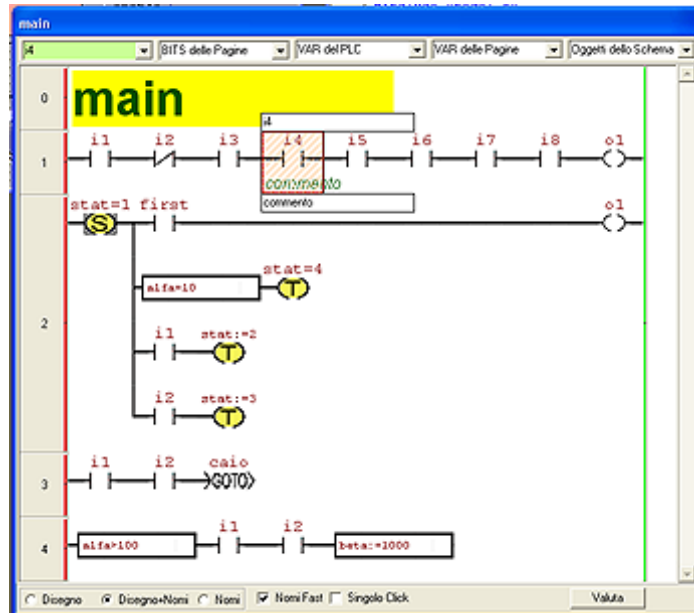
Manuale di programmazione

Volume 2

**Programmazione del PLC in linguaggio LD
Conforme allo standard IEC1131-3 LD**

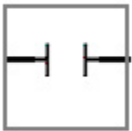
Funzioni di Edit nella finestra del PLC

Cliccando con il mouse sul bottone PLC della barra in alto, si aprirà la finestra di programmazione in linguaggio LADDER



Aperto la finestra del PLC appare anche la **barra degli strumenti** relativi. Con questa barra possiamo selezionare il componente da inserire nello schema.

Quando il cursore è dentro la finestra del PLC, un **quadrato grigio** indica la posizione di edit attuale. Se non è selezionato il campo SINGOLO CLICK possiamo piazzare il componente con un doppio click del mouse, altrimenti basta un singolo click (sconsigliato).



Nel quadrato attorno al cursore appare, in un angolo, il **simbolo del componente scelto**. Se quest'ultimo è barrato in **rosso** significa che non può essere piazzato nella posizione attuale.

Per **cancellare** un componente, posizionare il cursore su di esso, e quindi premere il tasto **DEL**. Se il componente è una linea verticale, per cancellarla occorre selezionare il componente linea verticale, altrimenti non riusciremo a piazzare il cursore sulla linea stessa, visto che si trova tra due celle adiacenti.

Se siamo *in modo DISEGNO*, possiamo piazzare velocemente i com-

ponenti, senza assegnare i nomi.

Se siamo *in modo DISEGNO+NOMI*, una volta piazzato un componente, appare la finestra di edit per assegnare il nome (in alto) ed eventualmente un commento (in basso). Se non vogliamo assegnare subito il nome, possiamo uscire dall' edit con un click del mouse. Potremo assegnarlo quando vogliamo, selezionando come componente la freccia, e cliccando sulla casella dello schema a cui vogliamo assegnare o cambiare il nome.

In modo NOMI la barra degli strumenti scompare e possiamo assegnare o cambiare i nomi ed i commenti ai componenti.

Il nome può essere battuto direttamente da tastiera, oppure cercato nei menu a discesa in alto nella finestra, oppure può essere assegnato cliccando nella finestra grafica sul componente relativo.

Per inserire una nuova riga vuota tra due preesistenti occorre un doppio click a sinistra sulla barra grigia con i numeri delle celle.

Per cancellare una riga, selezionarla tutta da sinistra a destra (dalla barra rossa a quella verde) e quindi premere il tasto **DEL**. Nella pagina del PLC possiamo selezionare più celle, tenendo premuto il bottone sinistro del mouse, e possiamo effettuare operazioni di **taglia-incolla** anche di intere aree, con i tasti **CONTROL+C** e **CONTROL+V**.

Con il **bottone VALUTA** possiamo valutare la correttezza dello schema, Se viene individuato un errore (una cella senza nome, oppure una connessione mancante) viene segnalato l' errore e il cursore, tratteggiato in rosso, si porta sulla cella dove c' è l' errore. Nel caso in cui l' errore sia in una SUB, la pagina visualizzata sarà quella della SUB in cui è stato riscontrato l'errore. La valutazione si ferma al *primo errore* trovato. Per continuare, occorre correggere l' errore, chiudere la valutazione premendo ancora il bottone VALUTA, e quindi premere ancora il bottone VALUTA per una nuova scansione dei files del progetto. La valutazione parte sempre dall' inizio del programma principale, indipendentemente dalla pagina visualizzata. Per un corretto editing, è meglio richiudere la finestra di valutazione degli errori, premendo ancora sul bottone VALUTA.

Da notare che un singolo componente (un filo orizzontale od un contatto) in prima colonna, non connesso ad altro, non genera errore. Questo per consentire certe funzioni speciali di debug, che vedremo dettagliatamente nel manuale del debug.

Per **scorrere verticalmente lo schema**, utilizzare la **barra di scorrimento** a destra. La barra a sinistra mostra il numero della cella. Una cella può essere composta da 1 a 8 linee.

Una linea può contenere fino a 8 contatti più la bobina. Per linee più lunghe utilizzare il simbolo di **continuazione**, che permette di continuare la linea su un'altra cella.

I **simboli utilizzabili** sono quelli standard della programmazione a contatti (contatti, bobine, monostabili, derivatori, ritardi, logic blocks, linee verticali ed orizzontali, goto, sub, labels, return) più i simboli di STATO e TRANSIZIONE che permettono di implementare macchine a stato.

Alcuni tasti hanno funzioni particolari:

DEL o CTRL+Z permette di cancellare la casella di un oggetto, nel caso lo volessimo sostituirlo con un altro (si possono piazzare oggetti solo in caselle vuote o contenenti un filo orizzontale)

CTRL+X come DEL, ma il componente attuale diviene quello cancellato (equivale a battere un DEL e dopo andare a scegliere nella barra dei componenti del PLC quello appena cancellato)

CTRL+C il componente attuale diviene quello che si trova sotto il cursore.

CTRL+V inserisce nella casella del cursore il componente selezionato (equivale ad un doppio click).

SPACE trasla a destra la riga attuale a partire dal cursore.

INSERT trasla in basso le righe della cella attuale a partire dal cursore.

BACKSPACE annulla ultima operazione (undo).

F3 cerca una casella che abbia lo stesso nome di quella sotto il cursore. **Se il cursore è su una SUB apre lo schema che contiene la SUB. Se il cursore è su un return**, oppure se è su una casella bianca e siamo in una SUB, **ritorna allo schema precedente.**

La maniera più semplice per creare un nuovo schema (cioè una nuova pagina di PLC, ovvero una SUB), consiste nell'aggiungere allo schema attuale una funzione SUB, dargli un nome e quindi battere il tasto F3. Il

nuovo schema si aprirá automaticamente.

Le **SUB** possono essere annidate recursivamente una dentro l'altra, fino ad un livello di 100.

Ricordiamo che una funzione SUB deve essere l' ultima della propria linea, anche se non è in colonna 8.

Per passare da uno schema all' altro possiamo anche usare *il **tasto Apri.***

I tasti **CTRL+X**, **CTRL+z**, **CTRL+C** e DEL possono essere usati per **cancellare e spostare intere aree** dello schema, selezionate tenendo premuto il bottone sinistro del mouse.

Regole generali

Ogni schema è composto da un certo numero di BLOCCHI.

Ogni blocco è costituito da una o più linee unite tra loro da collegamenti verticali. Il numero massimo di linee di un blocco è OTTO.

Ogni linea è composta da un massimo di nove celle (8 contatti + una bobina).

Un singolo schema può contenere fino a 999 linee.

Schemi diversi sono collegati tra loro mediante SUB e RETURN.

Un sottoschema può contenere dei SUB. In questo modo si può strutturare uno schema "a piramide".

Un blocco non può contenere fili od altri elementi scollegati, e ogni linea deve iniziare dalla barra di potenziale (linea rossa a sinistra) oppure da un collegamento verticale dalla linea precedente della cella, e deve terminare con un collegamento alla linea superiore del blocco oppure con una bobina, oppure con un altro elemento che non prevede necessariamente una continuazione (GOTO, RETURN, SUB, Blocco Funzione, Continuazione)

Ogni cella (salvo alcune, come i derivatori, i RETURN ecc..) ha due campi, uno sopra la cella ed uno sotto la stessa.

Nel **campo superiore si scrive il nome della variabile**, nel campo inferiore si può scrivere un commento.

L'ordine delle celle non è importante. I cambiamenti dello stato delle bobine è effettuato alla fine del ciclo del PLC.

Non è consentito utilizzare la funzione RETURN nello schema principale (MAIN)

Le funzioni GOTO e RETURN vengono sempre eseguite per ultime, indipendentemente dalla posizione in cui si trovano nel blocco, per cui è garantito che il blocco esegua correttamente tutte le funzioni che contiene prima di uscire.

Evitare di usare nello stesso blocco più funzioni GOTO che possono verificarsi contemporaneamente.

Le variabili del PLC

Variabili predefinite::

1- I registri del PLC:

<code>first_cycle</code>	vale 1 solo durante il primo ciclo
<code>is_running</code>	vale 1 nei cicli successivi.
<code>first</code>	vedi macchine a stati
<code>MB1...MB1000</code>	registri di tipo byte
<code>MW1...MW1000</code>	registri di tipo word
<code>MD1...MD1000</code>	registri di tipo doppio word
<code>MF1...MF1000</code>	registri flottanti

2- Le variabili di i/o

<code>i1,i2...i128</code>	Variabili di ingresso
<code>o1,o2... o128</code>	Variabili di uscita
<code>anal1...anal16</code>	uscite analogiche
<code>pot1..pot32</code>	ingressi analogici
<code>encoder1...encoder16</code>	contatori encoders

3- Tutti i **nomi assegnati alle bobine.**

4- Tutti i **nomi degli oggetti** dei form grafici

Altre variabili:

Per i contatti e per le bobine del PLC è possibile inoltre usare **qualsiasi nome di variabile precedentemente definita** nella sezione del linguaggio C (in Common.h).

I Componenti del PLC



Contatto normalmente aperto

Rappresenta il contatto normalmente aperto di un relè.

Il nome che appare sopra il contatto è quello della bobina relativa. Tale nome può essere il nome di un ingresso, di una uscita, di un oggetto della finestra grafica, oppure il nome di una bobina (memoria) definita altrove nello schema. Per i nomi delle bobine non ci sono particolari limitazioni, anche se si consiglia di usare per chiarezza la terminologia classica (R1, R2... o M1, M2 ...).

Più contatti normalmente aperti in serie implementano la funzione AND.

Più contatti normalmente aperti in parallelo implementano la funzione OR.



Contatto normalmente chiuso

Rappresenta il contatto normalmente chiuso di un relè.

Il nome che appare sopra il contatto è quello della bobina relativa. Tale nome può essere il nome di un ingresso, di una uscita, di un oggetto della finestra grafica, oppure il nome di una bobina definita altrove nello schema. Per i nomi delle bobine non ci sono particolari limitazioni, anche se si consiglia di usare per chiarezza la terminologia classica (R1, R2... o M1, M2 ...).

Più contatti normalmente chiusi in serie implementano la funzione NAND.

Più contatti normalmente aperti in parallelo implementano la funzione NOR.

Derivata positiva



Questo contatto dá in uscita un livello 1 per un solo ciclo di PLC ogni volta che il suo livello di ingresso (terminale di sinistra) passa da 0 ad 1.

 **Derivata negativa**

Questo contatto dá in uscita un livello 1 per un solo ciclo di PLC ogni volta che il suo livello di ingresso (terminale di sinistra) passa da 1 a 0.

 **Ritardatore**

Questo contatto dá in uscita il livello di ingresso ritardato di un tempo fisso. Il tempo è il valore sopra il ritardatore, ed è espresso in millisecondi.

 **Ritardatore positivo**

Questo contatto dá in uscita il livello di ingresso ritardando di un tempo fisso la transizione da 0 ad 1 dello stesso. Il tempo è il valore sopra il ritardatore, ed è espresso in millisecondi.

 **Ritardatore negativo**

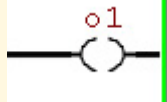
Questo contatto dá in uscita il livello di ingresso ritardando di un tempo fisso la transizione da 1 a 0 dello stesso. Il tempo è il valore sopra il ritardatore, ed è espresso in millisecondi.

 **Monostabile positivo**

Questo contatto dá in uscita il livello 1 per un tempo fisso ogni volta che vi è la transizione da 0 a 1 dell' ingresso. Il tempo è il valore sopra il monostabile ed è espresso in millisecondi.

 **Monostabile negativo**

Questo contatto dá in uscita il livello 1 per un tempo fisso ogni volta che vi è la transizione da 1 a 0 dell' ingresso. Il tempo è il valore sopra il monostabile ed è espresso in millisecondi.

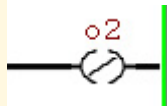


Bobina normalmente aperta

Una bobina è sempre l' elemento piú a destra di una linea di PLC. Di norma si trova in colonna n. 9, e rappresenta la bobina di un relé. Il suo simbolo associato è scritto sopra la stessa, e può essere un' uscita del PLC, oppure una variabile interna del PLC (memoria)

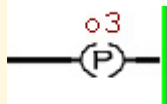
Per i nomi delle bobine non ci sono particolari limitazioni, anche se si consiglia di usare per chiarezza la terminologia classica (R1, R2... o M1, M2 ...).

La bobina normalmente aperta assume valore 1 (eccitata) quando il suo ingresso (terminale di sinistra) ha valore 1. Il suo terminale di destra va sempre alla barra di massa (linea verde).



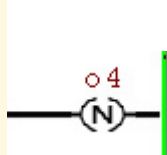
Bobina normalmente chiusa

La bobina normalmente chiusa assume valore 1 (eccitata) quando il suo ingresso (terminale di sinistra) ha valore 0. Il suo terminale di destra va sempre alla barra di massa (linea verde).



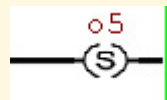
Bobina impulso positivo

La bobina impulso positivo assume valore 1 (eccitata) per un solo ciclo di PLC quando il suo ingresso (terminale di sinistra) passa da valore 0 a valore 1. Il suo terminale di destra va sempre alla barra di massa (linea verde). Equivale ad un derivatore positivo seguito da una bobina normalmente aperta.



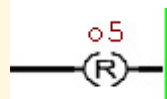
Bobina impulso negativo

La bobina impulso negativo assume valore 1 (eccitata) per un solo ciclo di PLC quando il suo ingresso (terminale di sinistra) passa da valore 1 a valore 0. Il suo terminale di destra va sempre alla barra di massa (linea verde). Equivale ad un derivatore negativo seguito da una bobina normalmente aperta.



Bobina Set

La bobina Set assume valore 1 (eccitata) quando il suo ingresso (terminale di sinistra) passa da valore 0 a valore 1 e conserva tale stato anche se l'ingresso torna a valore 0 (relé a memoria). Il suo terminale di destra va sempre alla barra di massa (linea verde).



Bobina Reset

La bobina Reset assume valore 0 (diseccitata) quando il suo ingresso (terminale di sinistra) passa da valore 0 a valore 1 e conserva tale stato anche se l'ingresso torna a valore 0 (relé a memoria). Il suo terminale di destra va sempre alla barra di massa (linea verde). Se la bobina Set e la bobina Reset con lo stesso nome hanno entrambe valore di ingresso 1 vale la condizione della bobina piú in basso nello schema (quella con il numero di cella piú alto).

Oltre a queste funzioni, che rappresentano la base per lo sviluppo di uno schema di PLC con programmazione a contatti, vi è un certo numero di funzioni ausiliarie che consentono di trattare grandezze analogiche e di estendere e semplificare la stesura dello schema.



Blocco condizionale

Un blocco condizionale consiste in un rettangolo al cui interno è scritta una condizione. Se questa è vera il blocco equivale ad un contatto chiuso, altrimenti ad un contatto aperto.

La condizione può contenere una qualsiasi uguaglianza o disuguaglianza tra variabili del PLC. Sono ammesse le seguenti disuguaglianze:

=	uguale
!=	diverso
<	minore
>	maggiore
<=	minore o uguale
>=	maggiore od uguale



Blocco funzionale

Un blocco funzionale consiste in un rettangolo al cui interno è scritto un comando. Se l'ingresso (il terminale a sinistra) è a livello 1, questo comando verrà eseguito.

Il comando deve essere del tipo: **VARIABILE := ESPRESSIONE**.

Si distingue dal blocco condizionale per la presenza del simbolo

: = assegnazione.

E' buona norma, anche se non tassativo, che un blocco funzionale sia l'ultimo elemento della propria linea di PLC.

→GOTO> GOTO – Label

`label1` Quando si trova un GOTO, se il suo ingresso è attivo, ovvero a livello 1, il ciclo del PLC prosegue a partire dalla linea che inizia con la label il cui nome è uguale a quello assegnato (che si trova sopra) il GOTO.

Label:

Tale linea deve fare parte dello stesso schema del PLC.

Per evitare condizioni imprevedibili è bene che il blocco che contiene il GOTO non contenga altri GOTO che possono essere soddisfatti contemporaneamente, in quanto, non essendone prevedibile l'ordine di esecuzione all'interno del blocco, non si saprebbe quale GOTO viene eseguito realmente.

SUB – RETURN

←SUB>

Quando si trova un SUB, se il suo ingresso è attivo, ovvero a livello 1, il ciclo passa allo schema il cui nome è uguale a quello assegnato (che si trova sopra) alla cella SUB.

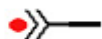
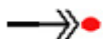
←RETURN>

Il ciclo riprenderà dal blocco successivo quando, nel sottoschema richiamato, si troverà una cella RETURN il cui ingresso (terminale di sinistra) è a livello alto. In un sottoschema possono esservi più RETURN. La funzione RETURN verrà comunque eseguita per ultima, dopo aver eseguito tutte le altre celle del blocco

Per creare od aprire un sottoschema, basta dare un nome alla SUB, posizionare il cursore su di essa e premere F3.

Per tornare allo schema precedente battere F3 sul RETURN.

CONTINUA

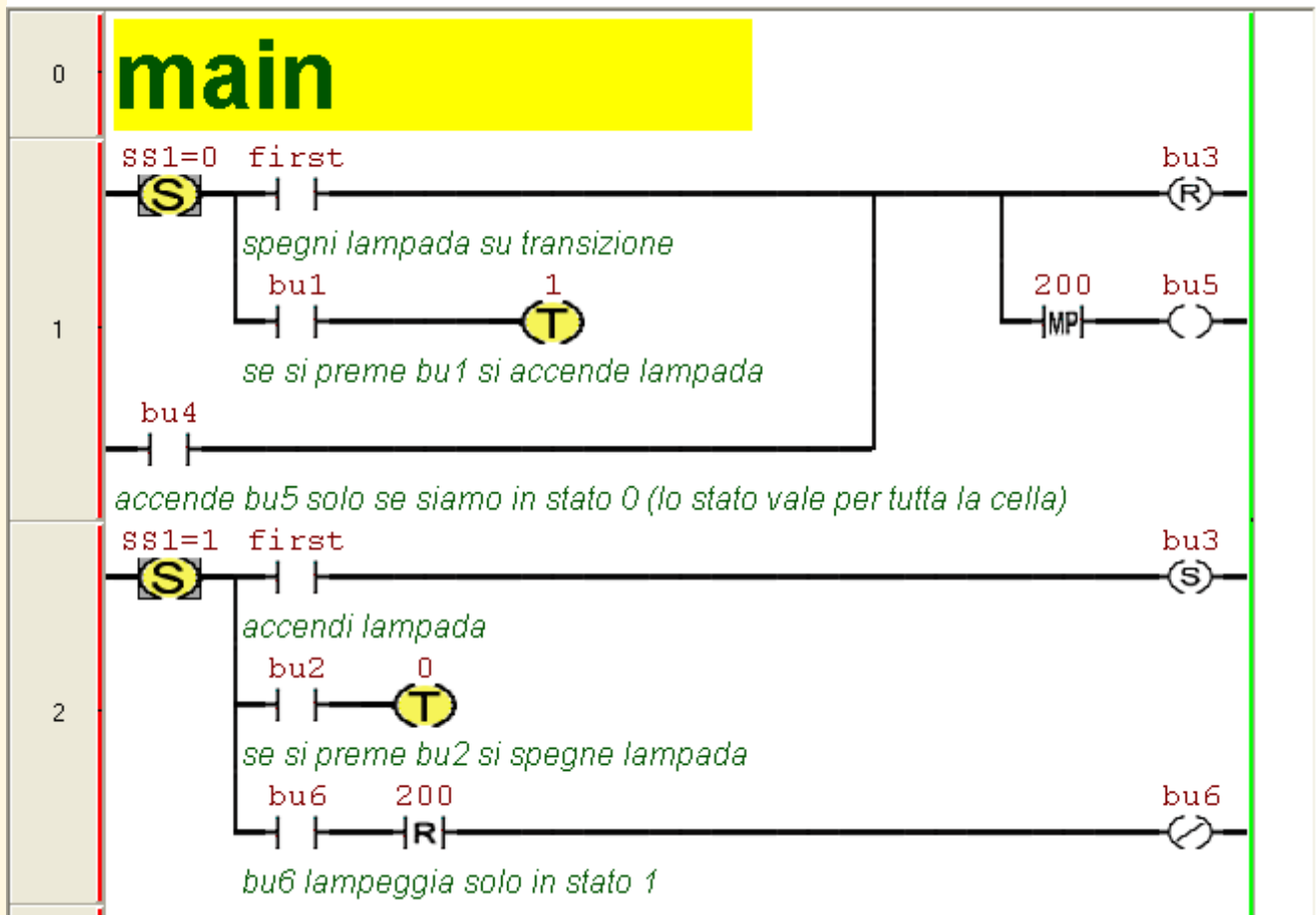


Serve ad estendere una linea che abbia più di otto contatti.

Quando si trova il simbolo CONTINUA la linea prosegue fisicamente a partire dal primo simbolo di rimando che si trova sulle linee successive.

Le celle CONTINUA-RIMANDO non hanno nomi, e sono associate due a due dall'alto verso il basso dello schema (il primo Continua col primo Rimando ecc...)

STATI e TRANSIZIONI



Mediante i componenti **stato** e **transizione** e' possibile implementare una **macchina a stati**.

La **cella STATO** (contatto) e' identificata da un nome a piacere, seguito dal carattere UGUALE (=) seguito dal valore dello stato.

Il nome della macchina è il nome dello stato per cui tutti i blocchi che contengono una cella STATO con lo stesso nome fanno parte della stessa macchina.

Esempio: **SS1=1** macchina a stati SS1, caso in cui lo stato vale 1
SS1=24 macchina a stati SS1, caso in cui lo stato vale 24

In altre parole, una macchina a stati non è altro che l'insieme di tutti i blocchi che fanno riferimento alla stessa variabile di stato.

Il componente TRANSIZIONE (bobina) e' identificato dal **valore** dello stato a cui passare se ci si riferisce alla macchina a stati correnti (transizione implicita) , oppure da **STATO = valore** se si vuole cambiare lo stato di un' altra macchina a stati (transizione esplicita) E' possibile usare la transizione esplicita anche per la macchina corrente.

Esempio: **1** **transizione implicita** a stato 1

Oppure **SS1=1** idem come sopra, ma **transizione esplicita**

Per i valori da assegnare agli stati e alle transizioni, nessuno vieta di utilizzare delle variabili al posto dei numeri.

Si possono così creare macchine a stati automodificabili, anche se in tale modo è molto facile perdere il controllo di cosa si sta realmente facendo.

Esempio: **MB12** **transizione implicita** a stato MB12

Oppure **SS1=MB6** idem come sopra, ma **transizione esplicita**

Regole di implementazione di una macchina a stati:

- Una macchina può assumere uno **stato** compreso tra **0 e 255**.
- Allo start del PLC lo stato di tutte le macchine è **ZERO**.
- Un blocco che inizia con una cella STATO e' **eseguito** solo quando lo stato della macchina è **uguale al valore associato allo stato**.
- In un blocco, l'eventuale **cella stato** deve essere **una e una sola, in colonna 1**, non può essere preceduta da contatti né da altro, e deve essere inoltre l'unica cella in colonna 1 nel blocco.
- Il **contatto first** e' chiuso solo la prima volta che la macchina **entra in un nuovo stato e durante il primo ciclo** del PLC
- Un blocco contenente una cella STATO può avere **una, nessuna o più celle di transizione**, purché non in contraddizione tra loro.
- Un blocco contenente una cella STATO può avere **celle di transizione esplicite** per stati di **altre macchine** della stessa pagina o di altre pagine.
- Un **blocco NON contenente celle di stato** può comunque contenere **transizioni**, ma in questo caso devono essere transizioni esplicite.
- Affinché una macchina a stati non si fermi indefinitamente in uno stato ... morto, in genere occorre che **per ogni stato** esista almeno una condizione di **transizione**.
- Non e' necessario che i blocchi di stato siano in ordine, né che siano contigui, né che siano nella stessa pagina, anche se ciò è consigliato per chiarezza e leggibilità. *Si può comunque scorrere i vari blocchi della stessa macchina a stati mediante il tasto di funzione F3.*
- Possono essere definite **più blocchi**, non necessariamente contigui, né necessariamente nella stessa pagina, per lo **stesso stato** della **stessa macchina**. Il risultato equivale all'unione di tutti tali blocchi.
- Non c'è un limite al numero di macchine a stati utilizzabili nel ciclo di PLC. L'unico limite è dato dalla dimensione della memoria.
- Un blocco che contiene una cella di STATO non deve contenere celle

di tipo GOTO, né RETURN, né CONTINUA.

Nel caso in cui per un blocco contenente una cella di stato non fossero sufficienti 8 linee, si può aggiungere un altro blocco contenente la stessa cella di stato. In tal caso i blocchi saranno fusi in fase di compilazione.