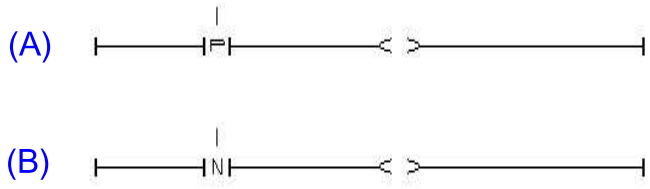
Appendice B

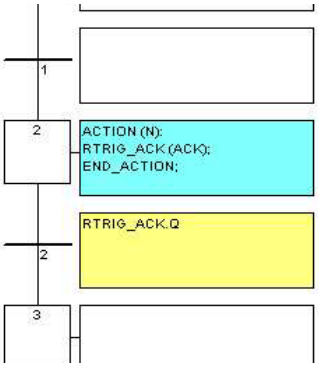
Note sull'utilizzo di ISAGRAF V. 3.31

B.1 Lettura del fronte di salita di una variabile

Nel linguaggio Ladder è sufficiente settare il contatore come in figura B.1(A) per leggere il fronte di salita della variabile di ingresso I; il contatto in figura B.1(B) legge invece il fronte di discesa della variabile I.



In linguaggio SFC la lettura risulta più complessa. Data la variabile da acquisire ACK (definita naturalmente come booleana nel *dizionario*) va definita anche la relativa variabile di servizio nelle **istanze** BF del dizionario (per esempio $RTRIG\_ACK$).



A questo punto il formalismo prevede un passo per inizializzare le variabili ed una transizione con il test sul fronte di salita.

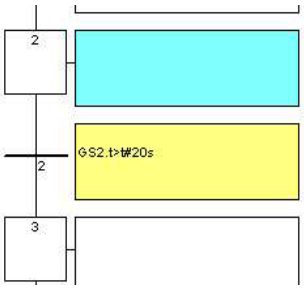
Per il fronte di salita la variabile di servizio va definita come tipo R_TRIG. Se si volesse leggere il fronte di discesa bisognerebbe definirla come F_TRIG.

B.2 Blocco del PC in simulazione

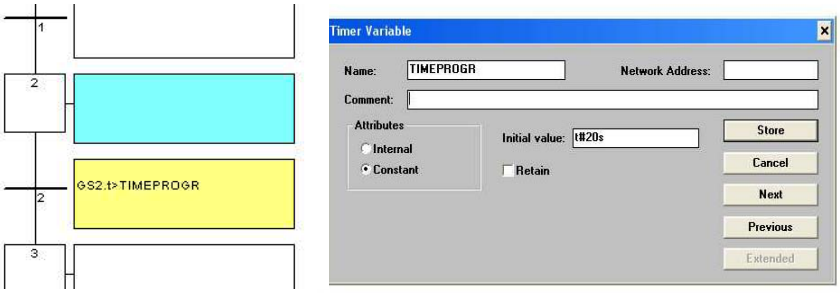
Accade che alcuni PC si blocchino in uscita quando si apre anche l'editor per monitorare l'evoluzione del programma. In tal caso bisogna *chiudere per primo l'editor, poi la finestra della simulazione.*

B.3 Variabili timer

Ogni volta che si vuole usare una variabile per un tempo da impostare, bisogna definire tale variabile come *timer* nel dizionario. In tal modo tale variabile si può usare in ingresso ai temporizzatori Ladder, oppure nelle transizioni temporizzate dell'SFC.



oppure



B.4 Azioni booleane

le azioni booleane assegnano ad una variabile booleana lo stato di attività del passo. La variabile booleana può essere interna o un output. Viene valorizzata ogni volta che viene avviata o fermata l'attività del passo.

Questa è la sintassi delle azioni booleane di base:

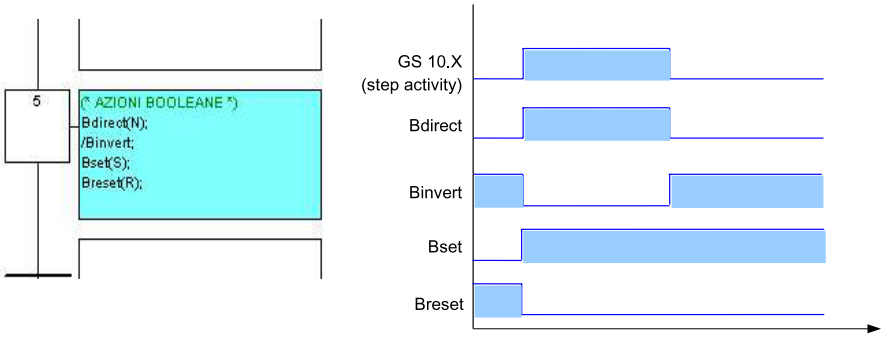
- <**variabile_booleana**> (**N**); assegna alla variabile lo stato di attività del passo;
- <**variabile_booleana**>; stesso effetto (l'attributo *N* è facoltativo);
- / <**variabile_booleana**>; assegna alla variabile l'inverso dello stato di attività del passo.

Esistono altre possibilità di modificare il valore della variabile booleana all'attivazione del passo.

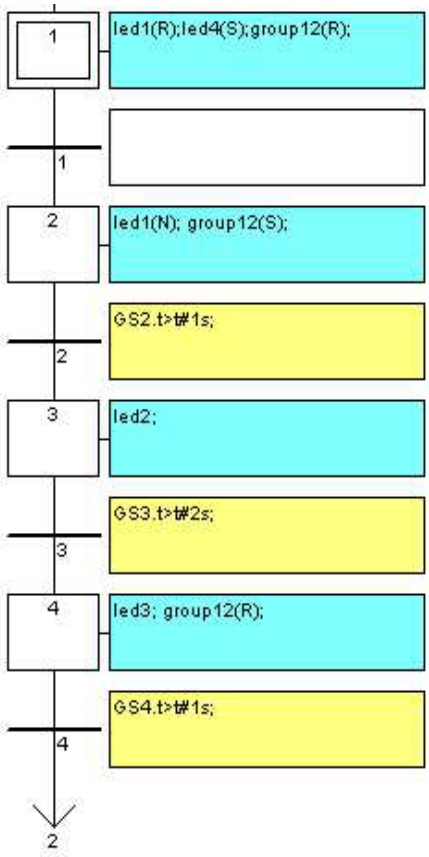
Questa è la sintassi delle azioni booleane di tipo ***set*** e ***reset***:

- <**variabile_booleana**> (**S**); imposta la variabile booleana a TRUE quando lo stato di attività del passo diventa TRUE;
- <**variabile_booleana**> (**R**); imposta la variabile booleana a FALSE quando lo stato di attività del passo diventa TRUE.

La variabile booleana deve avere ***attributo*** *di variabile di USCITA o INTERNA*. In figura si presenta il risultato della programmazione SFC:



Esempio di azioni booleane:
(*Programma SFC che usa azioni booleane*)



Funzionamento:
All'attivazione del passo 1, le variabili *led*1 e *group*12 si resettano e la variabile *led*4 si setta. Viene poi eseguito il passo 2, che pone a 1 la variabile *led*1, la quale permane a livello logico alto solamente se il passo 2 è in esecuzione. L'attivazione del passo 2 setta inoltre la variabile *group*12 (che era stata resettata al passo precedente). Prima che il sistema passi poi allo stato 3 viene atteso un tempo di 1s dettato dalla variabile temporale associata alla transizione 2. Trascorso 1s si attiva il passo 3 che assegna il proprio stato di attività alla variabile *led*2. Prima che il sistema passi poi allo stato 4 viene atteso un tempo di 2s dettato dalla variabile temporale associata alla transizione 3. Trascorsi 2s si attiva il passo 4 che assegna il proprio stato alla variabile *led*3 e resetta la variabile *group*12 (che era stata settata al passo 2). Il programma attende 1s prima di superare la transizione 4, poi riprende il ciclo di esecuzione dal passo 2.

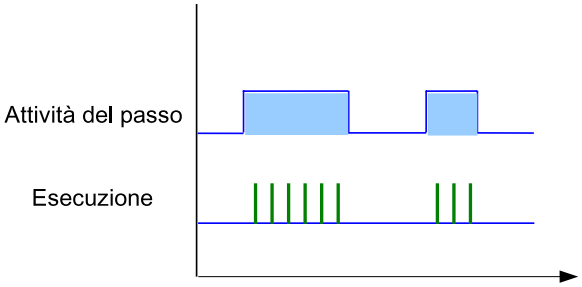
B.5 Azioni di tipo non-memorizzato

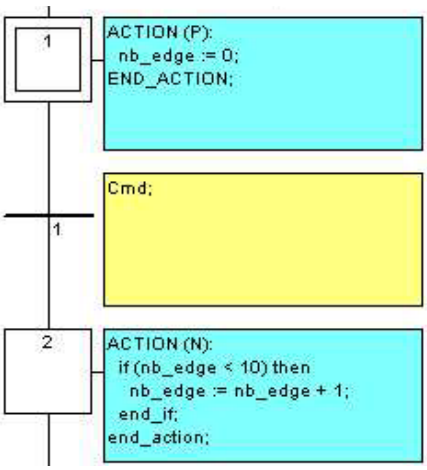
Un'azione di tipo non-memorizzato (normale) è costituita da un elenco di istruzioni ST o IL, che vengono eseguite **ad ogni ciclo** durante l'intero periodo di **attivazione** del passo. Le istruzioni rispettano la seguente sintassi SFC:

ACTION (N):
(* istruzioni ST *)
END_ACTION;

Nota: non confondere ACTION(N) con un'azione booleana Normale: questo significa che gli assegnamenti di variabili ***booleane*** con ACTION(N) sono equivalenti alle Azioni Booleane, ma con le Azioni Booleane ***non posso*** assegnare per esempio valori a *variabili numeriche* o *timer*!

Di seguito vengono illustrati i risultati di un'azione di tipo non-memorizzata e l'esempio di un'azione di tipo non-memorizzato:



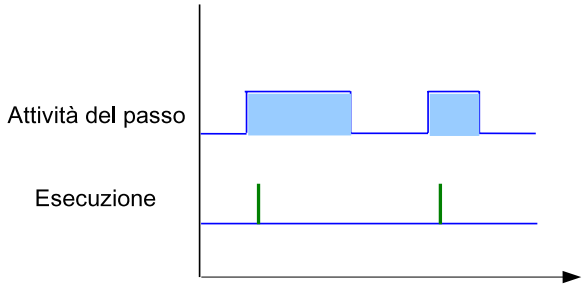


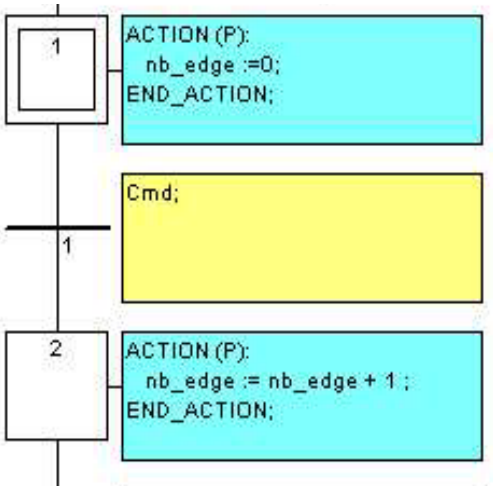
B.6 Azioni di tipo impulsivo

Un'azione di tipo impulsivo è costituita da un elenco di istruzioni ST o IL, che vengono eseguite solamente **una volta** al momento dell'**attivazione** del passo. Le istruzioni rispettano la seguente sintassi SFC:

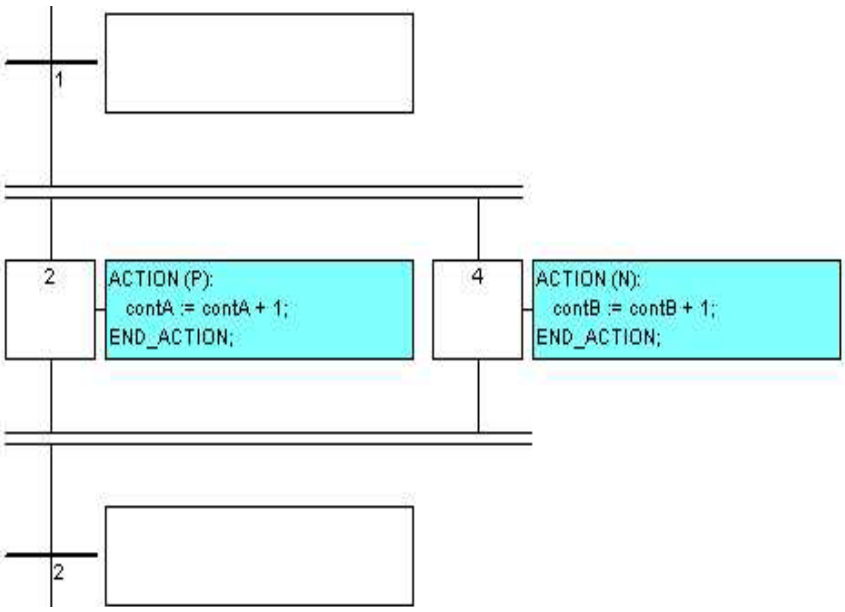
ACTION (P):
(* istruzioni ST *)
END_ACTION;

Di seguito vengono illustrati i risultati di un'azione di tipo impulsivo e l'esempio di un'azione di tipo impulsivo:





Differenza tra ACTION (N) e ACTION (P)



Per tutta la durata di attivazione dei passi indicati, *contA* incrementa di 1, *contB* incrementa a velocità di ciclo macchina.

B.7 Azioni SFC

Un'azione SFC è costituita da una sequenza figlia SFC, avviata o terminata in seguito al mutare dello stato di attività del passo. Un'azione SFC può venire qualificata come **N** (Non-memorizzata), **S** (Set) o **R** (Reset). Questa è la sintassi delle azioni SFC di base:

- <**propr_figlio**> (**N**); avvia la sequenza figlia al momento dell'attivazione del passo e termina la sequenza figlia quando il passo diventa inattivo;
- <**propr_figlio**>; stesso effetto (l'attributo *N* è facoltativo);
- <**propr_figlio**> (**S**); avvia la sequenza figlia al momento dell'attivazione del passo. Nessuna azione viene intrapresa quando il passo torna inattivo.
- <**propr_figlio**> (**R**); termina la sequenza figlia quando il passo diventa inattivo.

La sequenza SFC specificata come azione deve essere un **programma SFC figlio** del programma attualmente in fase di modifica. l'uso dei qualificatori **S** (Set) o **R** (Reset) in un'azione SFC, produce lo stesso effetto delle istruzioni programmate in un'azione **ST** di tipo impulsivo.

Segue sotto un'esempio di azione SFC. Il programma principale SFC viene chiamato **padre**: ha due figli SFC, chiamati **SeqMix** e **SeqPump**. La programmazione SFC per il programma SFC è la seguente:

(* Programma SFC che usa un'azione SFC *)