



*Manuale di Programmazione Siemens Step7 5.3
con Esercizi Svolti*

Professore
Ing. S. Cavalieri

Studente
D. Marassà

Introduzione	4
<i>Campi d'impiego del Simatic S7-300</i>	<i>4</i>
<i>STEP 7</i>	<i>4</i>
<i>Linguaggi di Programmazione</i>	<i>6</i>
<i>SIMATIC Manager</i>	<i>6</i>
1. Guida Simatic STEP7.....	9
1.1 Avvio di STEP 7	9
1.2 Programmazione con nomi simbolici	11
1.3 Indirizzi e tipi di dati ammessi nella tabella dei simboli	12
1.4 Creazione di un programma nell'OB1	13
1.5 Come configurare l'hardware	17
1.5.1 Creazione di stazioni.....	17
1.5.2 Operazioni fondamentali nella configurazione di stazioni.....	18
1.5.3 Definizione delle proprietà dei componenti	18
Avviamento	19
Ritenzione	20
1.5.5 Salvataggio della configurazione Hardware	21
1.6 Caricamento e test del programma.....	22
1.6.1 Come eseguire la cancellazione totale della CPU e commutarla in RUN	22
1.6.2 Come caricare il programma nella CPU	22
1.6.3 Come accendere la CPU e controllare lo stato di funzionamento	24
1.6.4 Come testare il programma controllandolo	24
1.6.5 Come testare con KOP	25
1.7 S7_PLCSIM	26
1.8 Simulazione con S7-PLCSIM.....	27
1.9 Come utilizzare STEP 7 per controllare la simulazione del programma	28
1.10 S7_GRAPH	28
1.10.1 Creazione di un progetto in S7-GRAPH.....	29
1.10.2 Programmazione di azioni.....	30
Azioni con funzioni standard	30
Azioni che richiamano blocchi.....	30
1.10.3 Programmazione di condizioni	31
Programmazione dei controlli.....	32
Operandi specifici di S7-GRAPH per condizioni	33
1.10.4 Salvataggio e chiusura della catena sequenziale.....	33
1.10.5 Inizializzazione della catena sequenziale e richiamo nel programma OB1	34
1.10.6 Caricamento del programma nella CPU e test della catena sequenziale.....	35
2 Applicazioni.....	36

2.1 Svolgimento del progetto - Trapano	36
2.1.1 Progetto Trapano realizzato in linguaggio KOP (Ladder)	39
2.1.2 Progetto Trapano realizzato in linguaggio SFC	50
2.1.3 Progetto Trapano realizzato in linguaggio SFC con la chiamata di funzione	59
2.2 Svolgimento del progetto - Nastri Trasportatori e Robot	72
2.2.1 Progetto dei Nastri Trasportatori e del Robot realizzato in linguaggio KOP (Ladder)	76
2.2.2 Progetto dei Nastri Trasportatori e del Robot realizzato in linguaggio SFC	88
2.3 Svolgimento del progetto - Distributore di Bibite	100
2.3.1 Progetto dei Nastri Trasportatori e del Robot realizzato in linguaggio KOP (Ladder)	103
2.3.2 Progetto del Distributore di Bibite realizzato in linguaggio SFC	112

Introduzione

I costruttori di macchine ed impianti fanno fronte sempre più spesso alla richiesta di una maggiore flessibilità e produttività a prezzi decrescenti. Ciò comporta nuove prestazioni dei sistemi di automazione, p. es. riguardo a set di funzioni, velocità di elaborazione e grandezza costruttiva ma anche in riferimento a engineering e possibilità di collegamento in rete. Sono richiesti controllori a logica programmabile (PLC) di piccole dimensioni, con un ampio set di funzioni ed elevate velocità di elaborazione. Contemporaneamente devono migliorare le possibilità di interconnessione in rete mentre i costi di engineering devono scendere.

Il SIMATIC® S7-300® è il controllore più venduto nell'ambito di Totally Integrated Automation, il cui successo è attestato da molte applicazioni di riferimento in tutto il mondo nei più svariati settori industriali. Gli utenti dell'S7-300 traggono quindi vantaggio dall'esperienza e dal service mondiale del leader di mercato nonché dalla qualità del SIMATIC.

L'S7-300 è concepito per soluzioni di sistema orientate in particolare alla tecnica manifatturiera e rappresenta, quale piattaforma di automazione universale, una soluzione ottimale per applicazioni in strutture centralizzate e decentralizzate. Grazie a continue innovazioni, questa piattaforma di automazione è sempre più potenziata. Esempi ne sono i sistematici sviluppi progressivi delle potenti unità centrali.

Campi d'impiego del Simatic S7-300

Il SIMATIC S7-300 offre soluzioni per i più svariati compiti di automazione nei seguenti settori:

- Tecnica manifatturiera
- Industria automobilistica
- Costruzione di macchine in genere
- Costruzione di macchine speciali
- Costruzione di macchine standard
- Lavorazione della plastica
- Industria del confezionamento
- Industria alimentare e dei generi voluttuari
- Tecnica di processo

La programmazione dell'S7-300 si effettua con il software di base **STEP 7**.

È così possibile sfruttare appieno, in modo facile e comodo, la potenza di questo PLC. Gli strumenti software comprendono le funzioni necessarie per tutte le fasi di sviluppo di un progetto di automazione, dalla progettazione fino alla messa in servizio, al test ed al service.

STEP 7

STEP 7 è il pacchetto di base utilizzato per configurare e programmare i sistemi di automazione SIMATIC. Esso rappresenta parte integrante del software industriale SIMATIC.

STEP 7 è utilizzato per applicazioni di dimensioni o complessità superiori per le quali è necessaria ad es. una programmazione con linguaggi evoluti o linguaggi di tipo grafico

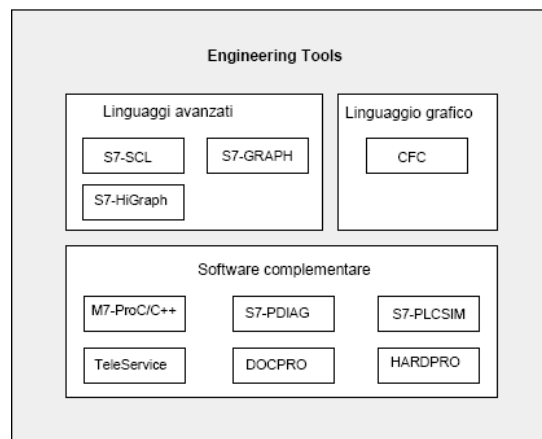
(vedi Engineering Tools) oppure è richiesto l'impiego di unità funzionali o di unità di comunicazione. Sotto STEP 7 possono essere utilizzati pacchetti software SIMATIC addizionali come ad. es. gli Engineering Tools.

Engineering Tools

Gli Engineering Tools aprono nuove possibilità di programmare con facilità soluzioni di automazione mirate al compito specifico.

Essi comprendono:

- linguaggi avanzati per programmatori
- linguaggio grafico per tecnici
- software complementare per diagnostica, simulazione, manutenzione a distanza, documentazione degli impianti, ecc.



Sono disponibili per la programmazione i seguenti strumenti:

[S7-SCL](#)

(Structured Control Language), il linguaggio evoluto basato su PASCAL per la programmazione di controllori SIMATIC S7/C7.

[S7-GRAPH](#)

è un linguaggio di programmazione che descrive i comandi sequenziali (programmazione con catene di passi). Il processo viene suddiviso in passi, che contengono in particolare azioni per comandare le uscite. La transizione da un passo all'altro viene controllata mediante condizioni di avanzamento.

[S7-PLCSIM](#) (solo S7) consente di simulare i controllori programmabili S7 collegati al sistema di origine a scopo di test.

[S7-HiGraph®](#)

per la descrizione grafica di processi sequenziali o asincroni con grafi di stato per SIMATIC S7/C7.

[CFC](#)

(Continuous Function Chart), lo schema tecnologico per la combinazione grafica di funzioni complesse con SIMATIC S7.

[Distributed Safety](#)

Pacchetto software per lo sviluppo di programmi di sicurezza in KOP e FUP inclusa biblioteca F con esempi di programmazione.

L'impiego degli Engineering Tools è vantaggioso soprattutto con applicazioni di dimensioni o complessità superiore e con CPU di potenza più elevata.

Linguaggi di Programmazione

Tutte le CPU possono essere programmate nei linguaggi base AWL, KOP e FUP.

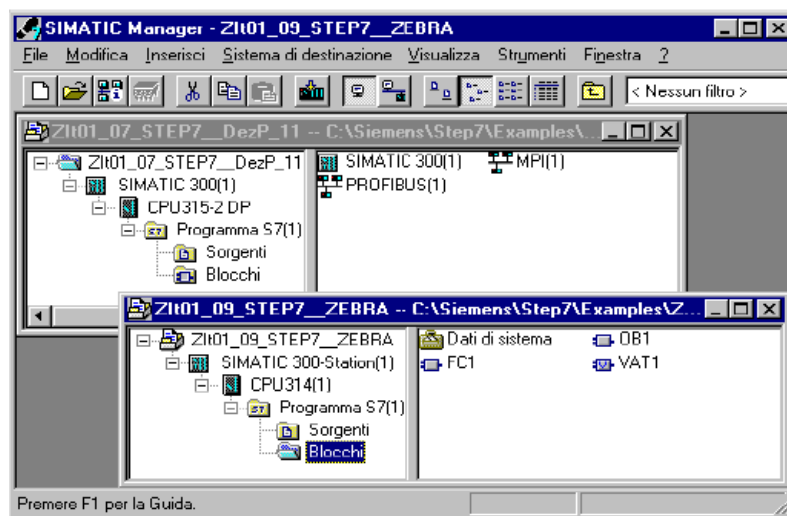
I linguaggi di programmazione KOP, AWL e FUP per S7-300/400 sono parte integrante del software di base.

1. **KOP** (schema a contatti) è un linguaggio di programmazione grafico. La sintassi delle istruzioni assomiglia ad uno schema di circuito. KOP consente all'utente di seguire in modo semplice il flusso dei segnali tra sbarre collettrici, contatti, elementi complessi e bobine.
2. **AWL** (lista istruzioni) è un linguaggio di programmazione testuale vicino al linguaggio macchina. Quando si crea un programma in AWL, le singole istruzioni corrispondono in larga misura alle operazioni con le quali la CPU elabora il programma.
3. **FUP** (schema funzionale) è un linguaggio grafico di programmazione che rappresenta la logica mediante i box dell'algebra booleana. Esso consente inoltre di rappresentare funzioni complesse (ad es. le funzioni matematiche) direttamente in connessione con i box logici.

Altri linguaggi di programmazione sono disponibili come software opzionali.

SIMATIC Manager

Il SIMATIC Manager è il software che gestisce i dati appartenenti a un progetto di automazione, indipendentemente dal sistema di destinazione per cui sono realizzati (S7/M7/C7). Le applicazioni necessarie per la modifica dei dati prescelti vengono avviate automaticamente dal SIMATIC Manager.



Il software di programmazione STEP 7 offre la possibilità di strutturare il programma utente, ovvero di suddividerlo in singole sezioni indipendenti, ottenendo i seguenti vantaggi:

- i programmi di grandi dimensioni possono essere programmati in modo chiaro
- le singole parti del programma possono essere standardizzate
- l'organizzazione del programma viene semplificata
- le modifiche del programma si possono eseguire più facilmente
- il test del programma viene semplificato, poiché può essere eseguito per sezioni
- la messa in servizio viene facilitata.

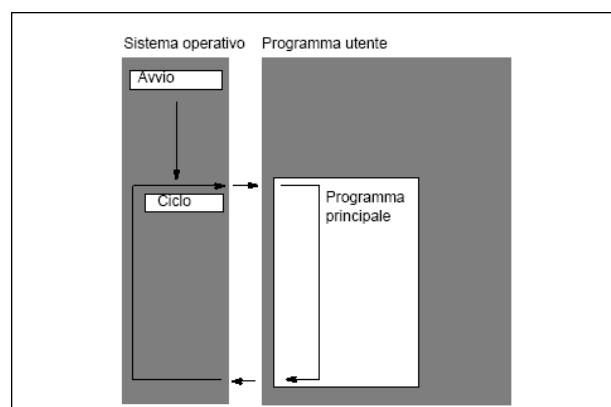
Un processo di automazione può essere suddiviso razionalmente in compiti distinti. Le sezioni di un programma utente strutturato corrispondenti ai singoli compiti vengono definite *blocchi di programma*.

Vi sono diversi tipi di blocchi che possono essere utilizzati all'interno di programmi utenti S7:

Blocco	Descrizione della funzione
Blocchi organizzativi (OB)	Gli OB determinano la struttura del programma utente.
Blocchi funzionali di sistema (SFB) e funzioni di sistema (SFC)	Gli SFB e le SFCs sono integrati nella CPU S7, e rendono accessibili alcune importanti funzioni di sistema.
Blocchi funzionali (FB)	Gli FB sono blocchi con "memoria", programmabili dall'utente.
Funzioni (FC)	Le FC contengono routine di programma per le funzioni più utilizzate.
Blocchi dati di istanza (DB di istanza)	I blocchi dati di istanza vengono assegnati al blocco quando viene richiamato un FB/SFB. Essi vengono generati automaticamente nella compilazione.
Blocchi dati (DB)	I DB sono aree di dati per la memorizzazione dei dati utente. Oltre ai dati rispettivamente assegnati a un blocco funzionale, possono essere definiti dati globali utilizzabili da blocchi qualsiasi.

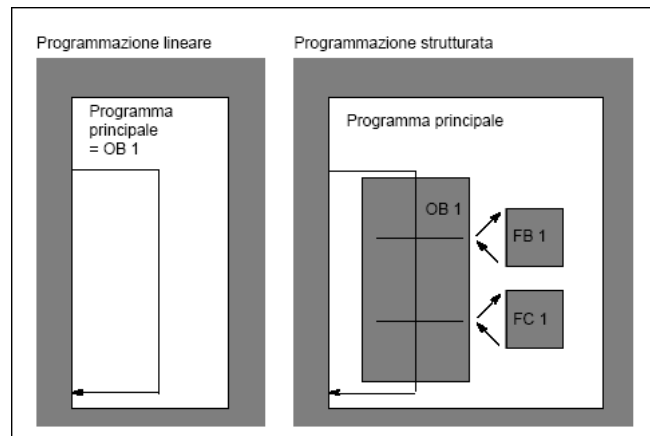
OB, FB, SFB, FC e SFC contengono parti del programma, e vengono pertanto definiti come blocchi di codice.

Nei controllori a logica programmabile l'elaborazione ciclica del programma rappresenta l'elaborazione prevalente: il sistema operativo viene eseguito in un loop di programma (detto ciclo), e richiama quindi in ogni ciclo per una volta il blocco organizzativo OB 1 nel programma principale. Anche il programma utente nell'OB 1 viene elaborato in modo ciclico.



È possibile scrivere l'intero programma utente nell'OB 1 (programmazione lineare). Questa operazione è consigliabile solo per programmi semplici che girano sulle CPU S7-300 occupando poca memoria.

I compiti di automazione complessi possono essere elaborati meglio se vengono suddivisi in compiti parziali più piccoli, che corrispondono alle funzioni tecnologiche del processo di automazione o che devono essere utilizzati più volte. Nel programma utente, i compiti parziali sono rappresentati dai corrispondenti segmenti di programma, ossia dai blocchi (programmazione strutturata).



1. Guida Simatic STEP7

1.1 Avvio di STEP 7

Dopo aver avviato Windows sulla superficie operativa compare l'icona del SIMATIC Manager che serve per entrare nel software STEP 7.

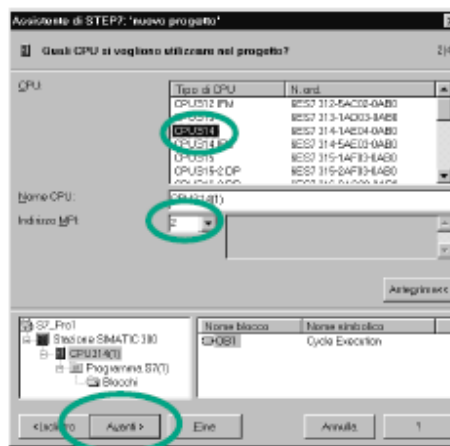


SIMATIC Manager

Fare doppio clic sul simbolo **SIMATIC Manager** nel desktop di Windows. Se l'Assistente non si avvia automaticamente, richiamare il comando di menu **File > Assistente "Nuovo progetto"**.



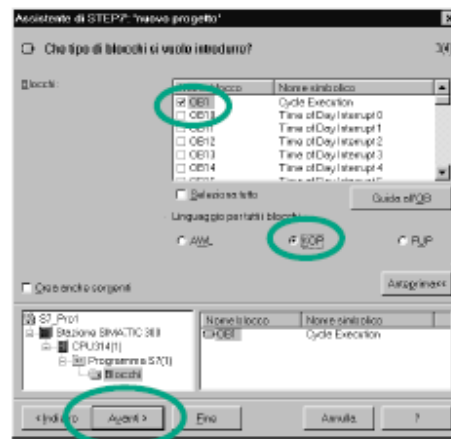
Con **Anteprima** è possibile attivare e disattivare la struttura del progetto che viene creato. Con **Avanti** si giunge alla seconda finestra di dialogo



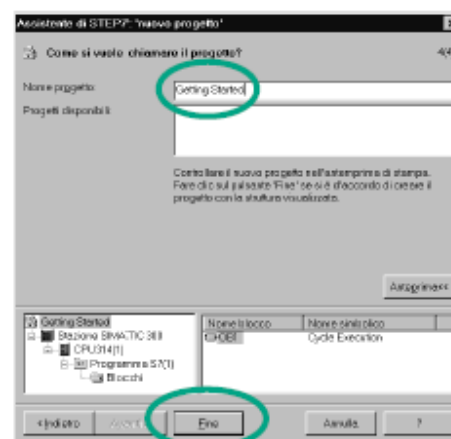
Selezionare la CPU314C-2DP

L'indirizzo MPI (Multi PointInterface), necessario per la comunicazione della CPU con il PG/PC, è preimpostato.

Con **Avanti** si confermano le impostazioni e si giunge alla finestra di dialogo successiva.



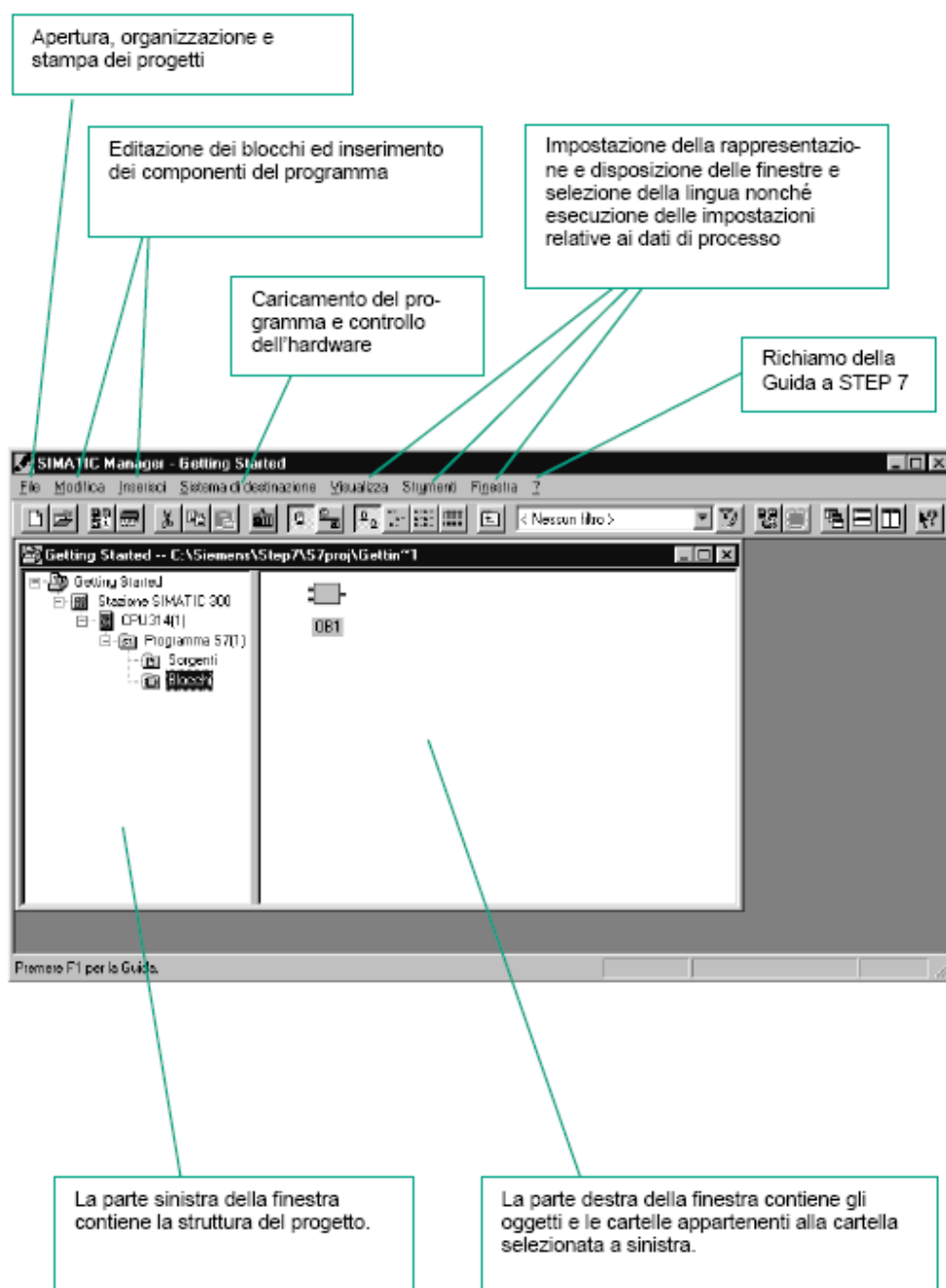
Selezionare il blocco organizzativo **OB1** (a meno che questo non sia già selezionato).
Scegliere il linguaggio di programmazione **KOP**, **FUP** o **AWL**.
Confermare le impostazioni con **Avanti**.



Nella casella "Nome progetto" scegliere con doppio clic il nome proposto, per esempio "Getting Started".

Con **Fine** il nuovo progetto viene generato come nell'anteprima e memorizzato.

Non appena è chiuso l'Assistente di STEP 7, compare il SIMATIC Manager con la finestra di progetto "Getting Started" aperta. Da questa finestra vengono richiamate tutte le funzioni e le finestre di STEP 7.



1.2 Programmazione con nomi simbolici

In un programma STEP 7 si utilizzano operandi quali segnali E/A, merker, contatori, temporizzatori, blocchi dati e blocchi funzionali. Questi operandi possono essere indirizzati in modo assoluto nel programma utente; tuttavia, la leggibilità dei programmi aumenta considerevolmente se si utilizzano simboli.

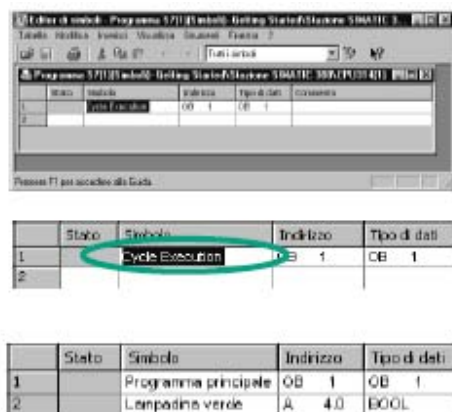
Nel programma utente sarà quindi possibile indirizzare l'operando mediante il simbolo prescelto.

L'indirizzo assoluto può essere sostituito da nomi simbolici a libera scelta.

Nella finestra del progetto "Getting Started" navigare a **Programmi S7 (1)** e con un doppio clic aprire **Simboli**.



La tabella dei simboli è costituita momentaneamente solo dal blocco organizzativo predefinito OB1.



Fare clic su **Cycle Execution** e sovrascriverlo, per il nostro esempio, con "Programma principale". Nella riga 2 introdurre "Lampadina verde" e "A 4.0". Il tipo di dati viene aggiunto automaticamente.

Fare clic nella colonna di commento della riga 1 o 2 per introdurre un commento relativo al simbolo in questione.

Le introduzioni di una riga vengono concluse con **Invio** per inserire una nuova riga. Introdurre nella riga 3 "Lampadina rossa" e "A 4.1" e terminare l'introduzione con **Invio**.

Salvare le introduzioni o le modifiche nella tabella dei simboli e chiudere la finestra.

1.3 Indirizzi e tipi di dati ammessi nella tabella dei simboli

È possibile una sola notazione in tutta la tabella dei simboli. La commutazione tra la notazione tedesca (ex SIMATIC) e internazionale (ex IEC) deve avvenire nel SIMATIC Manager sotto **Strumenti > Impostazioni** nella scheda "Lingua e mnemonico".

Di seguito si riporta: la notazione internazionale e tedesca, il tipo di dati e il campo di valori, ammessi nella creazione della tabella dei simboli.

Internazionale	Tedesca	Spiegazione	Tipo di dati	Campo di valori:
I	E	Bit di ingresso	BOOL	0.0..65535.7
IB	EB	Byte di ingresso	BYTE, CHAR	0..65535
IW	EW	Parola di ingresso	WORD, INT, S5TIME, DATE	0..65534
ID	ED	Doppia parola di ingresso	DWORD, DINT, REAL, TOD, TIME	0..65532
Q	A	Bit di uscita	BOOL	0.0..65535.7
QB	AB	Byte di uscita	BYTE, CHAR	0..65535
QW	AW	Parola di uscita	WORD, INT, S5TIME, DATE	0..65534
QD	AD	Doppia parola di uscita	DWORD, DINT, REAL, TOD, TIME	0..65532
M	M	Bit di merker	BOOL	0.0..65535.7
MB	MB	Byte di merker	BYTE, CHAR	0..65535
MW	MW	Parola di merker	WORD, INT, S5TIME, DATE	0..65534
MD	MD	Doppia parola di merker	DWORD, DINT, REAL, TOD, TIME	0..65532
PIB	PEB	Byte di ingresso periferia	BYTE, CHAR	0..65535
PQB	PAB	Byte di uscita periferia	BYTE, CHAR	0..65535
PIW	PEW	Parola di ingresso periferia	WORD, INT, S5TIME, DATE	0..65534
PQW	PAW	Parola di uscita periferia	WORD, INT, S5TIME, DATE	0..65534
PID	PED	Doppia parola di ingresso periferia	DWORD, DINT, REAL, TOD, TIME	0..65532
PQD	PAD	Doppia parola di uscita periferia	DWORD, DINT, REAL, TOD, TIME	0..65532
T	T	Temporizzatore	TIMER	0..65535
C	Z	Contatore	COUNTER	0..65535
FB	FB	Blocco funzionale	FB	0..65535
OB	OB	Blocco organizzativo	OB	1..65535
DB	DB	Blocco dati	DB, FB, SFB, UDT	1..65535
FC	FC	Funzione	FC	0..65535
SFB	SFB	Blocco funzionale di sistema	SFB	0..65535
SFC	SFC	Funzione di sistema	SFC	0..65535
VAT	VAT	Tabella delle variabili		0..65535
UDT	UDT	Tipo di dati definito dall'utente	UDT	0..65535

1.4 Creazione di un programma nell'OB1

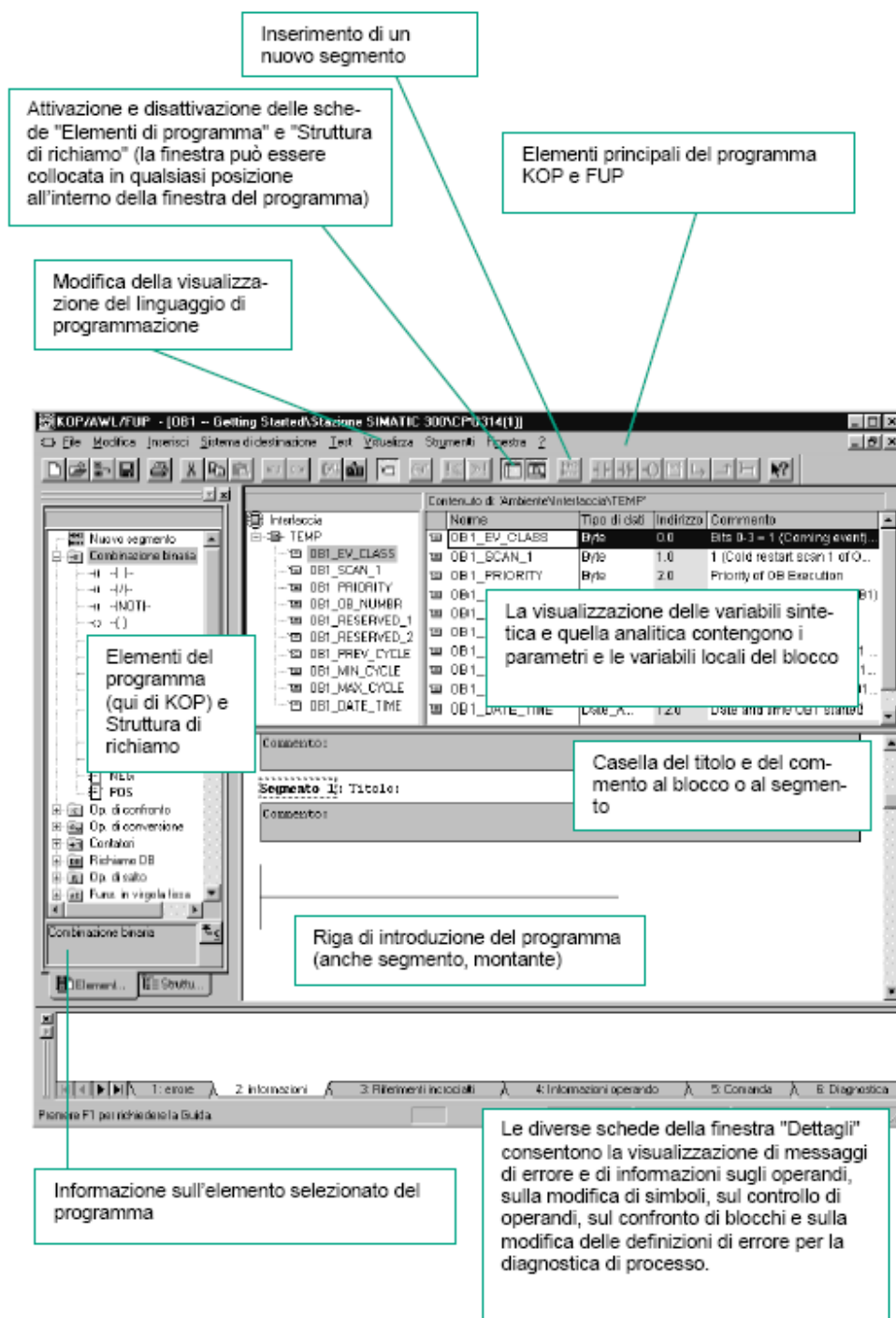
Con STEP 7 i programmi S7 vengono creati nei linguaggi di programmazione standard KOP, AWL oppure FUP. Nell'uso pratico si deve scegliere uno di questi linguaggi.



Fare doppio clic su **OB1** nel progetto "Getting Started". Si apre la finestra del programma "KOP/AWL/FUP".

In STEP 7 l'OB1 viene elaborato ciclicamente dalla CPU. La CPU legge riga per riga i comandi del programma e li esegue. Quando la CPU inizia di nuovo dalla prima riga del programma, significa che ha eseguito esattamente un ciclo. Il tempo impiegato viene definito tempo di ciclo.

Nella finestra del programma "KOP/AWL/FUP" vengono programmati tutti i blocchi.

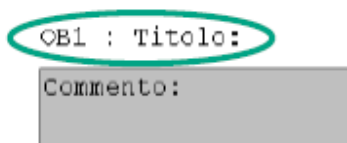


Come programmare un circuito in KOP (Ladder)

Impostare il linguaggio di programmazione **KOP** tramite il menu **Visualizza**.

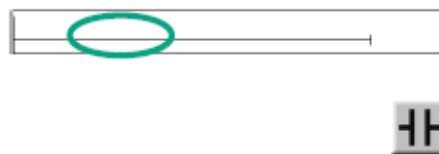


Fare clic nell'area **Titolo** dell'OB1 e introdurre per es. "Programma principale elaborato ciclicamente".

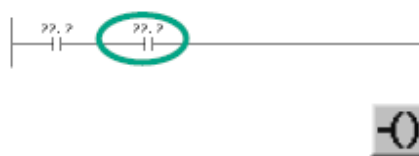


Selezionare il montante per il primo elemento.

Fare clic sul simbolo nella barra degli strumenti ed inserire un contatto normalmente aperto.



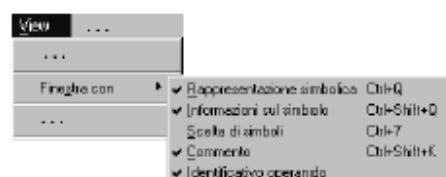
Inserire analogamente un secondo contatto normalmente aperto. All'estremità destra del montante inserire una bobina.



Nel circuito in serie manca ancora l'indirizzamento dei contatti normalmente aperti e della bobina.

Verificare se è attivata la rappresentazione simbolica.

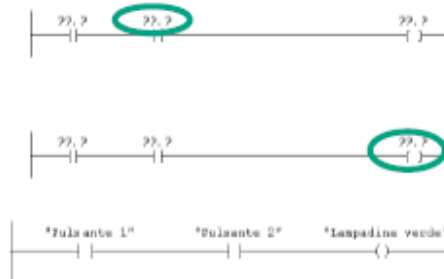
Visualizza > Finestra con > Rappresentazione simbolica



Fare clic su **???** e introdurre il nome simbolico "Pulsante 1" (tra virgolette). In alternativa, scegliere il nome nella casella di riepilogo visualizzata. Confermare premendo **Invio**.



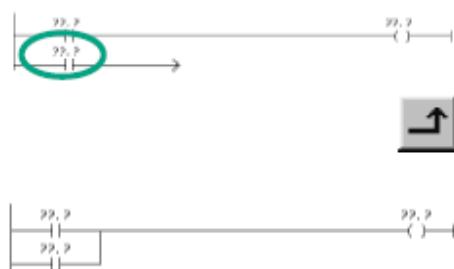
Introdurre il nome simbolico "Pulsante2" per il secondo contatto normalmente aperto. Introdurre il nome "Lampadina verde" per la bobina.



Per Inserire un ramo parallelo. Selezionare il ramo verticale del montante e cliccare sul simbolo inizio diramazione.



Nel ramo parallelo viene inserito per es. un altro contatto normalmente aperto. Chiudere la diramazione cliccando sul simbolo fine diramazione.



Per l'indirizzamento simbolico procedere come sopra.



Salvare il blocco se non vi sono più simboli contrassegnati in rosso. I simboli vengono contrassegnati in rosso se per es. il simbolo non è contenuto nella tabella dei simboli oppure vi è un errore sintattico.

1.5 Come configurare l'hardware

Per "configurazione" si intende la disposizione di telai di montaggio, unità, dispositivi di periferia decentrata e moduli di interfaccia nella finestra della stazione. I telai di montaggio vengono riportati in una tabella di configurazione che contiene sia i telai "reali" che un dato numero di unità innestabili.

Nella tabella di configurazione, STEP 7 assegna automaticamente un indirizzo ad ogni unità. Gli indirizzi delle unità di una stazione possono essere modificati se la CPU è liberamente indirizzabile.

1.5.1 Creazione di stazioni

Dopo aver avviato il SIMATIC Manager:

1. Selezionare il progetto nel lato sinistro della finestra di progetto
2. Selezionare il comando **Inserisci > Stazione > Stazione SIMATIC 300**

La stazione viene creata con un nome predefinito. Si potrà ovviamente sostituirlo con un altro nome più opportuno.

- Selezionare nella finestra di progetto l'oggetto "Stazione", in modo che sia visibile l'oggetto "Hardware" nel lato destro della finestra della stazione.



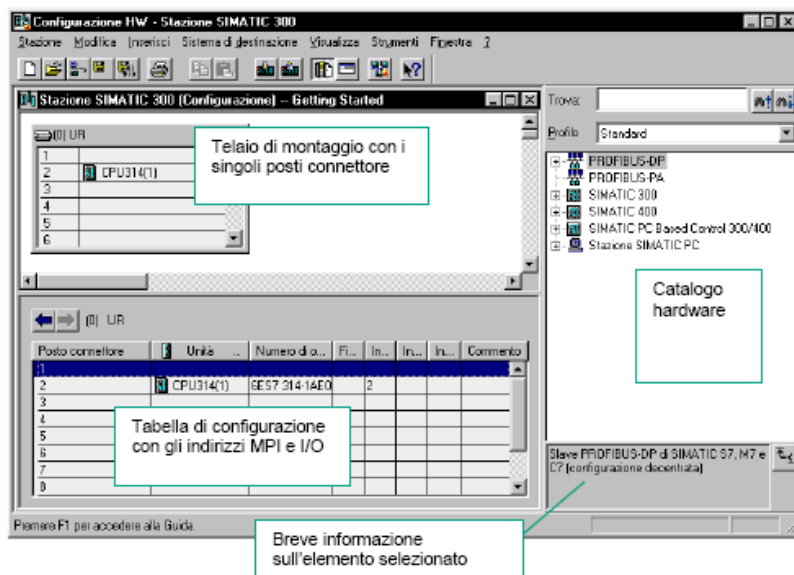
Oggetto "Stazione"

- Fare doppio clic sull'oggetto "Hardware".



Oggetto "Hardware"

E si aprirà la seguente maschera.



Nella configurazione di un sistema di automazione si opera sostanzialmente con due finestre.

- finestra delle stazioni in cui si collocano i telai di montaggio per la configurazione delle stazioni.
- finestra "Catalogo hardware", in cui si selezionano i componenti hardware di cui si ha necessità, p. es. telai di montaggio (rack), unità e moduli di interfaccia

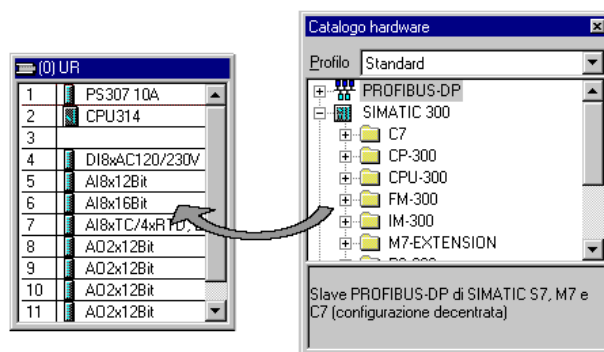
Se la finestra "Catalogo hardware" non è aperta, selezionare il comando **Visualizza > Catalogo** che consente di attivarla e disattivarla.

1.5.2 Operazioni fondamentali nella configurazione di stazioni

Per realizzare la configurazione si deve procedere nel seguente modo.

1. Selezionare i componenti hardware nella finestra "Catalogo hardware".
 2. Trascinare i componenti scelti nella finestra delle stazioni con drag & drop.
- (Si ricordi di selezionare dal catalogo la *CPU314C-2DP* che abbia la versione *firmware 2.0*).

La figura seguente illustra il modo di procedere standard.



Nella parte inferiore della finestra delle stazioni viene visualizzata la vista dettagli del telaio di montaggio inserito o selezionato. Essa contiene una tabella che riporta i numeri di ordinazione e gli indirizzi delle unità.

Posto connettore	Unità	Numero di ordinazione	Firmware	Indirizzo MPI	Indirizzo E	Indirizzo A	Commento
1	PS 307 10A	6ES7 307-1EA00-0AA0					
2	CPU 314	6ES7 314-6CG03-0AB0		2			
3							
4	D18x120/230V	6ES7 321-1FF10-0AA0			0		
5	AI8x12Bit	6ES7 331-7KF02-0AB0			272...287		
6	AI8x16Bit	6ES7 331-7NF10-0AB0			288...303		
7	AI8xTC/4xRTD, Ex	6ES7 331-7SF00-0AB0			304...319		
8	AO2x12Bit	6ES7 332-5HB00-0AB0				320...323	
9	AO2x12Bit	6ES7 332-5HB01-0AB0				324...327	
10							
11							

1.5.3 Definizione delle proprietà dei componenti

Dopo aver disposto i componenti nella finestra delle stazioni, sarà sempre possibile aprire una finestra di dialogo per modificare le proprietà preimpostate (parametri e indirizzi) nel modo qui descritto.

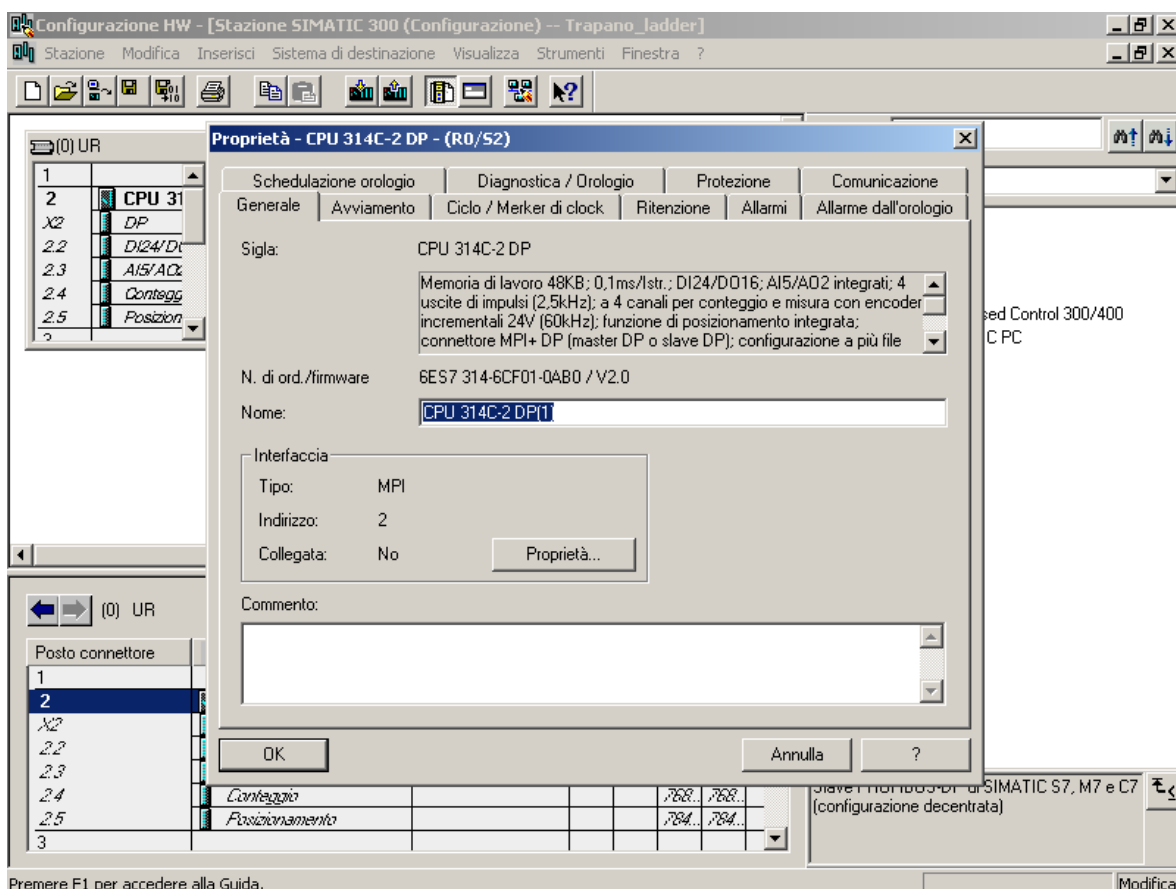
- Fare doppio clic sul componente, o selezionare il comando **Modifica > Proprietà dell'oggetto**

- Con il tasto destro del mouse: spostare il cursore del mouse sul componente, premere il tasto destro e selezionare dal menu contestuale il comando **Proprietà dell'oggetto Proprietà di unità centrali**

1.5.4 Proprietà della CPU.

Cliccando due volte sul componente CPU nella finestra delle stazioni, si apre la seguente maschera in cui è possibile impostare:

- comportamento all'avviamento,
- area dati locali e priorità di allarmi,
- aree di memoria comportamento ritentivo,
- merker di clock, tempo di ciclo.



Di seguito si chiariscono i concetti di Avviamento e Ritenzione:

Avviamento

In STEP7 esistono 3 tipologie di avviamento:

- Riavviamento
- Nuovo avviamento (avviamento a caldo)
- Avviamento a freddo

Per la CPU314C2 DP è previsto il Nuovo Avviamento.

Per definire la modalità di avviamento della CPU in STEP7 si utilizzano i blocchi organizzativi per l'avviamento. Questi vengono elaborati prima dell'elaborazione ciclica del programma (OB1).

Ne esistono tre tipi:

- Riavviamento OB101
- Nuovo avviamento (avviamento a caldo) OB100
- Avviamento a freddo OB102

In particolare il *Nuovo Avviamento* corrisponde all'avviamento a caldo secondo EN61131. Quando in S7 si avvia un'unità centrale, prima dell'elaborazione ciclica del programma (OB1) vengono elaborati il blocco organizzativo OB101 oppure il blocco organizzativo OB100 (nuovo avviamento) oppure il blocco organizzativo OB102 (avviamento a freddo). Per il tipo di avviamento "Nuovo avviamento" vale quanto segue (a condizione che la CPU sia bufferizzata):

- Tutti i blocchi dati e il loro contenuto vengono mantenuti
- Temporizzatori, contatori e merker ritentivi vengono mantenuti mentre quelli non ritentivi vengono resettati.

Nel caso di nuovo avviamento viene letta l'immagine di processo degli ingressi e viene elaborato il programma utente STEP 7 partendo dal primo comando nel blocco OB1.

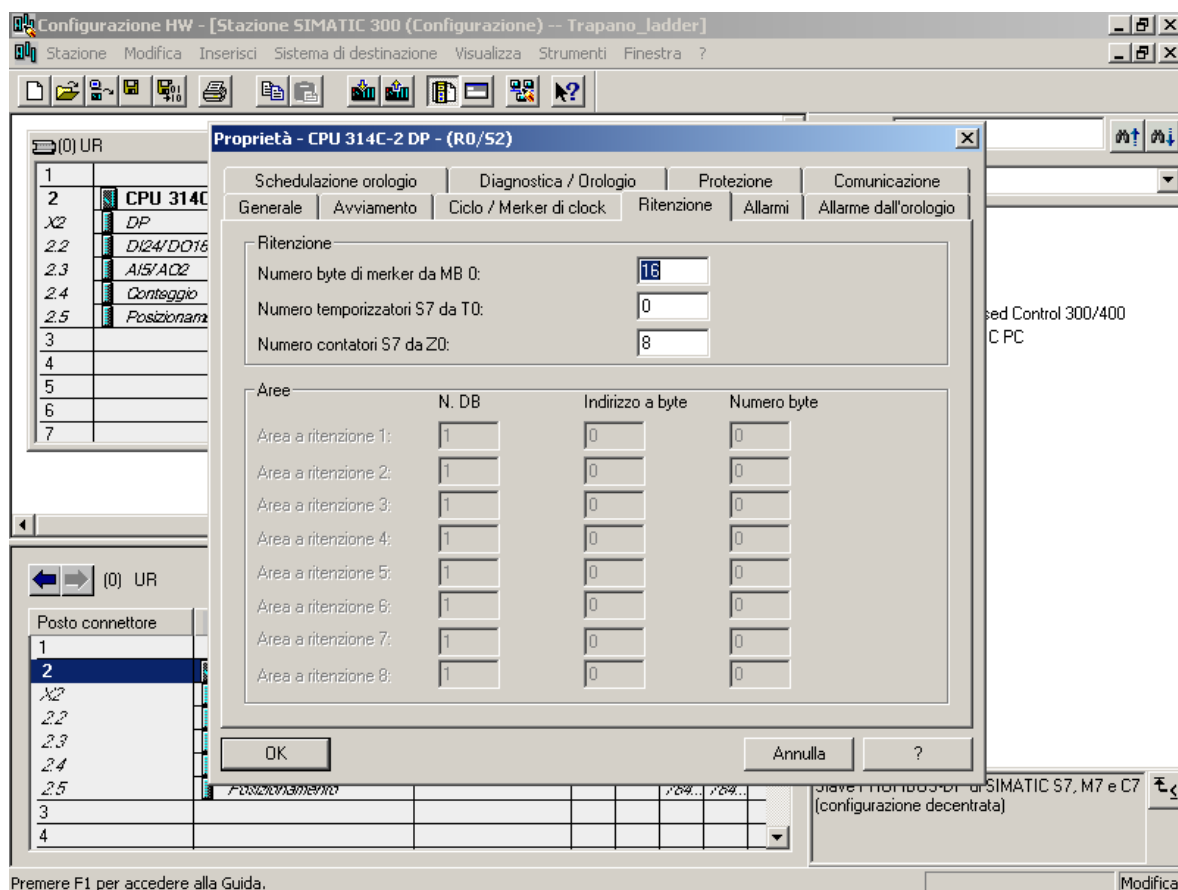
Ritenzione

Durante l'esecuzione del processo industriale possono verificarsi delle anomalie, cadute di tensione, ecc. che comportano l'interruzione del processo. Per far in modo che il processo riparta dall'ultimo stato, e non dall'inizio, è necessario utilizzare delle variabili capaci di memorizzare l'ultimo valore assunto prima del verificarsi dell'anomalia. Tali variabili vengono dette a "ritenzione". Il PLC può offrire la possibilità di salvare queste variabili in batterie tampone o in apposite aree di memoria.

Nel SIMATIC S7 le informazioni che si trovano nelle aree RAM (nella memoria di lavoro) possono essere:

1. salvate con l'ausilio di una batteria tampone; in questo caso il contenuto della memoria di lavoro e dell'area di memoria di scrittura/lettura della memoria di caricamento viene conservato, così come i contatori, temporizzatori e merker (l'area è parametrizzabile).
2. salvate senza batteria tampone (assenza di manutenzione); in questo caso può essere memorizzato in modo permanente un numero massimo (specifico della CPU) di dati provenienti dalla memoria di lavoro, dall'area di memoria di scrittura/lettura della memoria di caricamento e nonché un numero massimo di temporizzatori, contatori e merker nel buffer di backup della CPU.

La *CPU314C-2DP* offre la possibilità di salvare tali variabili senza batteria tampone, come indicato sopra al punto 2, e consente di impostare, attraverso una opportuna maschera, le aree di memoria a ritenzione. Di seguito viene illustrata la scheda "Ritenzione" nella quale è possibile indicare i byte di merker, i temporizzatori e i contatori con ritenzione preimpostata.



1.5.5 Salvataggio della configurazione Hardware

Per salvare la configurazione HW selezionare **Stazione>Salva e compila**

I dati vengono preparati per il trasferimento nella CPU.

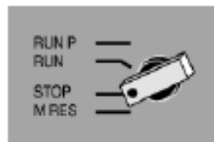
Una volta chiusa la "Configurazione HW", nella cartella Blocchi viene visualizzato il simbolo "Dati di sistema".

1.6 Caricamento e test del programma

Il presupposto per il caricamento del programma è la presenza del collegamento online. Inserire la rete sull'interruttore ON/OFF. Si accende il diodo "DC 5V" sulla CPU.



Ruotare il selettore dei modi operativi su STOP (a meno che non sia già su STOP). Si accende il LED rosso "STOP" .



1.6.1 Come eseguire la cancellazione totale della CPU e commutarla in RUN

Ruotare il selettore dei modi operativi in posizione **MRES** e tenerlo premuto per almeno 3 sec. finché il LED rosso "STOP" non lampeggia lentamente. Rilasciare il selettore dei modi operativi dopo max. 3 sec. ruotarlo di nuovo in posizione **MRES**. Quando il LED "STOP" lampeggia velocemente è stata eseguita la cancellazione totale della CPU. Se "STOP" non lampeggia velocemente, ripetere l'operazione.

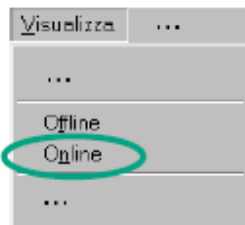


1.6.2 Come caricare il programma nella CPU

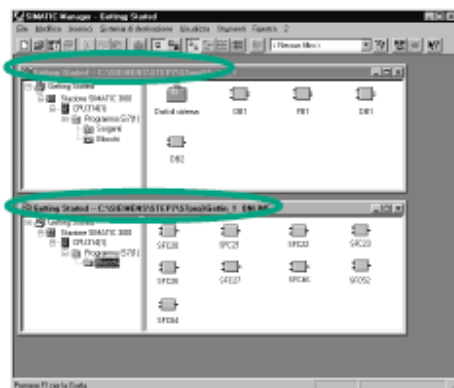
Per il caricamento del programma ruotare ora di nuovo il selettore dei modi operativi su "STOP".



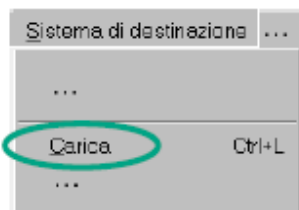
Avviare il SIMATIC Manager e aprire il progetto "Getting Started" (a meno che non sia già aperto) mediante la finestra di dialogo "Apri". Oltre alla finestra "Getting started Offline" richiamare la finestra "Getting Started Online". Lo stato Offline/Online è identificato da un cambiamento di colore nella riga di intestazione.



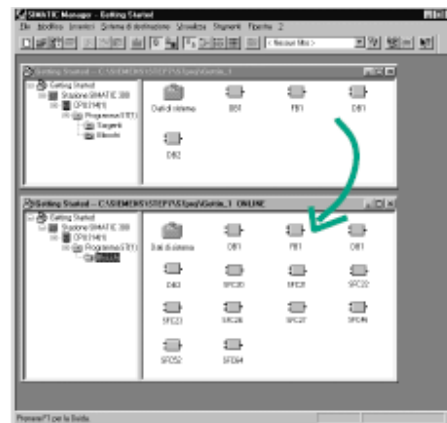
Nelle due finestre navigare alla cartella **Blocchi**. La finestra "Offline" mostra la situazione sul PG, la finestra "Online" mostra la situazione sulla CPU.



Selezionare nella finestra "Offline" la cartella **Blocchi** e caricare il programma nella CPU tramite **Sistema di destinazione > Carica**. Confermare l'interrogazione con **OK**.



Nella finestra "Online" vengono visualizzati, dopo il caricamento, i blocchi del programma.



1.6.3 Come accendere la CPU e controllare lo stato di funzionamento

Girare il selettore di modi operativi su **RUN-P**. Il LED verde "RUN" è acceso e si spegne il LED rosso "STOP". La CPU è pronta al funzionamento. Se è acceso il LED verde, è possibile iniziare con il test del programma. Se il LED rosso continua ad essere acceso, si è verificato un errore. Per una diagnostica dell'errore è necessario analizzare il buffer di diagnostica.



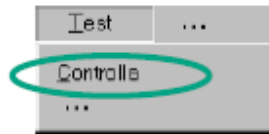
1.6.4 Come testare il programma controllandolo

Mediante la funzione di controllo dello stato del programma viene testato il programma di un blocco. Presupposto è che vi sia un collegamento online con la CPU, la CPU si trovi in RUN o RUN-P ed il programma sia stato caricato nella CPU.

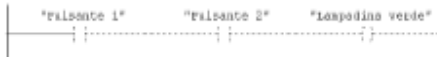
Aprire l'**OB1** nella finestra di progetto "Getting Started Online". Viene aperta la finestra del programma "KOP/AWL/FUP".



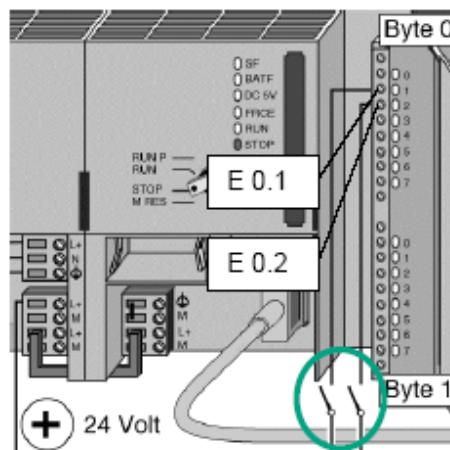
Attivare la funzione **Test > Controlla**.



1.6.5 Come testare con KOP



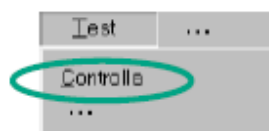
Viene visualizzato il circuito in serie nel segmento 1 in KOP. Il montante viene rappresentato come linea continua fino al pulsante 1 (E 0.1), il che significa che qui è applicata la tensione.



Collegare ora i due pulsanti alla configurazione da testare. Sull'unità di ingresso si accendono i diodi per gli ingressi E 0.1 e E 0.2. Sull'unità di uscita il diodo per l'uscita A 4.0.

Nei linguaggi di programmazione grafici KOP / FUP è possibile seguire l'andamento del risultato del test dal cambiamento di colore nel segmento programmato. Il cambiamento di colore sta ad indicare che il risultato logico combinatorio è fino a questo punto soddisfatto.

Disattivare **Test** > **Controlli** e chiudere la finestra. Chiudere nel SIMATIC Manager la finestra "Online".



1.7 S7_PLCSIM

Il software S7-PLCSIM consente di eseguire e testare i programmi utente su un controllore programmabile (PLC) simulato nel proprio PC o PG.

La simulazione viene effettuata interamente al livello del software STEP 7, per cui non è necessario essere collegati ad alcun componente hardware S7 (CPU od unità I/O).

Utilizzando il PLC S7 simulato l'utente può controllare e testare i programmi per le CPU S7-300 e S7-400 e per WinLC.

S7-PLCSIM mette a disposizione un'interfaccia molto semplice in cui controllare e modificare vari parametri del programma (ad esempio attivare e disattivare gli ingressi). Inoltre consente di utilizzare le diverse applicazioni di STEP 7 mentre si esegue il programma utente sul PLC simulato.

È quindi possibile usare tool quali esempio la tabella delle variabili (VAT) per controllare e modificare le variabili.

S7-PLCSIM presenta inoltre le seguenti caratteristiche.

- La simulazione viene attivata e disattivata da un pulsante collocato sulla barra degli strumenti del SIMATIC Manager. Attivando il pulsante di simulazione si apre il software S7-PLCSIM con la CPU simulata. Quando S7-PLCSIM è in esecuzione la comunicazione passa automaticamente da STEP 7 alla CPU simulata.
- Il PLC simulato esegue i programmi per la CPU S7-300 o S7-400.
- Possono essere create "sottofinestre" che permettono di accedere alle aree di memoria di ingressi e uscite, agli accumulatori e ai registri del PLC simulato. Si può accedere alla memoria anche mediante indirizzamento simbolico
- Si può decidere che i temporizzatori vengano eseguiti automaticamente oppure li si può impostare e resettare manualmente. Si possono inoltre resettare i temporizzatori individualmente o globalmente.
- Come per le CPU reali, il modo di funzionamento CPU (RUN, RUN-P, STOP) può essere commutato. Inoltre S7-PLCSIM prevede il modo Interrompi che permette di arrestare temporaneamente la CPU senza modificare lo stato del programma.
- Si possono utilizzare OB di allarme nel PLC simulato per verificare le reazioni del programma
- È possibile registrare una serie di eventi (manipolare le aree di memoria degli ingressi e delle uscite, merker, temporizzatori e contatori) e riprodurre la registrazione effettuata per automatizzare i test di programma.

S7-PLCSIM consente di utilizzare tutte le applicazioni di STEP 7 per controllare e modificare le attività del PLC simulato e per testare il programma utente. Malgrado il PLC simulato esista soltanto nel software (senza richiedere uno speciale hardware), STEP 7 opera come se il PLC simulato fosse un vero componente hardware.

1.8 Simulazione con S7-PLCSIM

Per poter accedere alla modalità di simulazione dal SIMATIC Manager non devono sussistere collegamenti con PLC reali. Per iniziare ad utilizzare S7-PLCSIM, effettuare le seguenti operazioni:

1. Aprire il SIMATIC Manager.

2. Fare clic su  oppure selezionare il comando di menu **Opzioni > Simula unità**.

Questa operazione avvia l'applicazione S7-PLCSIM ed apre una sottofinestra CPU (indirizzo MPI di default: 2).




3. Nel SIMATIC Manager, aprire il progetto.

4. All'interno del progetto, portarsi sull'oggetto "Blocchi" .

5. Nel SIMATIC Manager, fare clic su  oppure selezionare il comando di menu **Sistema di destinazione > Carica** per caricare l'oggetto "Blocchi" nel PLC simulato.

6. A questo punto, il sistema chiede se si vogliono caricare i dati di sistema; rispondere "NO" se non si intende caricare la configurazione hardware nel PLC simulato, SÌ per caricare la configurazione hardware.

7. Nell'applicazione S7-PLCSIM, creare ulteriori "sottofinestre" per controllare le informazioni dal PLC simulato:


- Fare clic su  oppure selezionare il comando di menu **Inserisci > Ingresso**. La sottofinestra visualizza **IB0** (byte di ingresso 0).
- Fare clic su  oppure selezionare il comando di menu **Inserisci > Uscita** per inserire un'ulteriore sottofinestra **QB0** (Output Byte 0).
- Fare clic su  oppure selezionare il comando di menu **Inserisci > Temporizzatore** per inserire l'oggetto "Temporizzatore".

8. Selezionare il menu **Sistema di destinazione** in S7-PLCSIM e verificare che accanto ad **Alimentazione** compaia un pallino (•).

9. Selezionare **Esegui > modo di scansione** e verificare che compaia un pallino (•) accanto a **Ciclo continuo**.

10. Commutare la CPU simulata in RUN facendo clic sulla casella RUN oppure RUN-P check.


11. Fare clic sul bit 0 di IB0 per simulare l'attivazione dell'ingresso 0.0 ed osservare gli effetti di questa operazione sul temporizzatore e su QB0.

12. Fare clic su  oppure selezionare il comando di menu **File > Salva PLC con nome...** per salvare come nuovo file lo stato corrente del PLC simulato.

1.9 Come utilizzare STEP 7 per controllare la simulazione del programma

È possibile utilizzare i tool di STEP 7 per controllare la simulazione del programma:

1. Dopo aver creato le sottofinestre (vedere sopra al punto 7), entrare nel SIMATIC Manager.

2. Fare clic su  oppure selezionare il comando **Visualizza > Online** per passare al modo online.

3. Spostarsi all'oggetto "Blocchi" nel progetto ed aprire OB1. Questa operazione richiama l'applicazione "KOP/AWL/FUP".

4. Quando la CPU simulata si trova in modo RUN ed è stato attivato il bit 0 di IB0 (vedere sopra al punto 11), passare alla finestra dell'applicazione "KOP/AWL/FUP" e selezionare **Test > Controlla** per verificare l'effetto di tali modifiche sul programma.

1.10 S7_GRAPH

S7-GRAPH permette di programmare con rapidità e chiarezza le operazioni sequenziali che si desiderano comandare con un sistema di automazione SIMATIC. Il processo viene suddiviso in singoli passi con un volume di funzioni tra loro ben definite.

Nei singoli passi vengono stabilite le funzioni da eseguire e i passaggi sono regolati dalle transizioni (condizioni di avanzamento). Queste condizioni vengono definite con i linguaggi di programmazione KOP (schema a contatti) o FUP (schema logico).

I comandi sequenziali comandano il processo nell'ordine prestabilito, e in dipendenza di determinate condizioni.

Perché un programma sia eseguibile sono necessari almeno tre blocchi per un comando sequenziale:

- un blocco STEP 7 (1), in cui viene richiamato l'FB di S7-GRAPH. Questo blocco può essere un OB, una FC o un altro FB.
- un FB di S7-GRAPH (2), che descrive singoli compiti e condizioni del comando sequenziale. Tali compiti (passi) e condizioni (transizioni) possono essere organizzati in una o più catene sequenziali;
- un DB di istanza (3), contenente i dati e parametri del comando sequenziale. Il DB di istanza è assegnato all'FB di S7-GRAPH, e può venire generato automaticamente dal sistema.

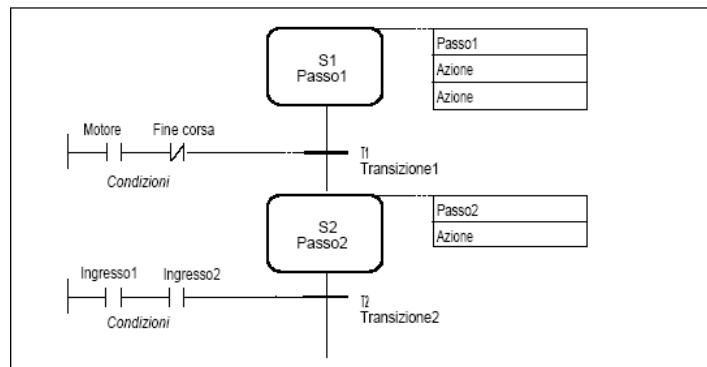
Le *catene sequenziali* vengono programmate nell'FB di S7-GRAPH. Ogni catena sequenziale consiste di una serie di passi attivati in una determinata sequenza, a seconda delle transizioni (condizioni di avanzamento). La struttura più semplice della catena sequenziale è la successione lineare di passi e transizioni senza diramazione.

Una catena lineare comincia con un passo e finisce con una transizione a cui può seguire un salto a un passo qualsiasi o un fine catena.

La sequenza lineare può essere ampliata tramite:

- diramazioni (alternative, simultanee)
- salti a passi qualsiasi
- ulteriori catene sequenziali che vengono eseguite in dipendenza dalla prima catena o in modo completamente indipendente

- operazioni cicliche, disposte prima o dopo la catena sequenziale.



1.10.1 Creazione di un progetto in S7-GRAPH

Dopo aver creato un nuovo progetto nel Simatic Manager attraverso la procedura **File > Assistente "Nuovo progetto"** e aver redatto la tabella dei simboli:

1. Aprire nel SIMATIC Manager la cartella "Blocchi".
2. Selezionare il comando di menu **Inserisci > Blocco S7 > Blocco funzionale**.
3. Indicare nella finestra di dialogo "Proprietà", "GRAPH" come linguaggio di programmazione.

Nella cartella "Blocchi" viene creato un FB vuoto a cui viene assegnato il numero 1.

Dopo l'avvio dell'editor S7-GRAPH mediante doppio clic su FB 1, il sistema inserisce il primo passo (passo iniziale) e la prima transizione. Il miglior modo per creare la struttura del comando sequenziale è nella rappresentazione "Catena". Per visualizzare condizione e azioni, attivare il comando di menu **Visualizza > Finestra con > condizioni e azioni**.

Posizionare la barra degli strumenti "Catena" al margine sinistro dello schermo e collocare con il mouse tutti gli ulteriori **passi**, le **transizioni**, la **diramazione alternativa** e il **salto** dalla fine all'inizio della catena. Di seguito vengono riportati alcuni simboli che si trovano alla sinistra dell'editor S7_GRAPH:



1.10.2 Programmazione di azioni

Nei passi vengono programmate le azioni, che tra l'altro comandano ingressi, uscite e merker e attivano e disattivano i passi della catena sequenziale, e richiamano i blocchi. Le azioni contengono quindi istruzioni per il comando del processo. Se il passo è attivo, le azioni vengono eseguite in successione "dall'alto al basso".

Per effettuare la programmazione delle azioni da associare agli step:

1. Selezionare il comando **Inserisci > Azione**.

il cursore assume la forma seguente



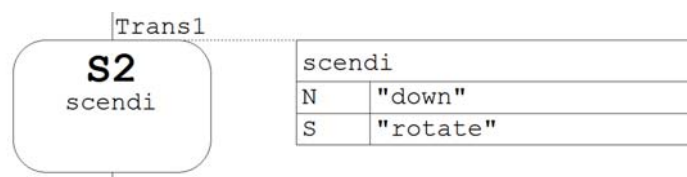
2. Inserire una riga vuota facendo clic sulle caselle apposite.

3. Indicare le azioni.

Un'azione consiste di un' operazione e un operando. Di seguito si riportano solo alcuni tipi di operazioni:

- S Imposta (**Set**): appena il passo è attivo, l'operando viene impostato a 1 e rimane così impostato
- R Resetta (**Reset**): Appena il passo è attivo , l'operando viene impostato a 0 e rimane così impostato
- N finché il passo è attivo, il segnale dell'operando è = 1.
- D Ritardo all'inserzione (**Delay**): n sec. dopo l'attivazione del passo, il segnale dell'operando è 1 per la durata di attivazione del passo. Ciò non vale se la durata di attivazione del passo è minore di n sec.
- CALL Richiamo del blocco: finché il passo è attivo, viene richiamato il blocco indicato(FB, FC, SFB,SFC). SFB/FB = blocco funzionale di sistema/blocco funzionale; SFC/FC = funzione di sistema/funzione. Per FB/SFB è necessario un DB di istanza poiché sono blocchi con memoria.

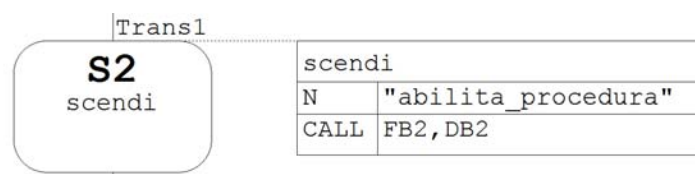
Azioni con funzioni standard



In questa figura si osserva che al passo "scendi" sono state associate due azioni :

- (1) finché il passo è attivo, il segnale di uscita down è =1
- (2) finché il passo è attivo l'uscita rotate viene impostato a 1 e rimane così impostato

Azioni che richiamano blocchi



In questa figura viene riportato l'esempio di richiamo del blocco funzionale. (Si ricorda che il blocco funzionale può essere realizzato in qualunque linguaggio di programmazione. In questo caso è stato usato il linguaggio KOP).

La prima azione indica che:

- (1) finché il passo è attivo, la variabile "abilita_procedura" è =1. Questa variabile è utilizzata all'interno del blocco FB2.

La seconda azione indica che:

- (2) finché il passo è attivo, viene richiamato il blocco FB2 con il relativo blocco dati di istanza DB2.

Alla variabile "abilita_procedura", presente all'interno dell'FB2 come contatto aperto, viene attribuito il significato di passo attivo nel momento in cui essa assume il valore uno. Infatti è fondamentale che tale variabile assuma il valore uno, in corrispondenza dell'attivazione del passo, per consentire al blocco funzionale di eseguire il suo codice.

Nel caso in cui si deve richiamare un blocco FC o SFC non è necessario inserire un DB di istanza poiché sono blocchi privi di memoria (Es. CALL FC10)

1.10.3 Programmazione di condizioni

Le condizioni sono stati binari del processo che possono essere combinati tra loro secondo la logica booleana con elementi KOP o FUP (contatto normalmente aperto, contatto normalmente chiuso, box And, box Or, box di confronto). Il risultato logico combinatorio (RLC) può influenzare le singole azioni di un passo, il passo intero, l'avanzamento al passo successivo o l'intera catena sequenziale. Le combinazioni logiche vengono programmate in KOP o FUP.

Le condizioni sono

- eventi (p. es., si esce dal passo attivo) e
- stati (p. es., quando viene impostato l'ingresso E2.1)

Le condizioni compaiono nelle aree seguenti:

- Transizione (avanzamento al passo successivo)
- Interlock (interblocco del passo)
- Supervision (controllo del passo)
- Operazioni cicliche (condizioni e richiami del blocco all'inizio e/o alla fine della catena sequenziale)

Transizione

In una transizione vengono programmate le condizioni che comandano l'avanzamento da un passo a quello successivo.

Transizione vuota

Le transizioni senza condizioni rappresentano transizioni vuote. Esse non interferiscono nello svolgimento della catena. Il passo attivo viene eseguito solo una volta. Il passo successivo è quello valido.

Interlock

L'interlock è una condizione programmabile per l'interblocco del passo che influenza l'esecuzione delle singole azioni.

Se è soddisfatta la combinazione delle condizioni, vengono eseguite le azioni combinate con l'interlock.

Supervision

La supervision è una condizione programmabile per il controllo del passo, che influenza l'avanzamento da un passo a quello successivo.

Operazioni cicliche

Le operazioni cicliche sono condizioni e richiami di blocchi che sono posti a monte o a valle della catena sequenziale. Esse vengono elaborate, indipendentemente dallo stato della catena sequenziale, una volta per ciclo.

Per le condizioni di avanzamento alle transizioni vengono utilizzate le operazioni combinatorie di bit "Contatto normalmente aperto", "Contatto normalmente chiuso" e "Box di confronto".

Come realizzare le transizioni

1. Scegliere il set mnemonico "KOP" e selezionare nella barra degli strumenti "KOP/FUP" il simbolo corrispondente

- Inserisci contatto normalmente aperto
- Inserisci contatto normalmente chiuso
- Inserisci box di confronto

2. Posizionare i simboli nel punto corrispondente facendo clic sulle linee di transizione.

Si può uscire dal modo di editazione in qualsiasi momento con il tasto ESC.

3. Indicare gli operandi. Selezionare a tal fine la casella di testo corrispondente con un clic sul segnaposto "???". Introdurre quindi un operando assoluto o simbolico (p. es. E0.7, "Pulsante di avvio").

4. Se si desidera, può essere indicato un commento sulla catena sequenziale. Il campo di commento si trova nel tipo di rappresentazione "Catena" in alto a sinistra, e può essere aperto con un semplice clic.

Per la programmazione di box di confronto è possibile utilizzare come operandi le informazioni di sistema sui passi. In questo caso gli operandi hanno il seguente significato:

1. *Nome_passo.T*: momento attuale o ultimo momento di attivazione del passo
2. *Nome_passo.U*: momento attuale o ultimo momento di attivazione del passo senza tempo di avaria

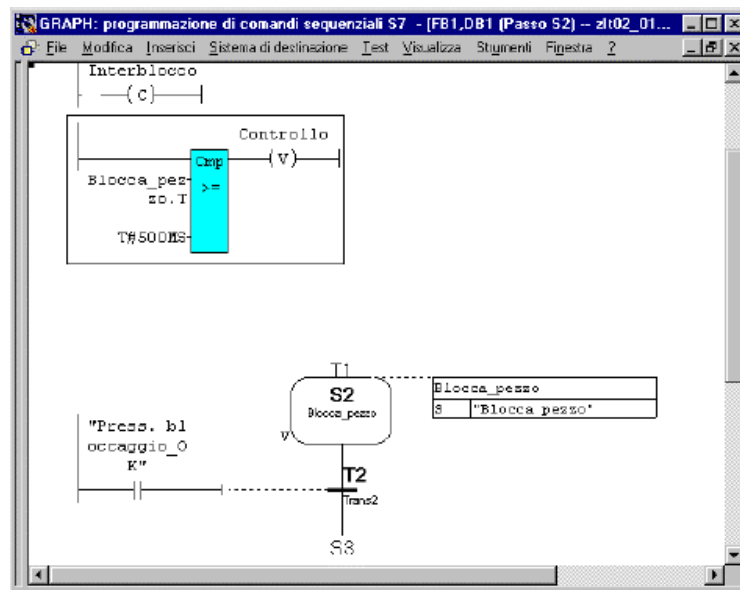
Programmazione dei controlli

Procedere come segue per programmare i controlli.

1. Con un doppio clic sul passo passare dal tipo di rappresentazione "Catena" al tipo "Passo singolo".

2. Selezionare nella barra degli strumenti "KOP/FUP" il simbolo Inserisci box di confronto

3. Posizionare il box di confronto nel punto corrispondente sulla linea di controllo e immettere il tempo di controllo desiderato.



Operandi specifici di S7-GRAPH per condizioni

Vi è la possibilità di utilizzare le informazioni di sistema sui passi come operandi in transizioni, supervision, interlock e condizioni cicliche.

Operando	Significato	Utilizzato in
Si.T	Tempo di attivazione corrente o più recente del passo	Comparatore, assegnazione
Si.U	Tempo di attivazione totale del passo i senza tempo di avaria	Comparatore, assegnazione
Si.X	Indica se il passo i è attivo	Contatto normalmente aperto/Contatto normalmente chiuso
Transi.TT	La transizione i è soddisfatta Verifica che tutte le condizioni della transizione i siano soddisfatte	Contatto normalmente aperto/Contatto normalmente chiuso

1.10.4 Salvataggio e chiusura della catena sequenziale

S7-GRAPH salva e compila automaticamente gli **FB**. Compilare significa verificare che la catena sequenziale sia priva di errori di sintassi. Soltanto gli FB di S7-GRAPH esenti da errori possono essere salvati e trasmessi alla CPU. Non è possibile chiudere gli FB se contengono errori.

Se gli errori vengono corretti solo in un secondo momento, si potrà invece salvare l'FB di S7-GRAPH come **sorgente**, oppure creare una sorgente al posto di un FB di S7-GRAPH. Al contrario che con l'FB di S7-GRAPH, è possibile salvare una sorgente di S7-GRAPH anche se la catena sequenziale contiene degli errori.

Per effettuare il salvataggio

1. Selezionare il comando di menu **File > Salva**. Viene aperta la finestra di dialogo "Seleziona DB di istanza" con il DB di istanza (DB 1) predefinito.
2. Immettere le impostazioni facendo clic su "OK". Il DB di istanza viene creato automaticamente e copiato nella cartella "Blocchi".
3. Per chiudere la catena sequenziale selezionare il comando di menu **File > Chiudi**.

1.10.5 Inizializzazione della catena sequenziale e richiamo nel programma OB1

Inizializzazione della catena sequenziale

Il blocco funzionale FB realizzato in S7-GRAPH contiene la catena sequenziale che descrive il processo. Per avviare tale catena è necessario creare all'interno dell'OB1 un richiamo dell'FB, e si deve specificare il parametro di inizializzazione della catena sequenziale: INIT_SQ (init-sequence)

Nel momento in cui il parametro booleano INIT_SQ (init_sequence) passa a "1", vengono attivati tutti i passi della catena contrassegnati come passi iniziali e tutti gli altri passi vengono disattivati.

Il passo iniziale è il passo di una catena sequenziale che diventa attivo al primo richiamo di un FB di S7-GRAPH senza una precedente interrogazione di condizioni. Il passo iniziale non deve essere necessariamente il primo passo di una catena sequenziale e il suo simbolo è diverso dagli altri passi.

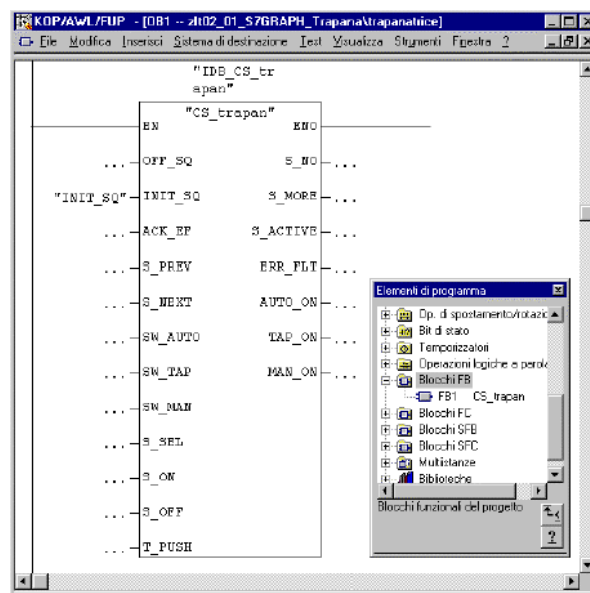
Nel funzionamento ciclico della catena sequenziale questo passo, come ogni altro passo, diventa attivo solo se sono soddisfatte le condizioni della transizione precedente.

Programmazione dell'OB1

Il programma della catena sequenziale viene richiamato ed avviato dal blocco organizzativo OB 1. È possibile creare l'OB 1 in KOP, FUP, AWL o SCL. In questo caso viene creato in KOP.

Procedere nel modo seguente.

1. Aprire in SIMATIC Manager nel programma S7 la cartella "Blocchi".
2. Avviare l'editor "KOP/AWL/FUP" facendo doppio clic sull'OB1.
3. Selezionare mediante il comando di menu **Visualizza** il linguaggio di programmazione KOP.
4. Selezionare il segmento 1, fare doppio clic sul blocco FB 1 nel catalogo degli elementi del programma per inserire il richiamo della catena sequenziale.
5. Indicare il nome del rispettivo blocco dati di istanza per mezzo del box KOP.
6. Selezionare il parametro di ingresso INIT_SQ, inserire un elemento di contatto normalmente aperto per mezzo della barra degli strumenti "KOP", e definirlo come "INIT_SQ". Con questo parametro si ha la possibilità di avviare la catena sequenziale, definita nel modo online, dal passo iniziale.
7. Selezionare il comando di menu **File > Salva** e chiudere il blocco organizzativo con il comando di menu **File > Chiudi**.



1.10.6 Caricamento del programma nella CPU e test della catena sequenziale

Caricamento del programma utente

Per poter caricare il programma nella CPU del sistema di automazione, è necessario caricare dal SIMATIC Manager tutti i blocchi (DB1, FB1, OB1, FC70/71, FC72 o FC73). Procedere quindi nel modo seguente.

1. Aprire nel SIMATIC Manager il programma S7 e selezionare la cartella "Blocchi".
2. Selezionare il comando di menu **Sistema di destinazione > Carica**.

In maniera alternativa è possibile avviare S7-PLCSIM per simulare il progetto.

Test del programma utente

Per testare il programma utente è necessario essere collegati in modo online con una CPU o aver avviato S7-PLCSIM.

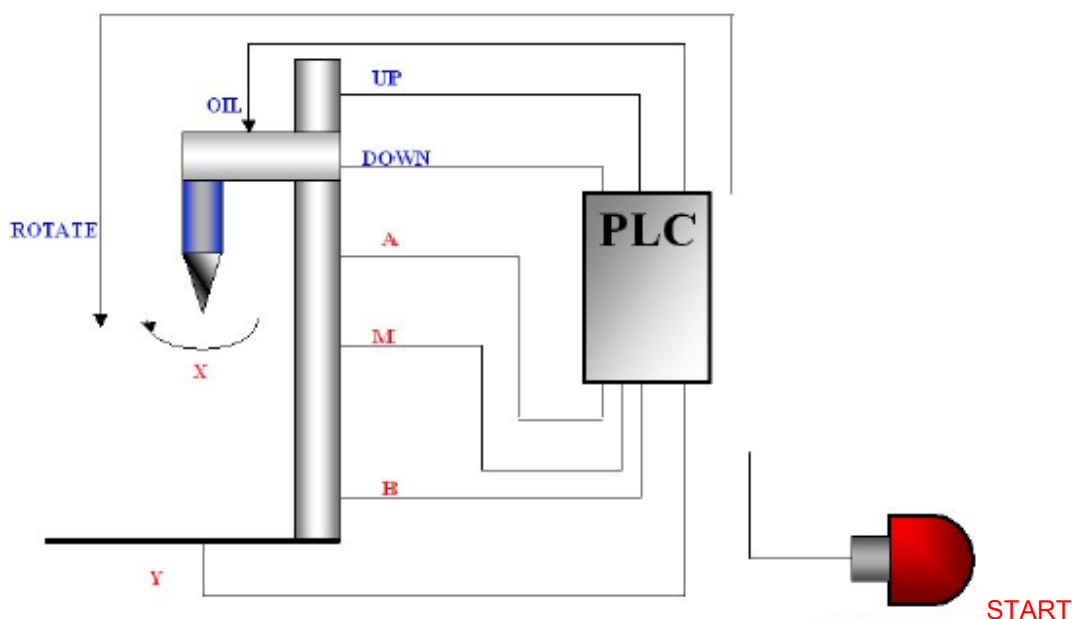
1. Aprire nel SIMATIC Manager la finestra di progetto.
2. Aprire la catena sequenziale tramite doppio clic sull'FB 1.
3. Selezionare il comando di menu **Test > Controlla**.

Viene visualizzato lo stato di programma (il passo iniziale è attivo). I passi attivi vengono contrassegnati a colori.

2 Applicazioni

2.1 Svolgimento del progetto - Trapano

Il sistema da analizzare è costituito da un trapano che deve effettuare un foro su un pezzo posizionato alla base del sistema. Il trapano è montato su un supporto che gli permette di muoversi dall'alto verso il basso e viceversa. Il dispositivo è azionato da un pulsante che viene attivato dall'utente, quando il pulsante viene premuto la punta del trapano inizia a roteare e il trapano scende verso il basso. Per consentire che anche pezzi di grandi dimensioni siano forati correttamente, a metà corsa si controlla se la punta del trapano ha già toccato il pezzo, in tal caso il trapano viene fatto risalire alla posizione iniziale per oleare la punta, nell'altro caso si procede nella discesa verso il basso. Quando la punta del trapano giunge nella posizione più bassa indicata dal sensore B il pezzo è stato completamente forato e il trapano può risalire. Gli ingressi della parte di controllo sono il segnale START del pulsante utente, il segnale A che indica la posizione alta del trapano, il segnale M che indica la posizione media del trapano, il segnale B che indica la posizione bassa del trapano e il segnale Y che indica la presenza del pezzo da forare. Le uscite sono i segnali UP e DOWN che agiscono sulle azioni di discesa e risalita del trapano, il segnale ROTATE per roteare la punta e il segnale OIL per oleare la punta.



Analisi del problema

Individuiamo le variabili del sistema:

Uscite del PLC

UP:variabile digitale utilizzata per far muovere verso l'alto il trapano

DOWN:variabile digitale utilizzata per far muovere verso il basso il trapano.

ROTATE:variabile digitale utilizzata per far ruotare la punta del trapano

OIL:variabile digitale utilizzata per oleare la punta del trapano.

Ingressi del PLC

Start: variabile booleana di ingresso che aziona il sistema

A: Fine corsa A. Se il valore è 1 indica che il trapano si trova nella posizione più alta del supporto.

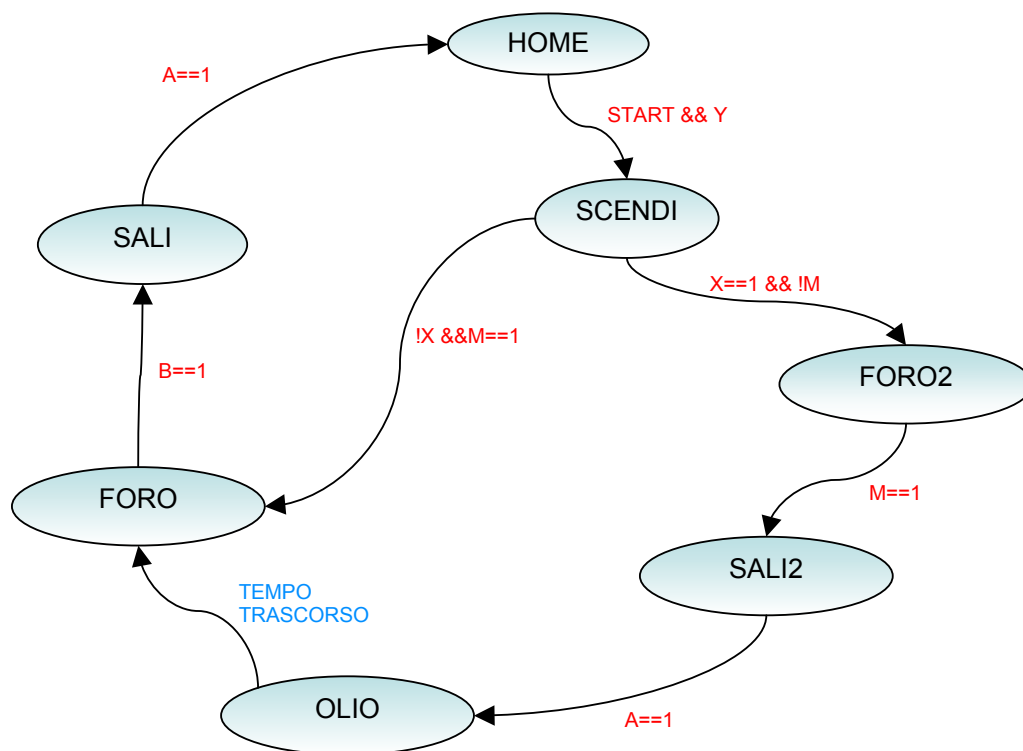
M: Fine corsa M. Se il valore è 1 indica che il trapano si trova nella posizione centrale del supporto.

B: Fine corsa B. Se il valore è 1 indica che il trapano si trova nella posizione più bassa del supporto.

Y: Sensore di pressione Y. Se il valore è 1 indica che il pezzo è presente nella base

X: Sensore di pressione X. Se il valore è 1 indica che il trapano è vicino al pezzo da forare.

La macchina di stato è la seguente:



Lo stato iniziale è denominato Home, variabile locale sempre inizializzata ad 1, cioè sempre true. In tale stato non viene effettuata alcuna operazione.

Il ciclo può avere inizio se il pulsante di START è stato attivato ed il pezzo si trova sulla base del sistema (Y). Il passaggio allo stato "scendi" quindi avviene se la variabile d'ingresso (START and Y) = 1

2) **Scendi** :

ROTATE = 1

DOWN = 1

Il trapano in questo stato entra in funzione, in particolare inizia a scendere e la punta inizia a ruotare.

A questo punto bisogna effettuare un controllo sulla variabile X, in particolare si ha che:

Se X = 1 gli stati da seguire sono:

3) **FORO2** :

ROTATE=1

DOWN = 1

In questa fase il trapano continua a scendere sino a quando incontra il pezzo ed inizia a forarlo. Il passaggio allo stato successivo avviene se la variabile d'ingresso M è posta pari ad 1, cioè se il trapano si trova in una posizione centrale.

4) **SALI2** :

ROTATE = 1

UP = 1

In questo stato il trapano viene riportato nella parte alta del supporto per oleare la punta. Il passaggio allo stato successivo avviene se la variabile d'ingresso A è posta pari ad 1.

5) **OLIO** :

ROTATE = 1

UP = 0

OIL = 1

Il passaggio allo stato "foro" si ha mediante il segnale TEMPO TRASCORSO fornito da un timer. Questo timer viene introdotto per fare in modo che la punta venga raffreddata.

Se X non è uguale a 1 gli stati da seguire sono:

6) **FORO** :

ROTATE = 1

DOWN = 1

OIL = 0

In questo stato il trapano si muove verso il basso mentre la punta continua a ruotare. Il passaggio allo stato successivo avviene se la variabile d'ingresso B è posta pari ad 1, cioè se il trapano si trova nella posizione più bassa ovvero è stato effettuato il foro

7) **SALI** :

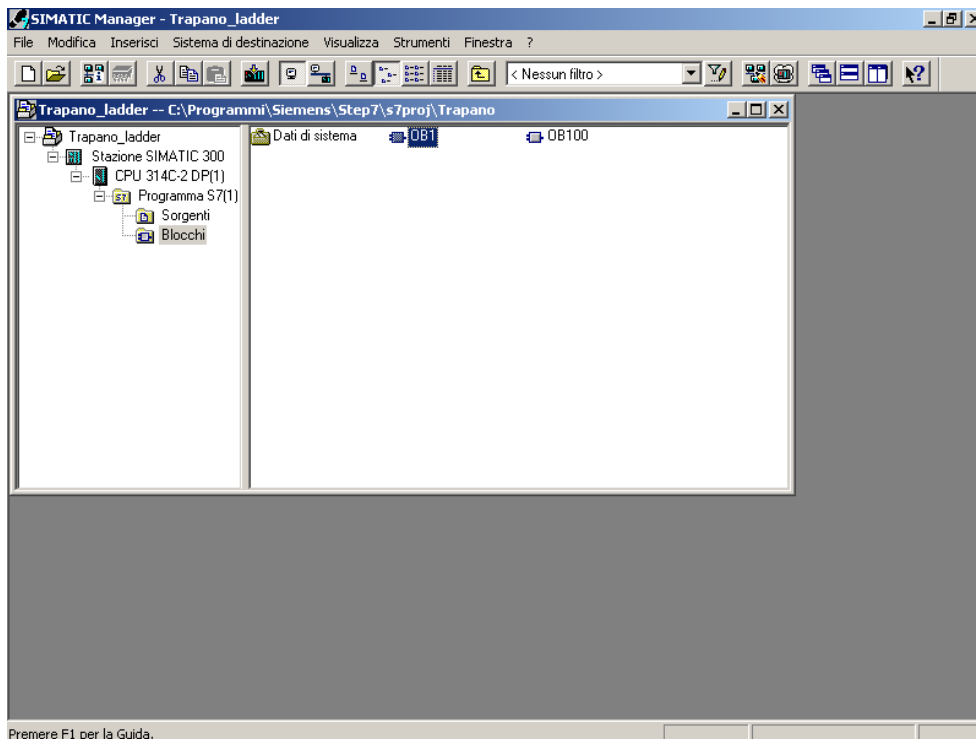
ROTATE = 1

UP = 1

In questo stato il trapano continua a ruotare ed inizia a risalire perché ha completato il foro sul pezzo. Il passaggio allo stato di Home avviene se la variabile di ingresso A, è posta pari ad 1, cioè se il trapano si trova nella posizione più alta.

2.1.1 Progetto Trapano realizzato in linguaggio KOP (Ladder)

Aprire il Simatic Manager e avviare la procedura **File > Assistente "Nuovo progetto"**. Scegliere la CPU314C-2DP (si ricordi che nella configurazione hardware bisogna impostare la versione firmware 2.0), selezionare OB1 e OB100, scegliere come linguaggio di programmazione KOP e indicare il nome del progetto "trapano_ladder". Si otterrà:



Per effettuare la programmazione con nomi simbolici bisogna creare una tabella dei simboli. Aprire la cartella "Simboli" che si trova dentro **Programma S7(1)** e inserire le varie righe.

La sintassi per le variabili di ingresso/uscita è:

I 124.0 / Q 124.0

"I" sta ad indicare che ci riferiamo ad un ingresso, "Q" ad un'uscita. Questo secondo il linguaggio mnemonico inglese.

"124.0" rappresenta l'indirizzo di default dell'ingresso/dell'uscita integrato che si ricava dalle caratteristiche tecniche della CPU utilizzata.

In particolare per la seguente CPU gli indirizzi di default integrati sono:

- Ingressi digitali da 124.0 a 126.7
- Uscite digitali da 124.0 a 125.7
- Ingressi analogici da 752 a 755
- Uscite analogiche da 752 a 755

Stato	Simbolo	Indirizzo	Tipo di dati	Commento
1	START	I 124.0	BOOL	
2	Y	I 124.1	BOOL	
3	A	I 124.2	BOOL	
4	M	I 124.3	BOOL	
5	B	I 124.4	BOOL	
6	X	I 124.5	BOOL	
7	home	M 20.0	BOOL	
8	scendi	M 20.1	BOOL	
9	foro	M 20.2	BOOL	
10	foro2	M 20.3	BOOL	
11	sali	M 20.4	BOOL	
12	sali2	M 20.5	BOOL	
13	olio	M 20.6	BOOL	
14	COMPLETE RESTART	OB 100	OB 100	Complete Restart
15	rotate	Q 124.0	BOOL	
16	down	Q 124.1	BOOL	
17	up	Q 124.2	BOOL	
18	oil	Q 124.3	BOOL	
19	TON	T 0	TIMER	
20				

Premere F1 per accedere alla Guida. NUM

Si può osservare inoltre, che agli stati vengono associati bit di Merker a partire da M 20.0. (Si potevano scegliere anche a partire da MB 17.0). Questa scelta è legata al fatto che la CPU utilizzata, ha i byte di Merker da MB 0 a MB 16 con ritenzione preimpostata. Si ricorda che i dati vengono definiti a ritenzione quando, dopo una caduta di tensione, essi hanno lo stesso valore che avevano prima della caduta di tensione.

Affinché il programma funzioni correttamente è necessario che la variabile Home sia settata a true. Avendo assegnato agli stati tali indirizzi si è garantito che tali variabili non siano ritentive. In tal modo se dovesse verificarsi una caduta di tensione il programma ripartirebbe correttamente, perché la variabile Home in particolare non assumerebbe il vecchio valore essendo non ritentiva (In realtà bastava impostare solo la variabile Home non ritentiva!).

Dopo aver creato la tabella dei simboli cliccare su OB1 per scrivere il programma in KOP(Ladder). Si aprirà la maschera "KOP/AWL/FUP" dove si inseriranno le varie istruzioni in linguaggio KOP(Ladder). Il programma si articola in segmenti (rung). Quindi bisogna selezionare "inserisci segmento" per introdurre i vari rung dove andranno inseriti i simboli del linguaggio ladder. Sulla sinistra si trova la finestra "elementi di programma". Questa contiene tutta la simbologia relativa al linguaggio KOP(ladder).

Ogni qualvolta si inserisce un elemento (contatto aperto, contatto chiuso, temporizzatore, ecc.) esso presenterà dei punti interrogativi rossi. Cliccando su di essi è possibile inserire l'indirizzo assoluto oppure se si è costruita la tabella dei simboli è possibile scegliere il nome nella casella di riepilogo visualizzata.

Il programma in KOP sarà composto da tre fasi. A partire dalla macchina di stato bisogna scrivere questa parti di codice:

1. stato attuale – ingresso – nuovo stato
2. nuovo stato – resettaggio vecchio stato
3. stato attuale – condizioni che devono essere compiute

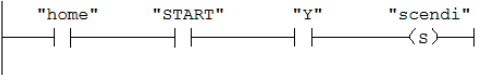
OB1 - <offline>

""
Nome: Famiglia:
Autore: Versione: 0.1
Versione blocco: 2
Data e ora Codice: 17/02/2005 17.53.06
Interfaccia: 15/02/1996 16.51.12
Lunghezza (blocco / codice / dati) 00288 00134 00022

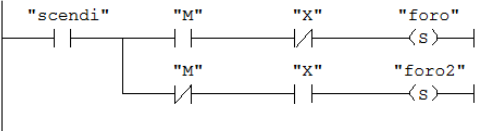
Nome	Tipo di dati	Indirizzo	Commento
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Blocco:OB1 "Main Program Sweep (Cycle)"

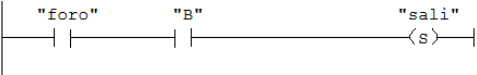
Segmento: 1 Prima parte: Avviamento Trapano



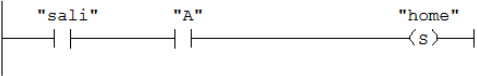
Segmento: 2 Abbassamento della punta del trapano
Distinguiamo il caso del pezzo piccolo e di quello grande



Segmento: 3
Se il pezzo è piccolo faccio il foro



Segmento: 4
risalgo allo stato di partenza



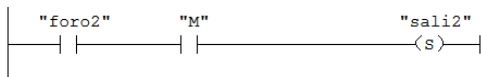
SIMATIC

OB1

22/02/2005 12.24.28

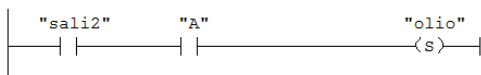
Segmento: 5

In questo caso il pezzo è grosso



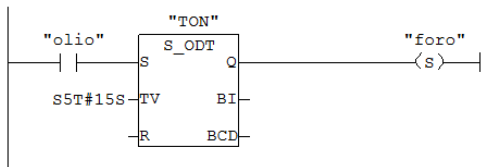
Segmento: 6

Devo oleare la punta del trapano per farlo raffreddare



Segmento: 7

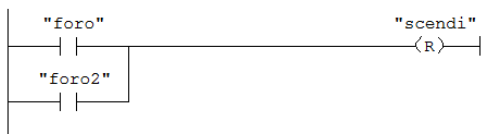
dopo un certo tempo faccio il foro



Segmento: 8 seconda parte:resetto Gli stati Precedenti



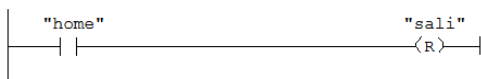
Segmento: 9



Segmento: 10



Segmento: 11



SIMATIC

OB1

22/02/2005 12.24.28

Segmento: 12



Segmento: 13



Segmento: 14



Segmento: 15 Terza parte- definisco le azioni



Segmento: 16



Segmento: 17

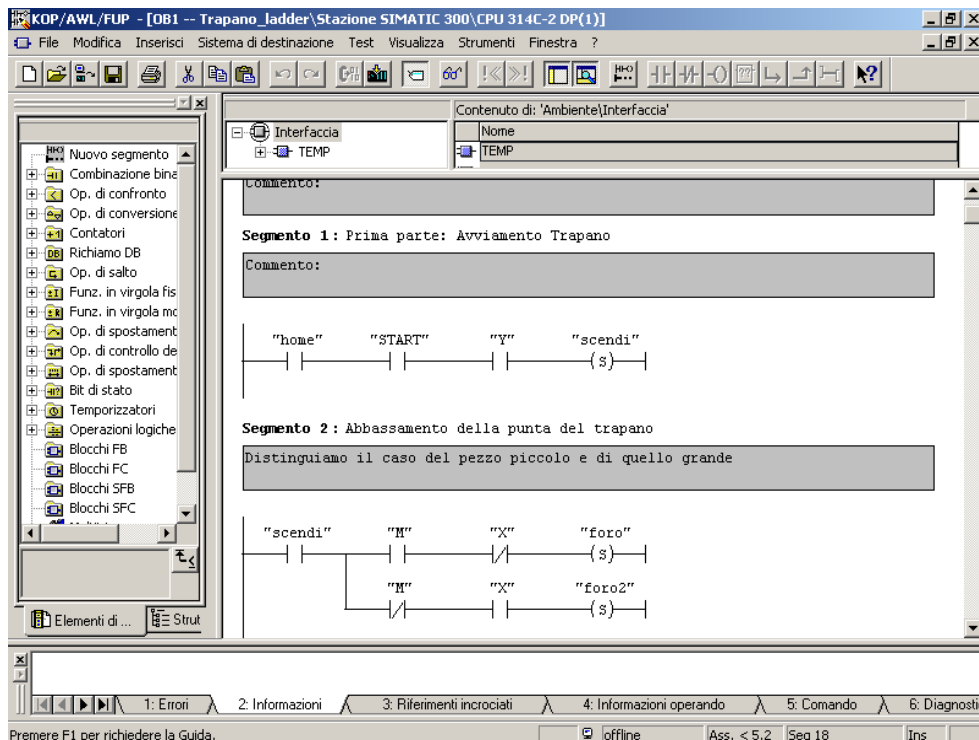


Segmento: 18





Di seguito viene riportato un esempio dell'interfaccia "KOP/AWL/FUP".



Dopo aver salvato il programma in KOP cliccare su OB100 per inserire le istruzioni relative all'avviamento.

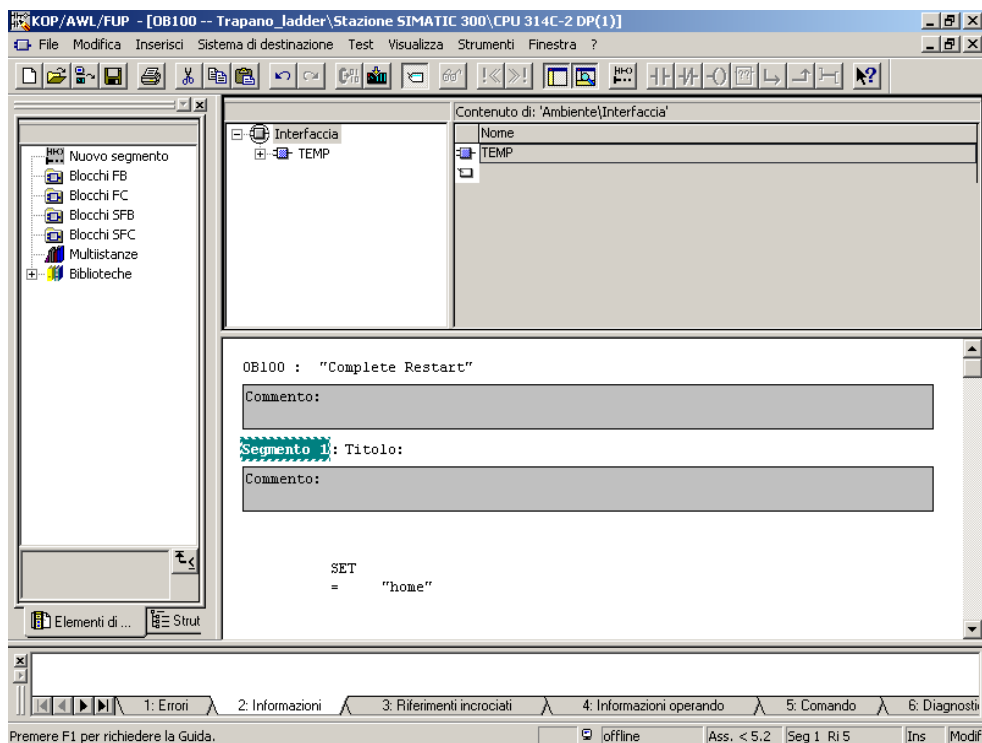
Nel caso in esame è necessario settare la variabile home a true.

Per creare il blocco organizzativo di avviamento cliccare su OB100, scelto in fase di creazione, e dopo aver scelto dal menu **Visualizza > AWL** inserire la seguente istruzione, che al momento dell'avviamento setta home a true:

SET





= "home"


Si ricorda che AWL (lista istruzioni) è un linguaggio di programmazione testuale vicino al linguaggio macchina. Quando si crea un programma in AWL, le singole istruzioni corrispondono in larga misura alle operazioni con le quali la CPU elabora il programma.



Simulazione del progetto Trapano con S7-PLCSIM

Dopo aver salvato il progetto si avvia S7-PLCSIM per far partire la simulazione.

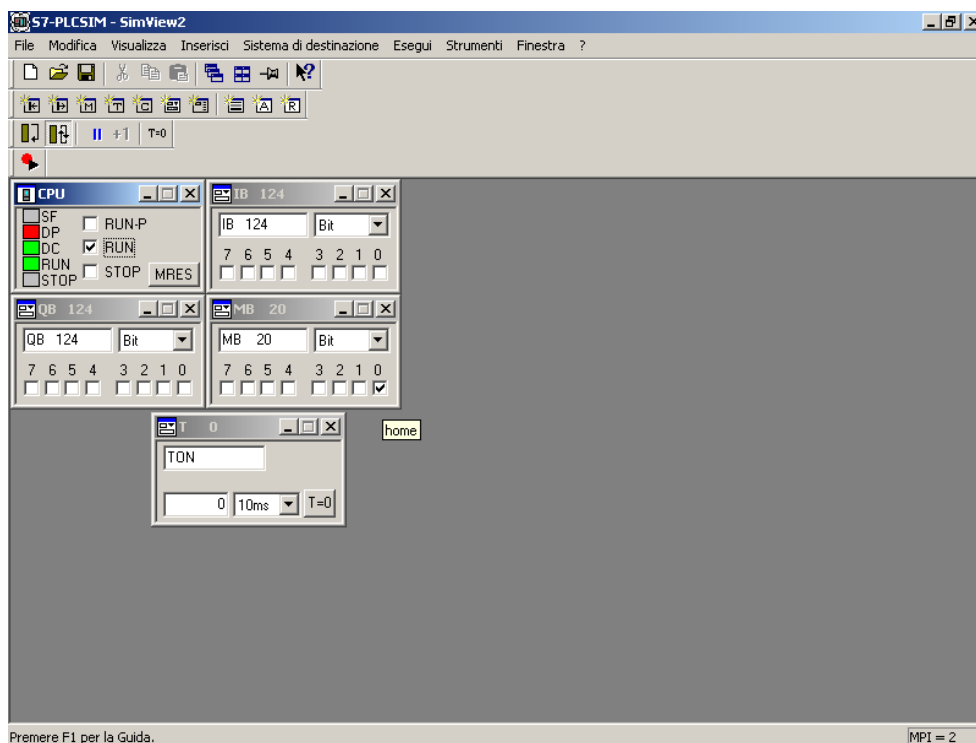
1. Aprire il SIMATIC Manager.
2. Fare clic su  oppure selezionare il comando di menu **Opzioni > Simula unità**. Questa operazione avvia l'applicazione S7-PLCSIM ed apre una sottofinestra CPU (indirizzo MPI di default: 2).
3. Nel SIMATIC Manager, aprire il progetto "Trapano_ladder".
4. All'interno del progetto portarsi sull'oggetto "Blocchi".
5. Nel SIMATIC Manager, fare clic su  oppure selezionare il comando di menu **Sistema di destinazione > Carica** per caricare l'oggetto "Blocchi" nel PLC simulato.
6. A questo punto, il sistema chiede se si vogliono caricare i dati di sistema; rispondere "NO" se non si intende caricare la configurazione hardware nel PLC simulato, SÌ per caricare la configurazione hardware.
7. Nell'applicazione S7-PLCSIM, creare ulteriori "sottofinestre" per controllare le informazioni dal PLC simulato:
 - Fare clic su  oppure selezionare il comando di menu **Inserisci > Ingresso**. Digitare **IB124** (byte di ingresso 124).
 - Fare clic su  oppure selezionare il comando di menu **Inserisci > Uscita**. Digitare **QB124** (Output Byte 124).

- Fare clic su  oppure selezionare il comando di menu **Inserisci>Temporizzatore**. Digitare **TON**
- Selezionare il comando di menu **Inserisci> Merker**. Digitare **MB 20**

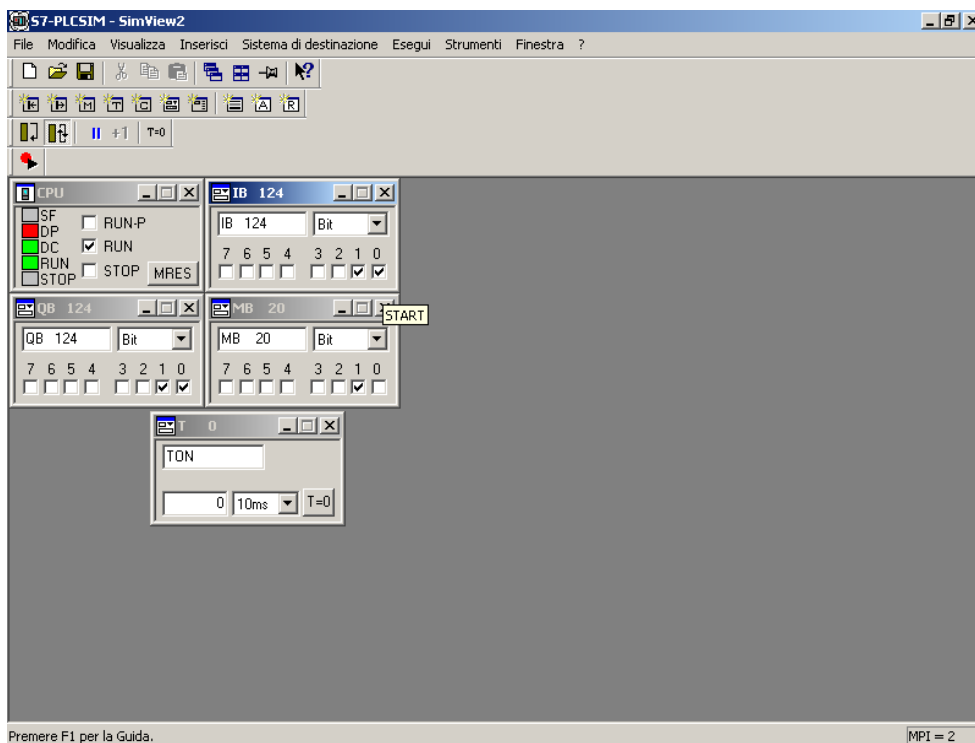
8. Selezionare il menu **Sistema di destinazione** in S7-PLCSIM e verificare che accanto ad **Alimentazione** compaia un pallino (•).

9. Selezionare **Esegui > modo di scansione** e verificare che compaia un pallino (•) accanto a **Ciclo continuo**.


10. Commutare la CPU simulata in RUN facendo clic sulla casella RUN oppure RUN-P check. Dall'immagine si osserva che quando la CPU va in RUN la variabile home è true.



11. Fare clic sul bit 0 e sul bit 1 di IB124 per simulare l'attivazione degli ingressi start e y ed osservare gli effetti di questa operazione su QB124.




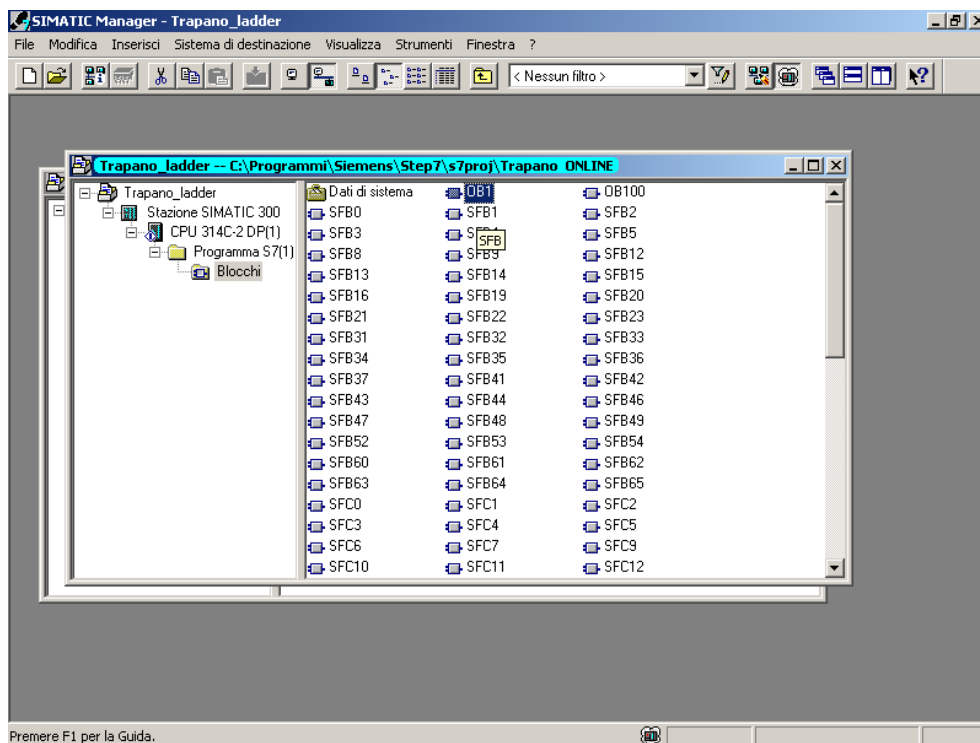
È possibile associare ad ogni elemento (IB,MB,QB, T0) il corrispondente simbolo della tabella dei simboli. Bisogna cliccare su **Strumenti > opzioni > aggiungi simboli**. Si aprirà la maschera “Apri” che permette di selezionare la tabella dei simboli relativa al progetto in esame.

12. Fare clic su  oppure selezionare il comando di menu **File > Salva PLC con nome...** per salvare come nuovo file lo stato corrente del PLC simulato.

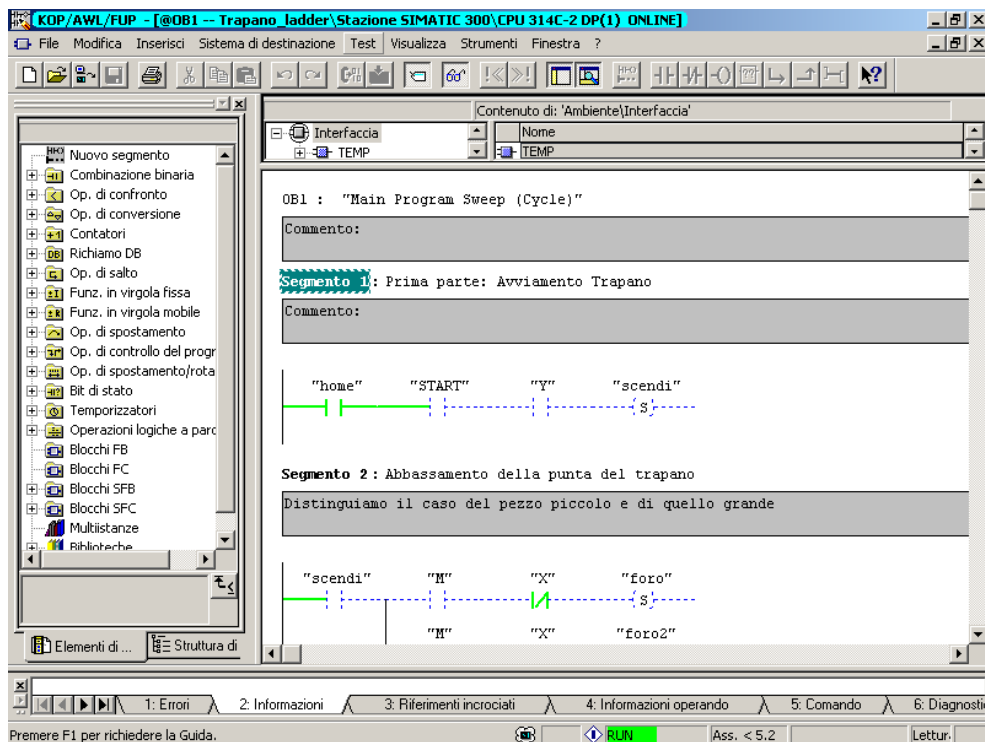
Come utilizzare STEP 7 per controllare la simulazione del programma

È possibile utilizzare i tool di STEP 7 per controllare la simulazione del programma:

1. Dopo aver creato le sottofinestre (vedere sopra al punto 7), entrare nel SIMATIC Manager.
2. Fare clic su  oppure selezionare il comando **Visualizza > Online** per passare al modo online.

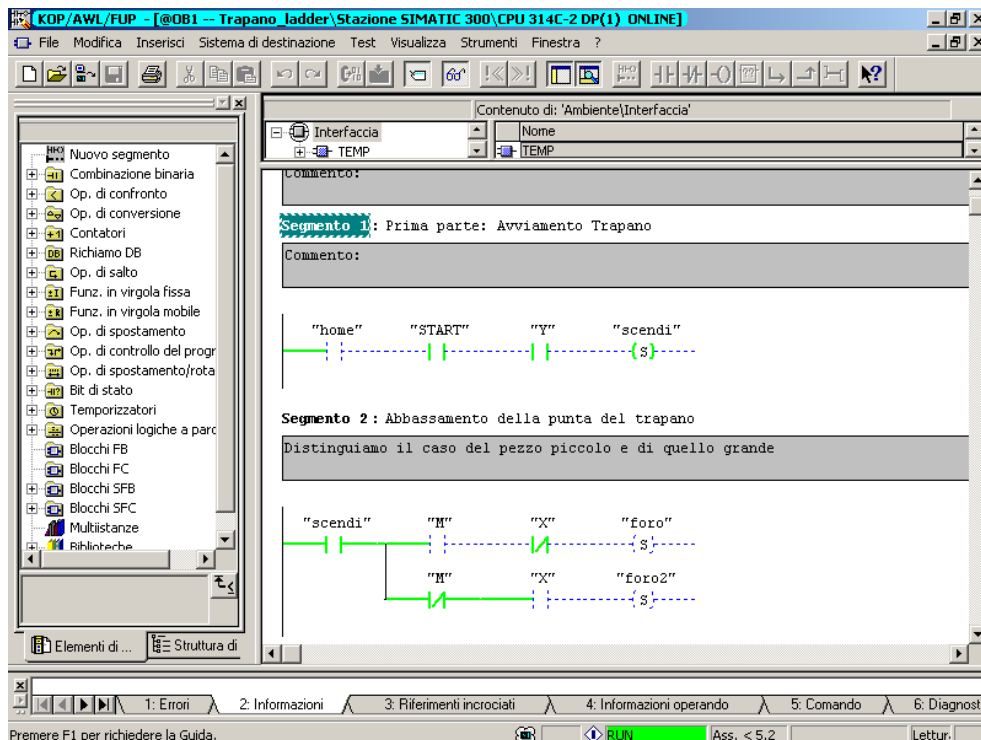


3. Spostarsi all'oggetto "Blocchi" nel progetto trapano_ladder ed aprire OB1. Questa operazione richiama l'applicazione "KOP/AWL/FUP".
4. Quando la CPU simulata si trova in modo RUN ed sono stati attivati i bit 0 e 1 di IB124 (vedere sopra al punto 11), passare alla finestra dell'applicazione "KOP/AWL/FUP" e selezionare **Test > Controlla** per verificare l'effetto di tali modifiche sul programma. Questa prima immagine mostra quello che accade quando la CPU è commutata in RUN



In questa immagine viene mostrato cosa accade quando i bit "START" e "Y" vengono attivati.

Si verifica il passaggio allo stato successivo."scendi".



Procedendo alla stessa maniera è possibile verificare l'intero programma.

Per uscire dalla simulazione basta cliccare su **Test > Controlla** e chiudere la finestra del collegamento online.

2.1.2 Progetto Trapano realizzato in linguaggio SFC

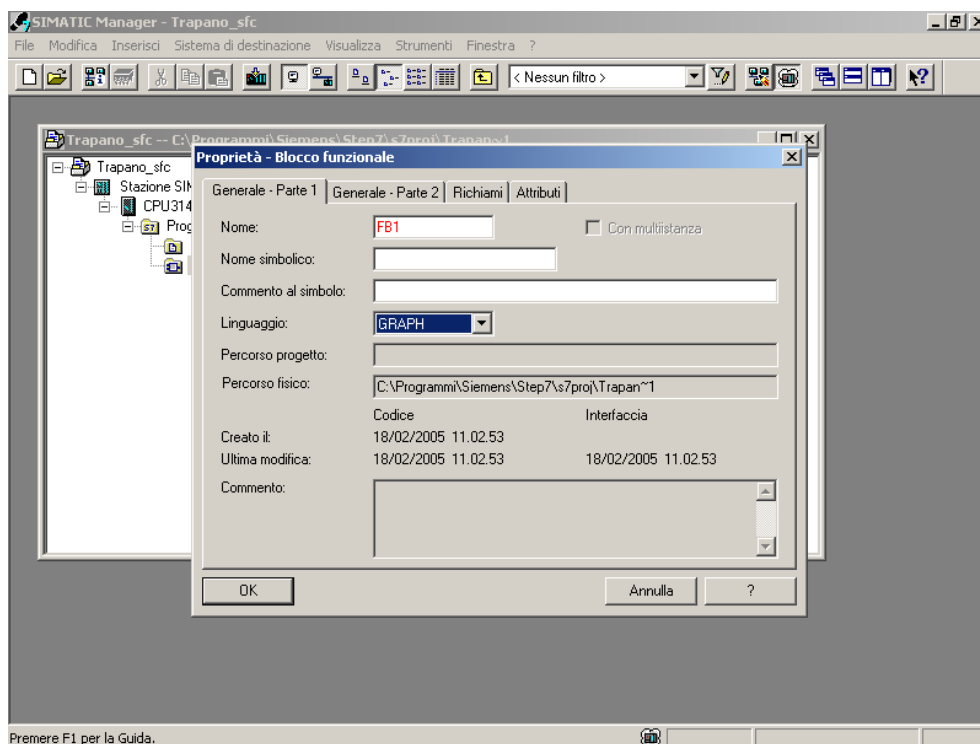
Per realizzare il progetto del trapano in SFC nell'ambiente di programmazione STEP7 bisogna utilizzare il programma S7-GRAPH (Vedi paragrafo 1.10).

Nella prima fase bisogna aprire il Simatic Manager e richiamare il comando di menu **File > Assistente "Nuovo progetto"**. Creare un nuovo progetto "Trapano_sfc". In questo caso l'OB100 non è necessario. Se si desidera usare la notazione simbolica bisogna creare la tabella dei simboli. Inizialmente è necessario inserire solo gli indirizzi degli ingressi e delle uscite. Non è necessario definire gli stati.

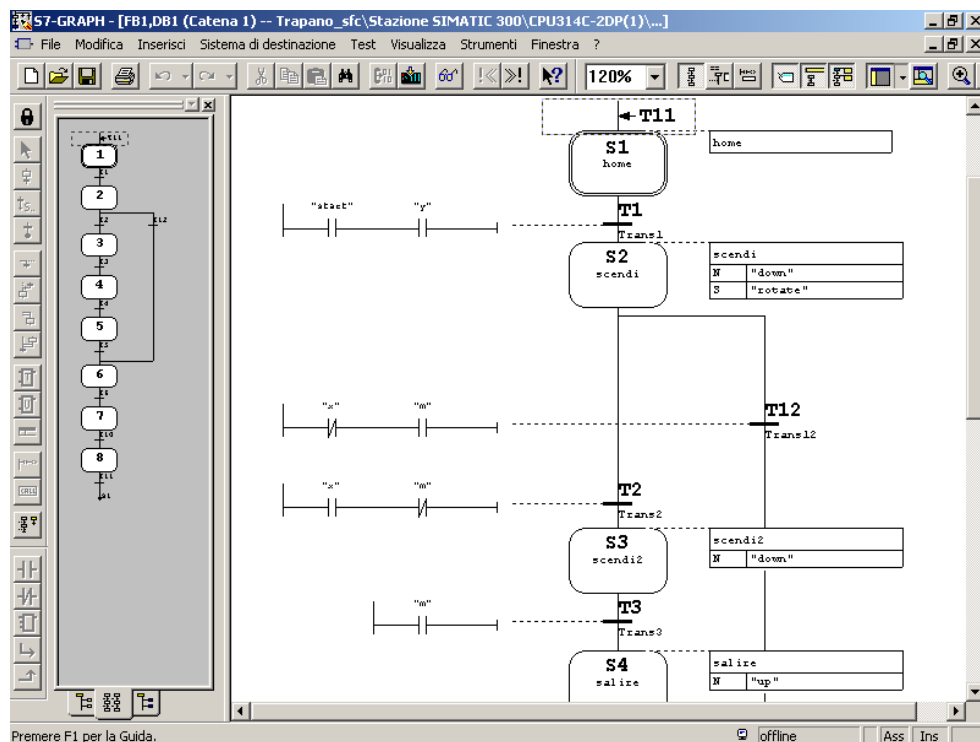
In figura viene riportata la tabella dei simboli che si ottiene dopo aver compilato il file di S7_GRAPH.

Stato	Simbolo	Indirizzo	Tipo di dati	Commento
1	S7_STD_3	FC 72	FC 72	
2	start	I 124.0	BOOL	
3	y	I 124.1	BOOL	
4	b	I 124.2	BOOL	
5	a	I 124.3	BOOL	
6	x	I 124.4	BOOL	
7	m	I 124.5	BOOL	
8	INIT_SQ	M 20.0	BOOL	
9	programma trapano	OB 1	OB 1	
10	rotate	Q 124.0	BOOL	
11	down	Q 124.1	BOOL	
12	up	Q 124.2	BOOL	
13	oil	Q 124.3	BOOL	
14	TIME_TCK	SFC 64	SFC 64	Read the System Time
15				

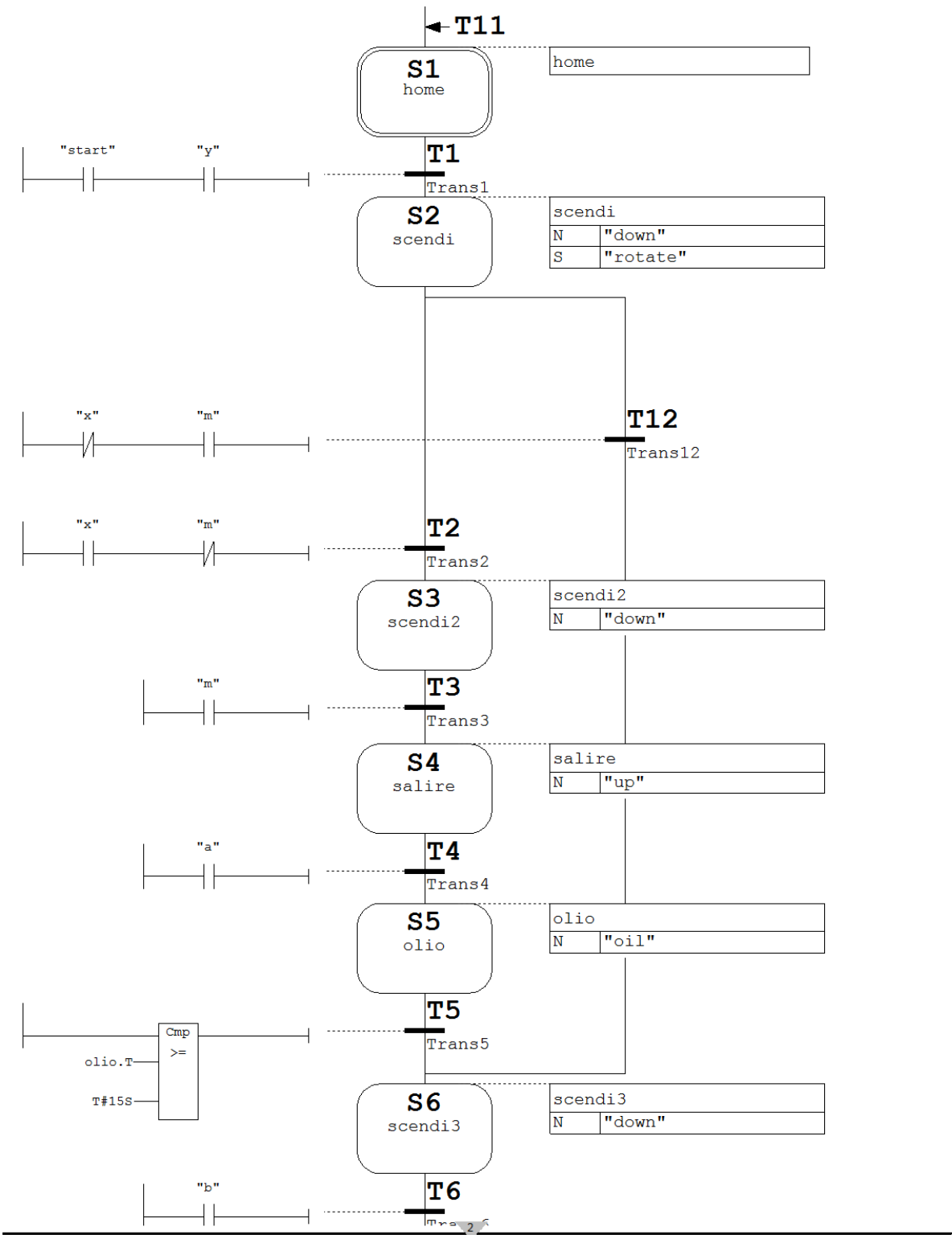
Dopo aver creato il progetto e aver definito la tabella dei simboli portarsi dentro "Blocchi" e con il tasto destro del mouse scegliere **Inserisci nuovo oggetto > Blocco funzionale**. Si aprirà la finestra "Proprietà-Blocco funzionale" e nella casella linguaggio dobbiamo scegliere "GRAPH".

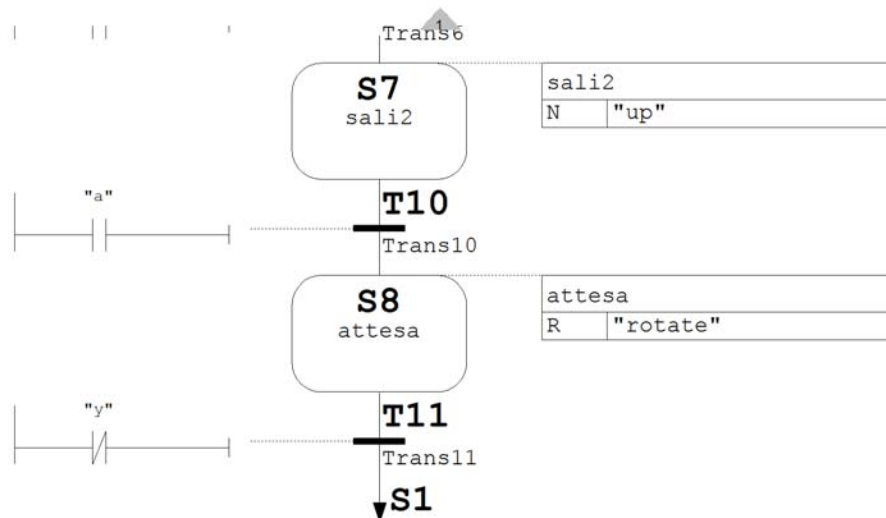


Cliccando su FB1 si apre l'editor di S7-GRAPH. Di seguito viene mostrata l'interfaccia dell'editor con una parte del codice SFC composto da passi, transizioni, azioni e diramazioni alternative.



Progetto "trapano_sfc".





Step Home :

Allo step iniziale, contrassegnato con una doppia linea, non è associato alcuna azione.

Transizione T1:

Alla prima transizione vengono associati due contatti start e y.

Step scendi :

Allo step scendi associamo l'azione:

- N “down” questa indica che finché il passo è attivo down è 1
- S “rotate” questa indica che appena il passo è attivo rotate viene impostato a 1 e rimane così impostato

Transizione T2 :

Alla seconda transizione associamo un contatto normalmente aperto x e un contatto normalmente chiuso m

Step scendi2 :

A tale step associamo l'azione N "down" questa indica che finché il passo è attivo down è 1

Transizione T3:

Alla terza transizione associamo il contatto m.

Step salire :

A tale step associamo l'azione N "up" questa indica che finché il passo è attivo up è 1.

Transizione T4 :

Alla quarta transizione associamo il contatto a.

Step olio :

A questo step associamo l'azione N "oil" questa indica che finché il passo è attivo oil è 1.

Transizione T5:

Alla quinta transizione associamo un comparatore dove confrontiamo olio.T, che rappresenta il tempo di attivazione corrente o più recente del passo, con un tempo prefissato di 15 secondi (T#15S). Questo equivale a fissare il tempo necessario per oleare la punta del trapano.

Esaminiamo adesso l'altro percorso (diramazione alternativa).

Transizione T12:

A questa transizione è associato il contatto normalmente chiuso x e il contatto normalmente aperto m.

A questo punto si ha un “percorso convergente” in quanto le due sequenze alternative convergono.

Step scendi3 :

A tale step associamo l'azione N “down” questa indica che finché il passo è attivo down è 1.

Transizione T6:

A questa transizione è associato il contatto b.

Step Sali2 :

A tale step associamo l'azione N “up” questa indica che finché il passo è attivo up è 1

Transizione T10:

A tale transizione associo un contatto normalmente aperto a.

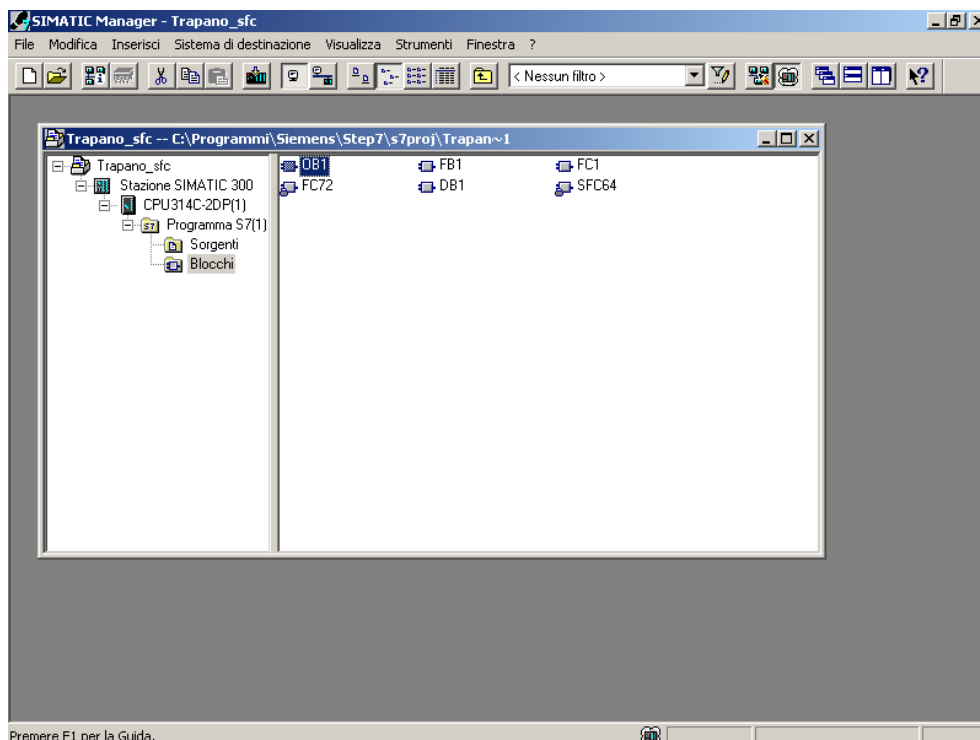
Step attesa :

A questo step associamo l'azione R “rotate”. Appena il passo è attivo rotate viene impostato a 0 e rimane così impostato

Transizione T11:

A tale transizione associamo il contatto negato y. Si ritorna allo stato di home.

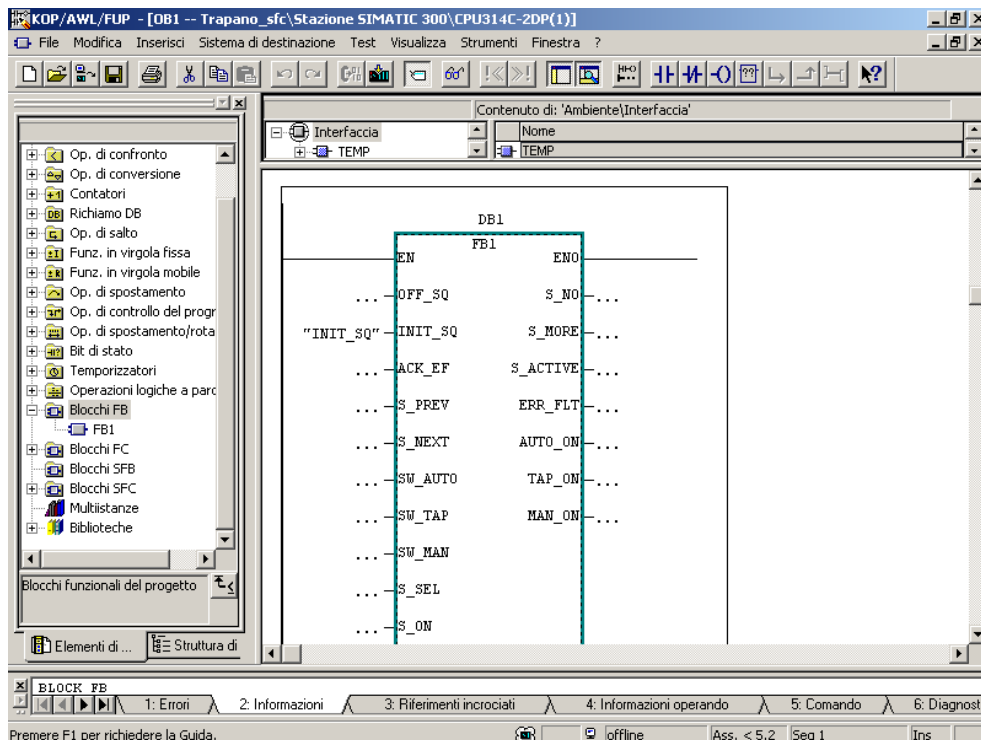
Dopo aver salvato e compilato il progetto realizzato in S7_GRAPH, si troveranno nella cartella “Blocchi” i seguenti elementi generati dalla compilazione.




Per realizzare il richiamo di FB1 all'interno del blocco organizzativo 1 è necessario aprire OB1 e dalla finestra "elementi di programma", posizionarsi su "Blocchi" e scegliere FB1. Introdurre al posto dei punti interrogativi DB1.

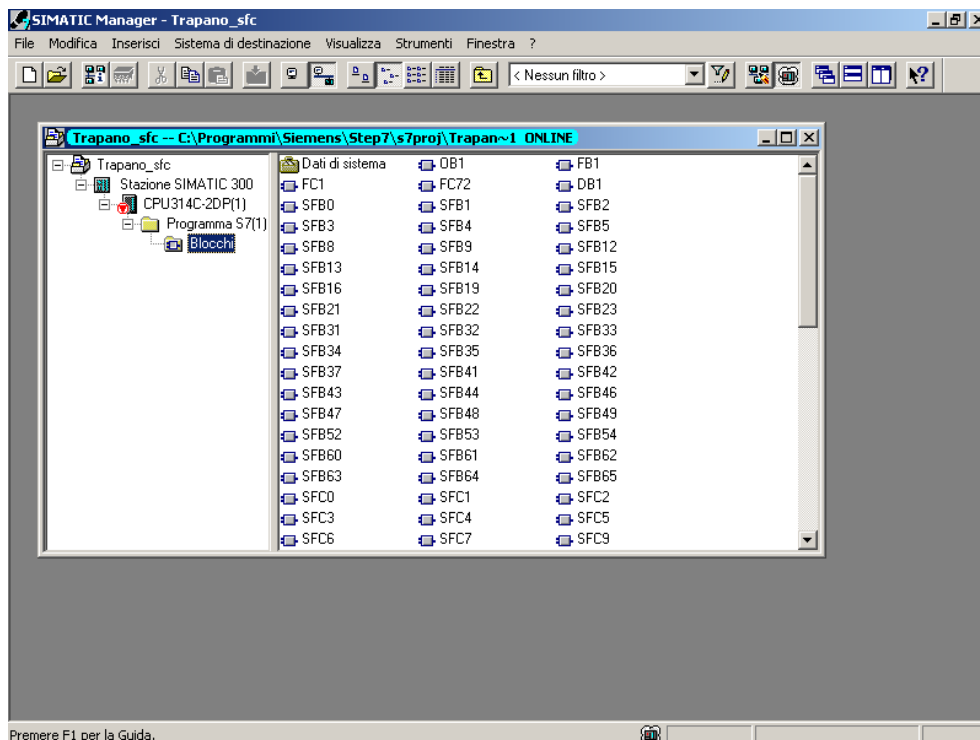
Si ricorda che i blocchi dati di istanza(DB), generati automaticamente nella compilazione, vengono assegnati al blocco quando viene richiamato un FB.

Associare a INIT_SQ (init_sequence) la variabile creata nella tabella dei simboli (INIT_SQ M20.0). Questa consente di inizializzare la catena.



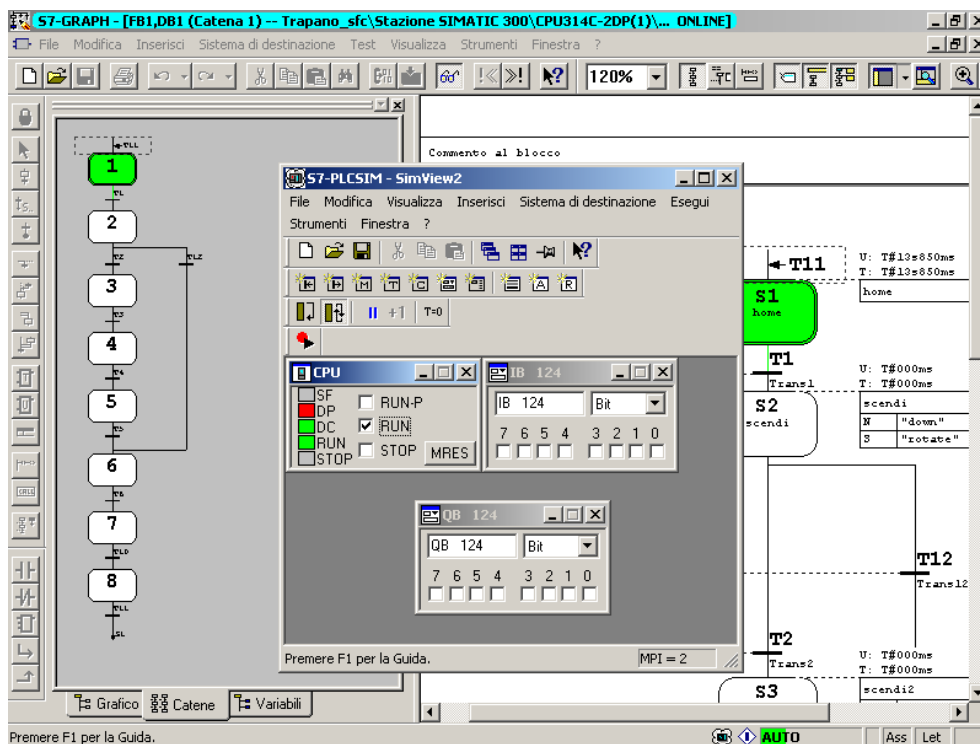
Per eseguire la simulazione del progetto:

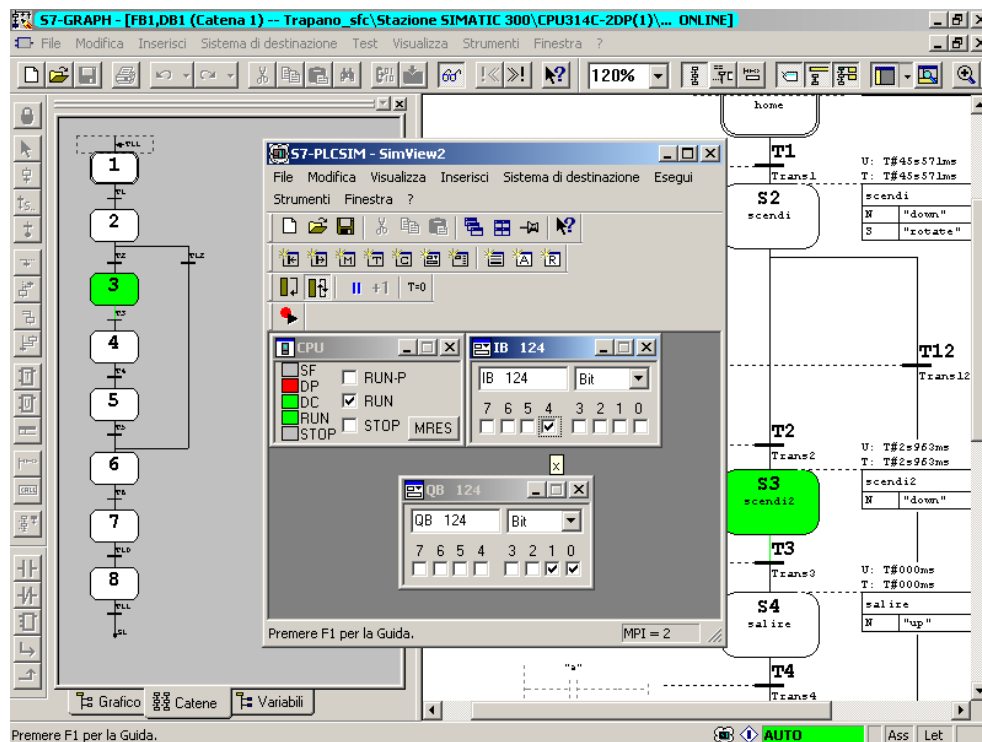
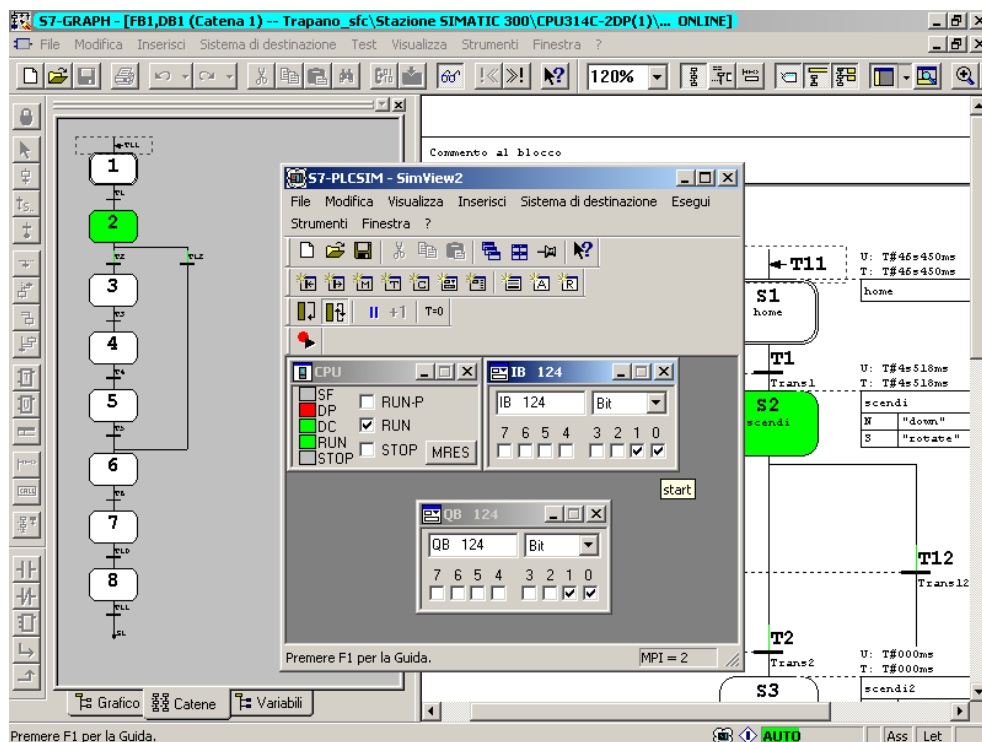
1. seguire gli stessi passi del paragrafo **1.8 Simulazione con S7-PLCSIM**
2. cliccare su  e avviare FB1 (non OB1)

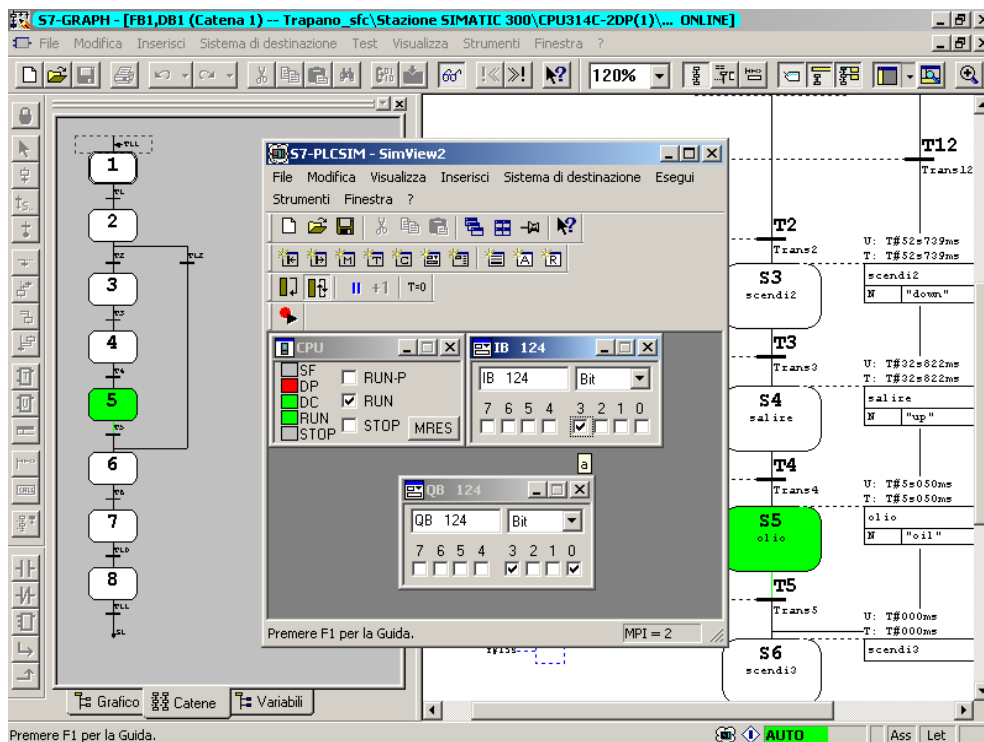


3. Commutare la CPU o il simulatore in RUN e attivare i vari ingressi.

Di seguito si riportano alcune immagini che sintetizzano alcuni passi della simulazione.



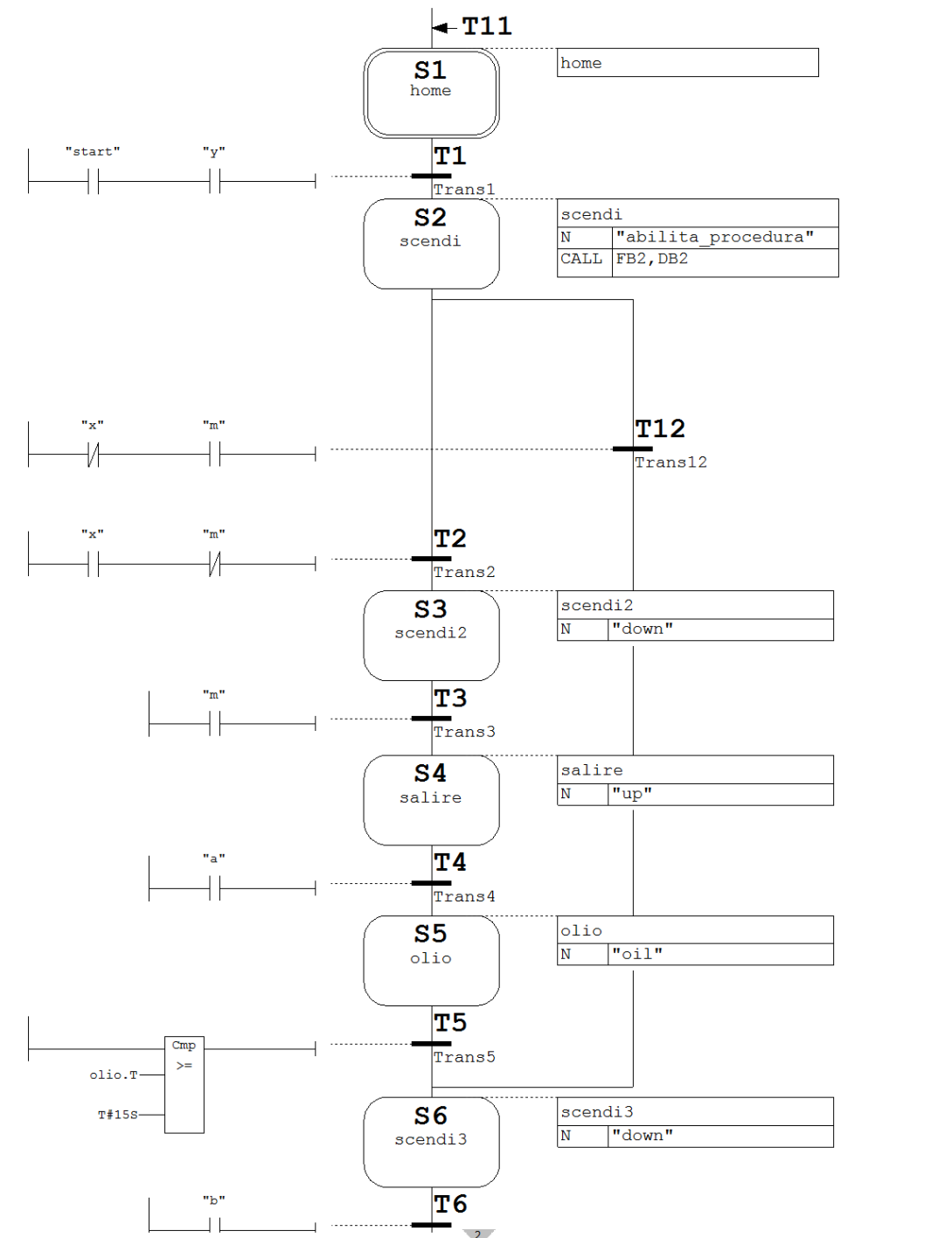


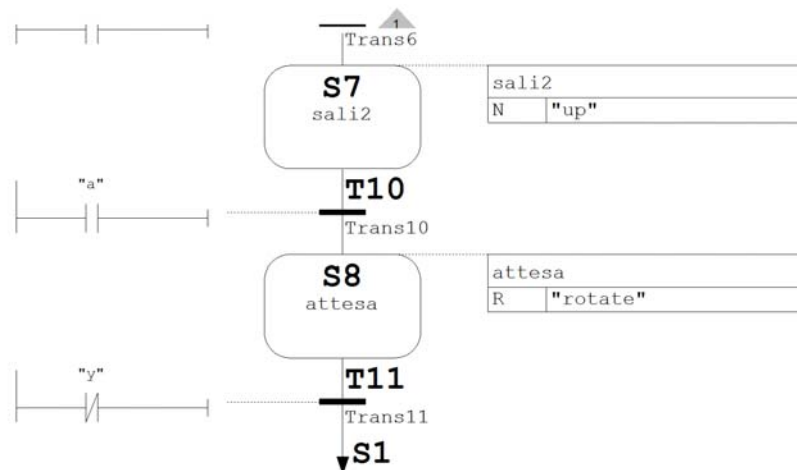


2.1.3 Progetto Trapano realizzato in linguaggio SFC con la chiamata di funzione

Viene riproposto l'esercizio del trapano del paragrafo 2.1.2 per far vedere come si implementa la chiamata di blocchi attraverso l'istruzione "CALL".

Dopo aver eseguito le operazioni di creazione del progetto (vedi paragrafo 2.1.2), introdurre all'interno dell'S7_GRAPH il seguente codice SFC in cui sono state apportate delle modifiche alle azioni del passo "Scendi".



**Step Home :**

Allo step iniziale, contrassegnato con una doppia linea, non è associata alcuna azione.

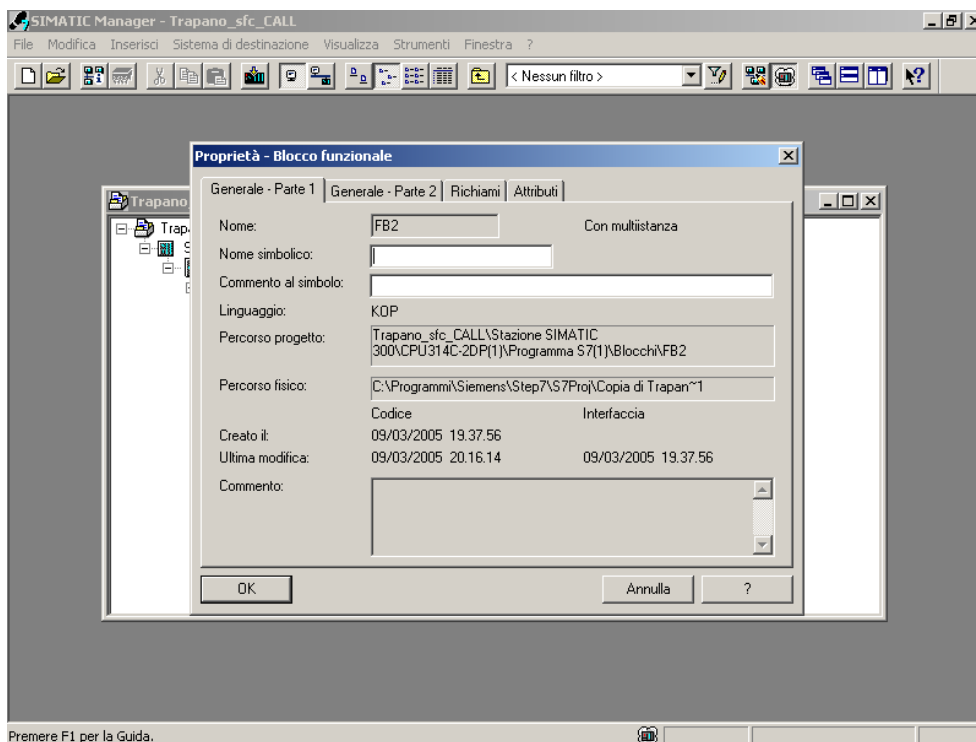
Transizione T1:

Alla prima transizione vengono associati due contatti start e y.

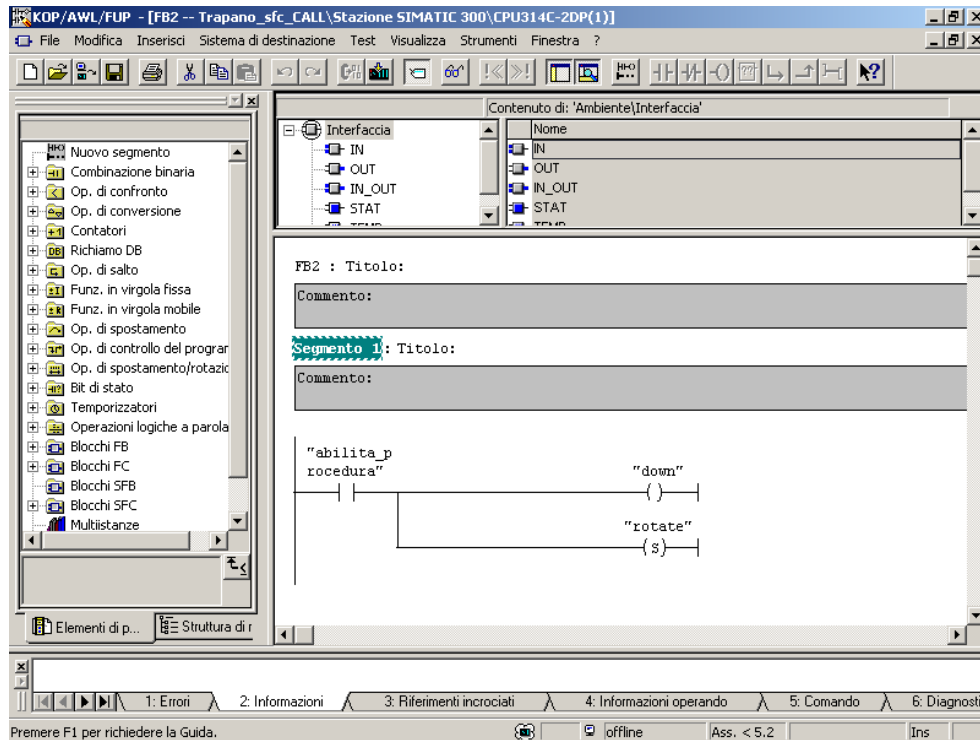
Step scendi :

In questo step si effettua un richiamo di blocco funzionale.

Per realizzarlo è necessario portarsi nella cartella Blocchi del Simatic Manager e selezionare con il tasto destro del mouse **Inserisci nuovo oggetto> blocco funzionale**. Inserire FB2 e scegliere il linguaggio di programmazione KOP.



Cliccare su FB2 per aprire la maschera “KOP/FUP/AWL” all'interno della quale si inserirà il codice che dovrà attivare le uscite down e rotate. (Ovviamente il codice è molto semplice ma serve per comprendere il meccanismo del richiamo del blocco!)

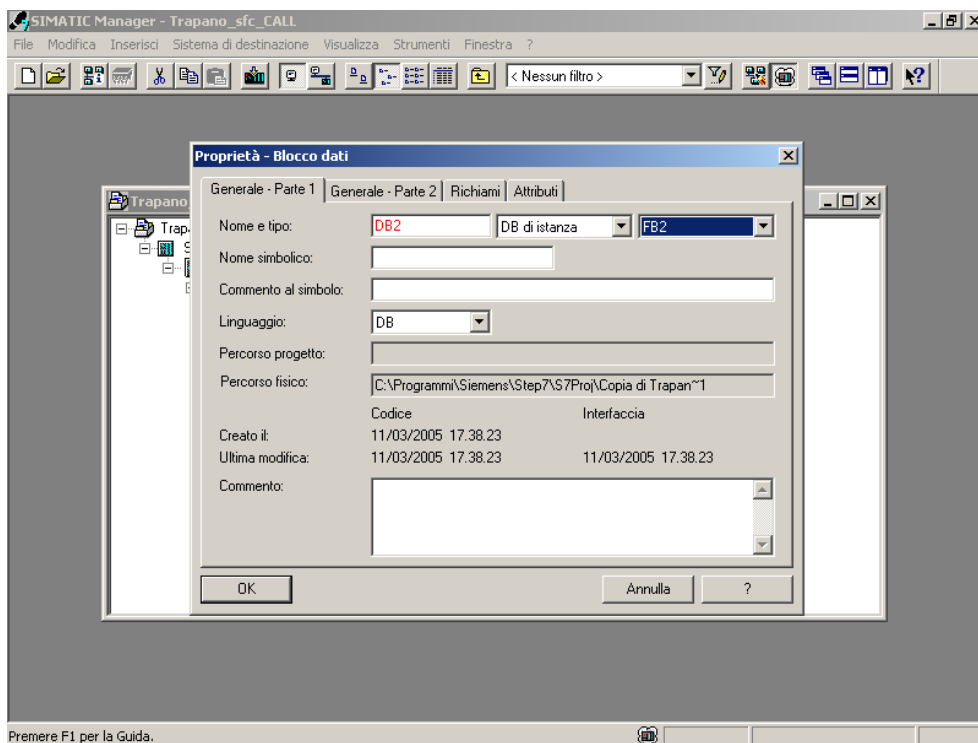


Poiché nella maschera “KOP/FUP/AWL” non è possibile inserire dei rung che contengono solo coil, è stato necessario inserire un contatto normalmente aperto “abilita_procedura”. Quando il contatto diventa 1 la variabile di uscita “down” si attiverà e la variabile di uscita “rotate” si setterà.

Dopo aver salvato il codice chiudere la maschera “KOP/FUP/AWL”, portarsi dentro la cartella “Blocchi” e creare il blocco dati di istanza da associare al blocco funzionale.

Si ricorda che un blocco funzionale è un blocco "con memoria". Esso dispone di un blocco dati correlato come memoria (blocco dati di istanza). Sia i parametri che vengono trasmessi all'FB, sia le variabili statiche vengono memorizzati nel blocco dati di istanza.

Con il tasto destro del mouse scegliere **Inserisci nuovo oggetto>blocco dati** e inserire i parametri della figura seguente



A questo punto si ritorna all'S7-GRAPH per impostare l'azione, associata allo step "scendi", che richiama il blocco FB2.

Se scrivessimo semplicemente l'azione:

- CALL FB2,DB2

al momento dell'attivazione del passo, il blocco FB2 non avrebbe la possibilità di essere richiamato perché la variabile "abilita_procedura" ha valore zero.

È necessario allora inserire, prima del richiamo del blocco, l'azione:

- N "abilita_procedura"

La quale fa assumere a tale variabile il valore 1 fino a quando il passo è attivo.

A questo punto il blocco FB2 può essere richiamato correttamente e i coil possono essere attivati.

Inoltre attribuendo ad "abilita_procedura" lo stesso significato di Si.X, che indica se il passo Si è attivo, si è risolto il problema di non poter usare, all'interno della maschera "KOP/FUP/AWL", gli operandi specifici che S7-GRAPH mette a disposizione (per es. Si.T, Si.X).

Transizione T2 :

Alla seconda transizione associamo un contatto normalmente aperto x e un contatto normalmente chiuso m

Step scendi2 :

A tale step associamo l'azione N "down" questa indica che finché il passo è attivo down è 1

Transizione T3:

Alla terza transizione associamo il contatto m.

Step salire :

A tale step associamo l'azione N "up" questa indica che finché il passo è attivo up è 1.

Transizione T4 :

Alla quarta transizione associamo il contatto a.

Step olio :

A questo step associamo l'azione N "oil" questa indica che finché il passo è attivo oil è 1.

Transizione T5:

Alla quinta transizione associamo un comparatore dove confrontiamo olio.T, che rappresenta il tempo di attivazione corrente o più recente del passo, con un tempo prefissato di 15 secondi (T#15S). Questo equivale a fissare il tempo necessario per oleare la punta del trapano.

Esaminiamo adesso l'altro percorso (diramazione alternativa).

Transizione T12:

A questa transizione è associato il contatto normalmente chiuso x e il contatto normalmente aperto m.

A questo punto si ha un "percorso convergente" in quanto le due sequenze alternative convergono.

Step scendi3 :

A tale step associamo l'azione N "down" questa indica che finché il passo è attivo down è 1

Transizione T6:

A questa transizione è associato il contatto b.

Step Sali2 :

A tale step associamo l'azione N "up" questa indica che finché il passo è attivo up è 1

Transizione T10:

A tale transizione associo un contatto normalmente aperto a.

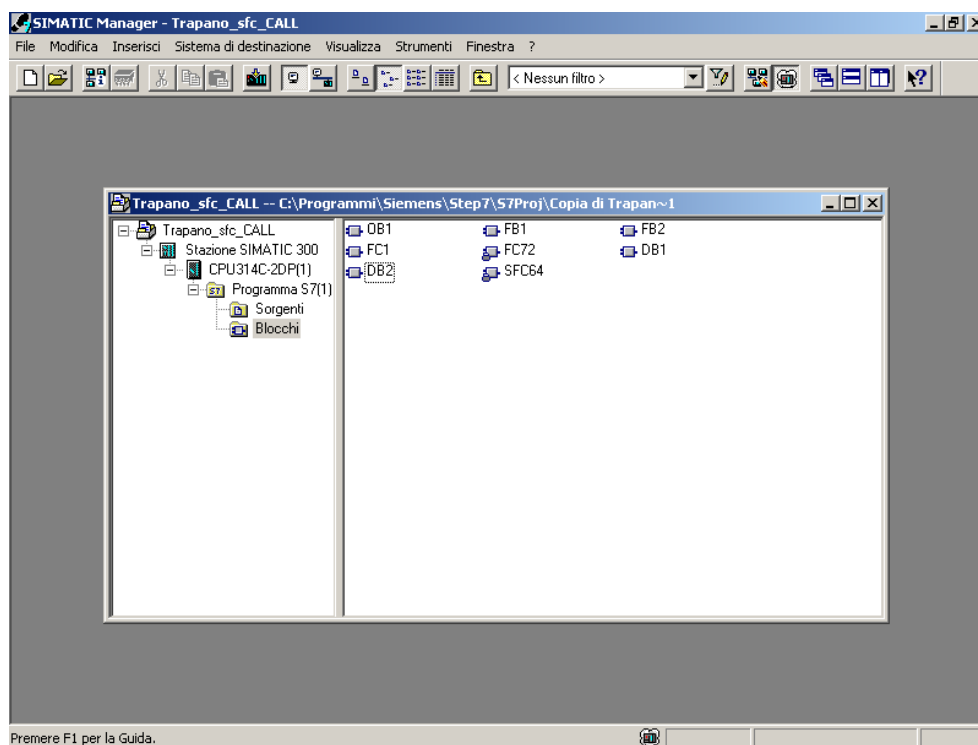
Step attesa :

A questo step associamo l'azione R "rotate". Appena il passo è attivo rotate viene impostato a 0 e rimane così impostato

Transizione T11:

A tale transizione associamo il contatto negato y. Si ritorna allo stato di home.

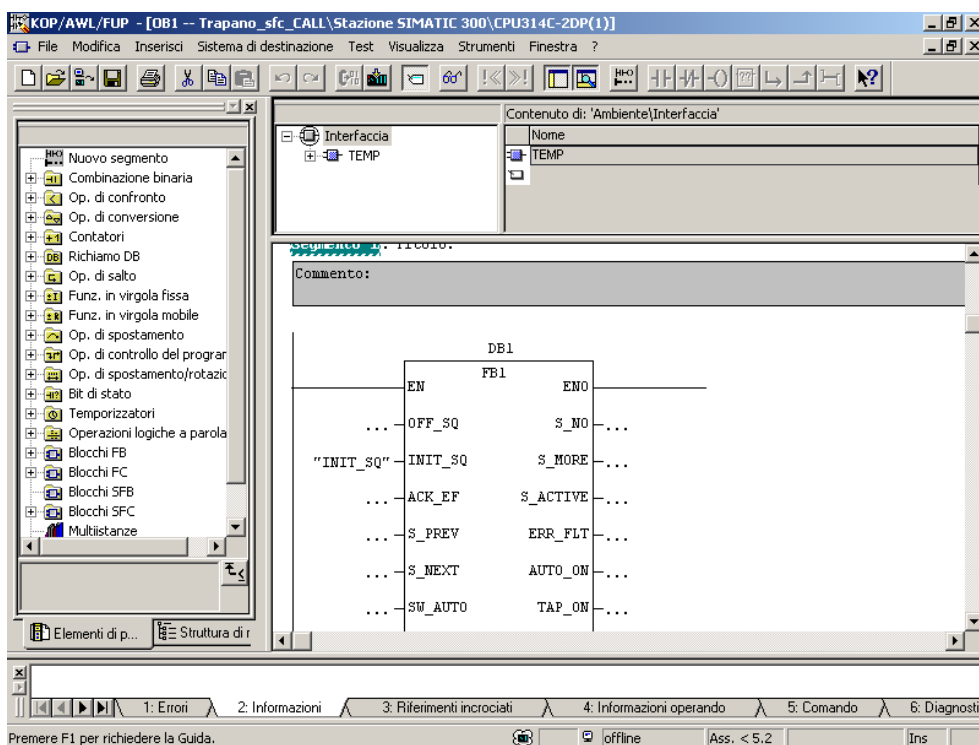
Dopo aver salvato e compilato il progetto realizzato in S7_GRAPH, si troveranno nella cartella "Blocchi" i seguenti elementi generati dalla compilazione.



Per realizzare il richiamo di FB1 all'interno del blocco organizzativo 1 è necessario aprire OB1 e dalla finestra "elementi di programma", posizionarsi su "Blocchi" e scegliere FB1. Introdurre al posto dei punti interrogativi DB1.

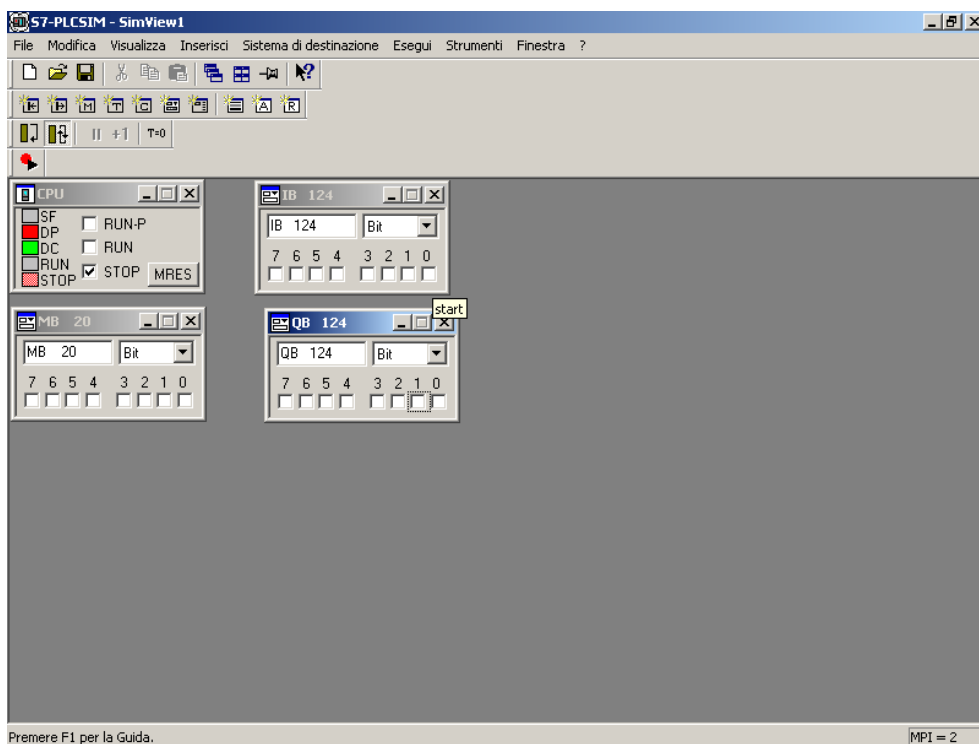
Si ricorda che i blocchi dati di istanza(DB), generati automaticamente nella compilazione, vengono assegnati al blocco quando viene richiamato un FB.


Associare a INIT_SQ (init_sequence) la variabile creata nella tabella dei simboli (INIT_SQ M20.0). Questa consente di inizializzare la catena.

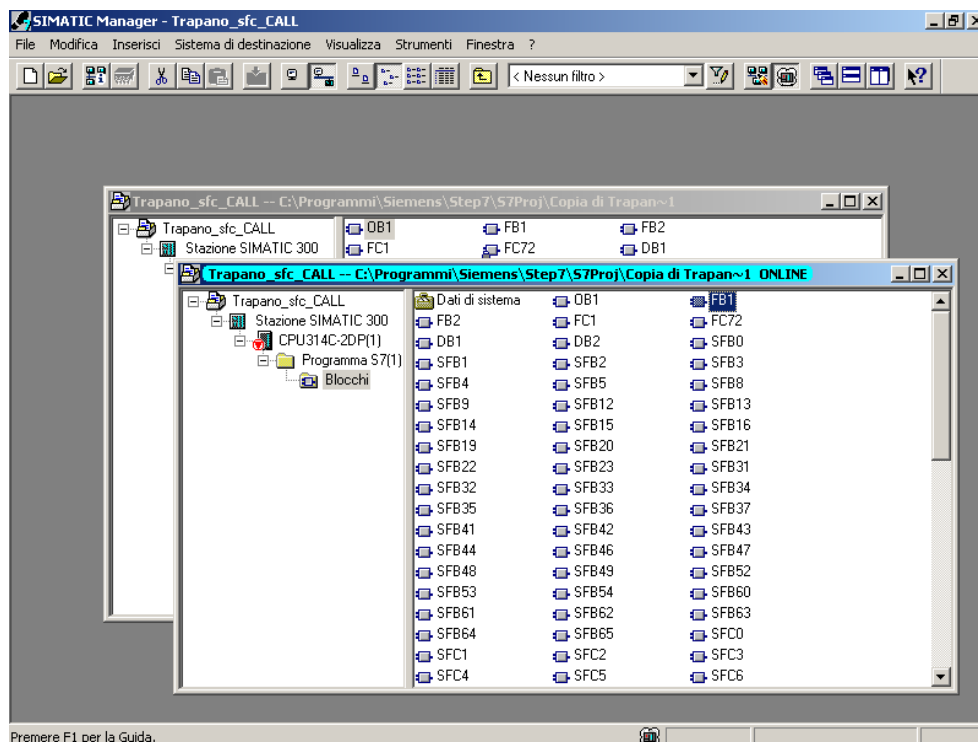


Per eseguire la simulazione del progetto:

- seguire gli stessi passi del paragrafo **1.8 Simulazione con S7-PLCSIM**

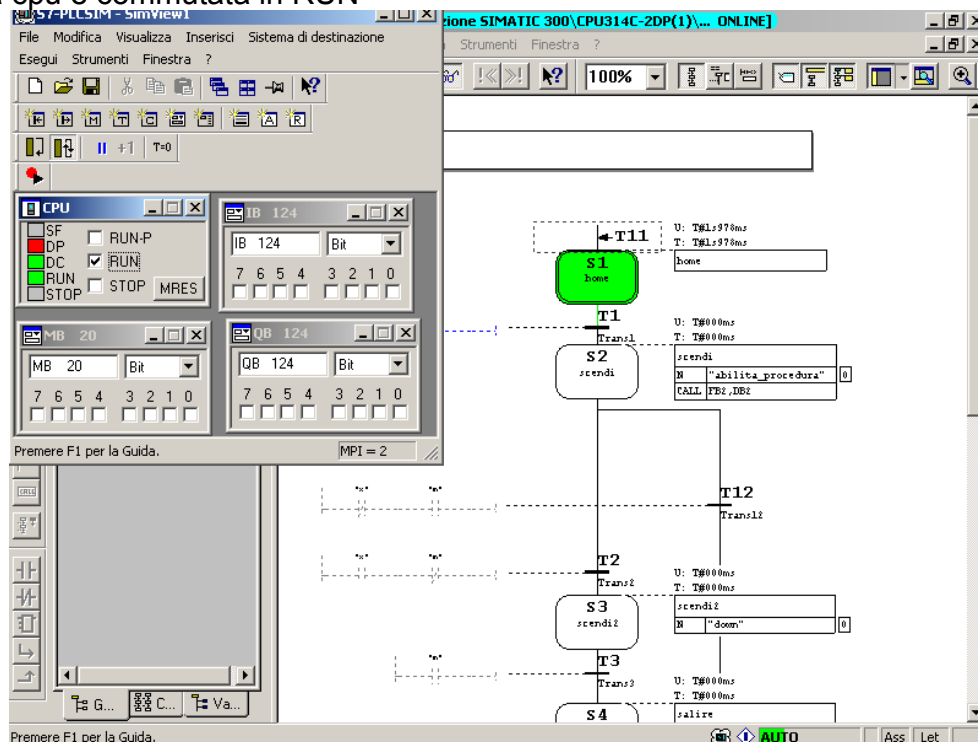


- cliccare su  e avviare FB1 (non OB1)

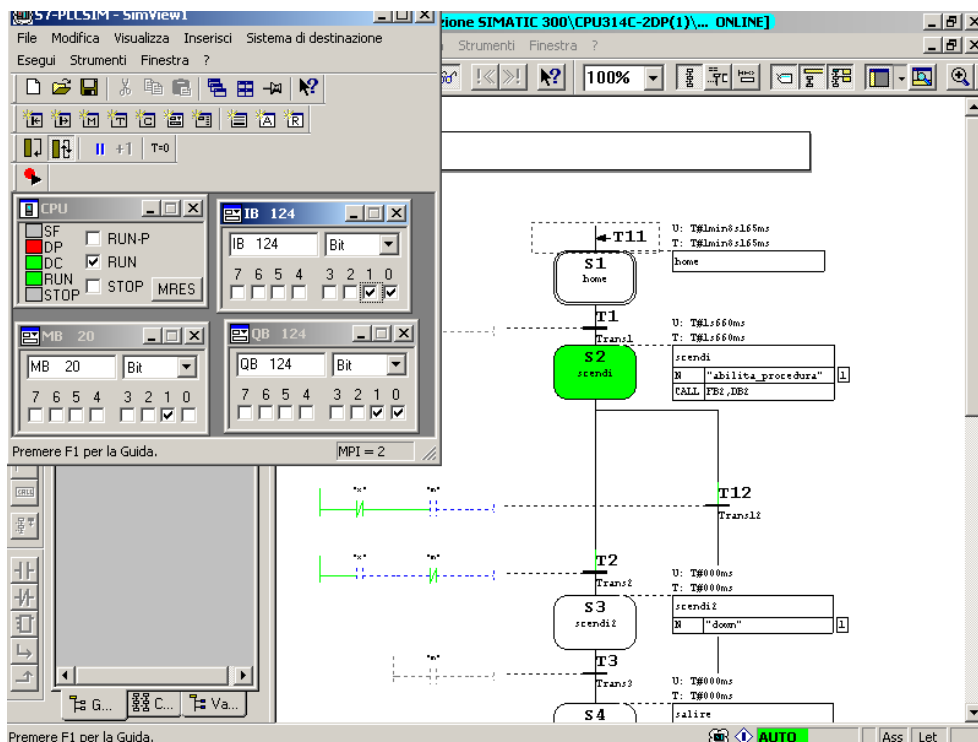


- Commutare la CPU o il simulatore in RUN e attivare i vari ingressi. Di seguito si riportano alcune immagini che sintetizzano alcuni passi della simulazione.

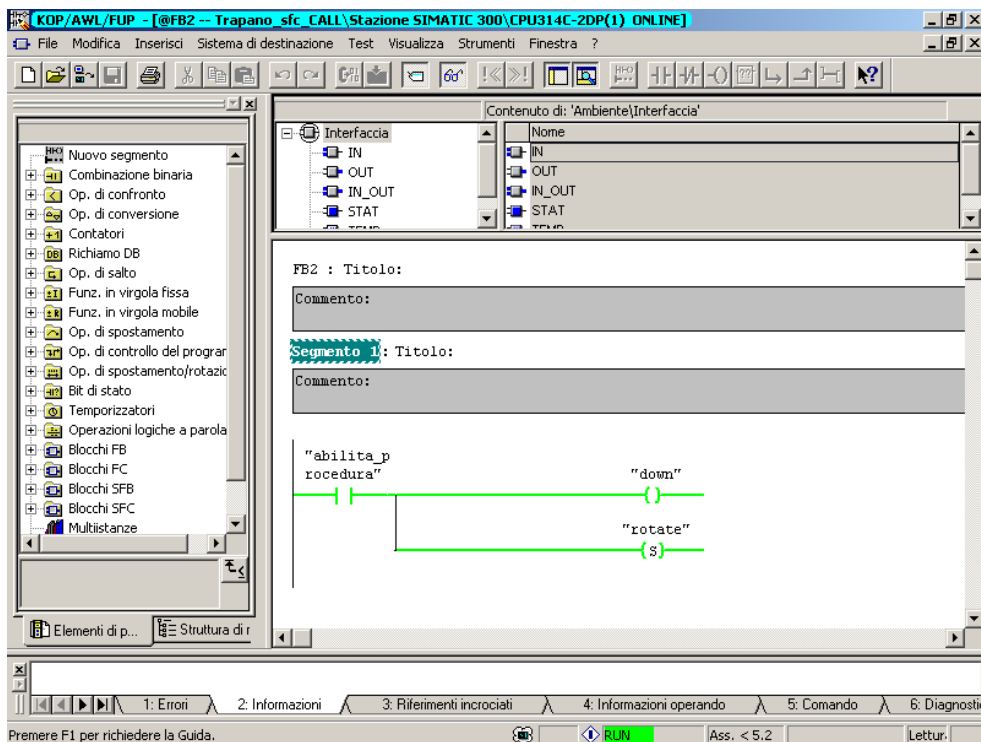
- La cpu è commutata in RUN



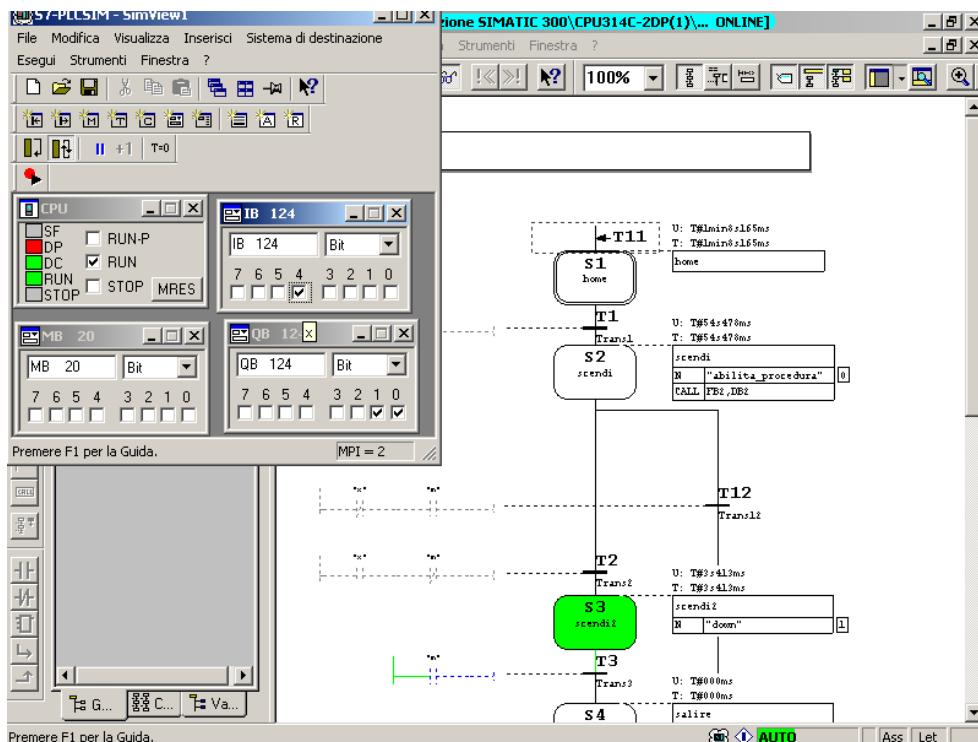
- Si attivano start e y



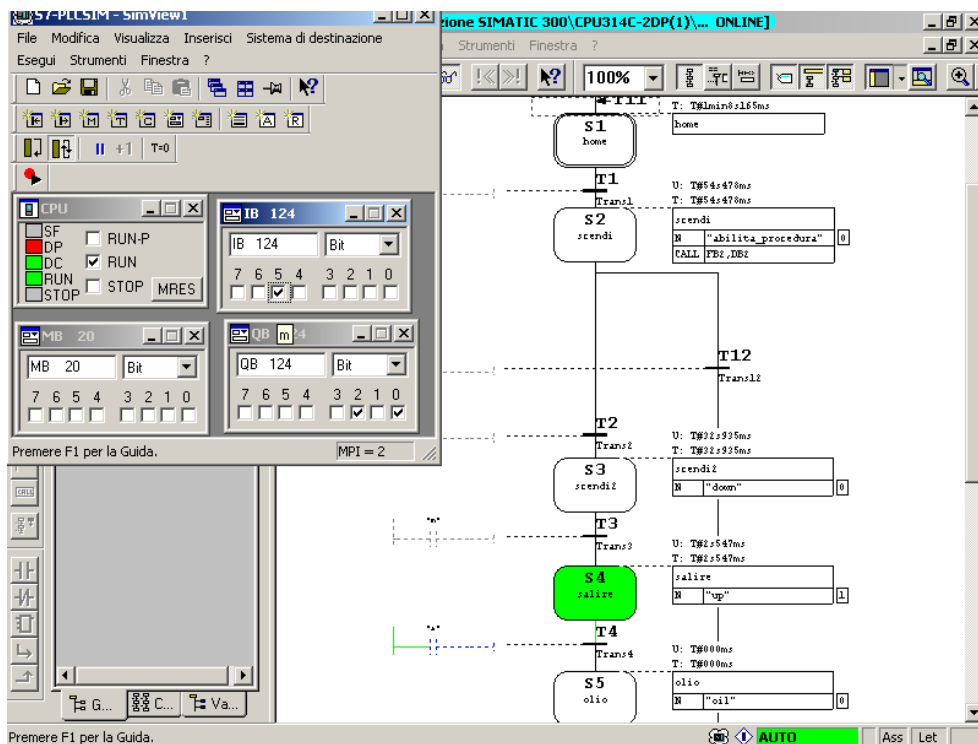
- Parte il richiamo di blocco funzionale



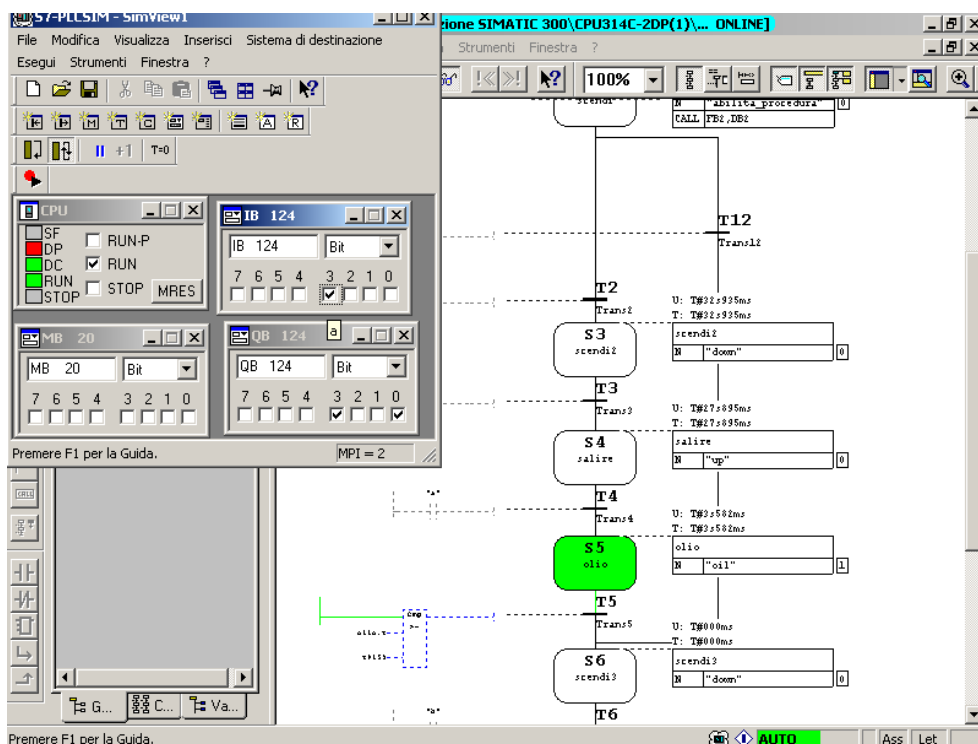
- Si attiva l'ingresso x



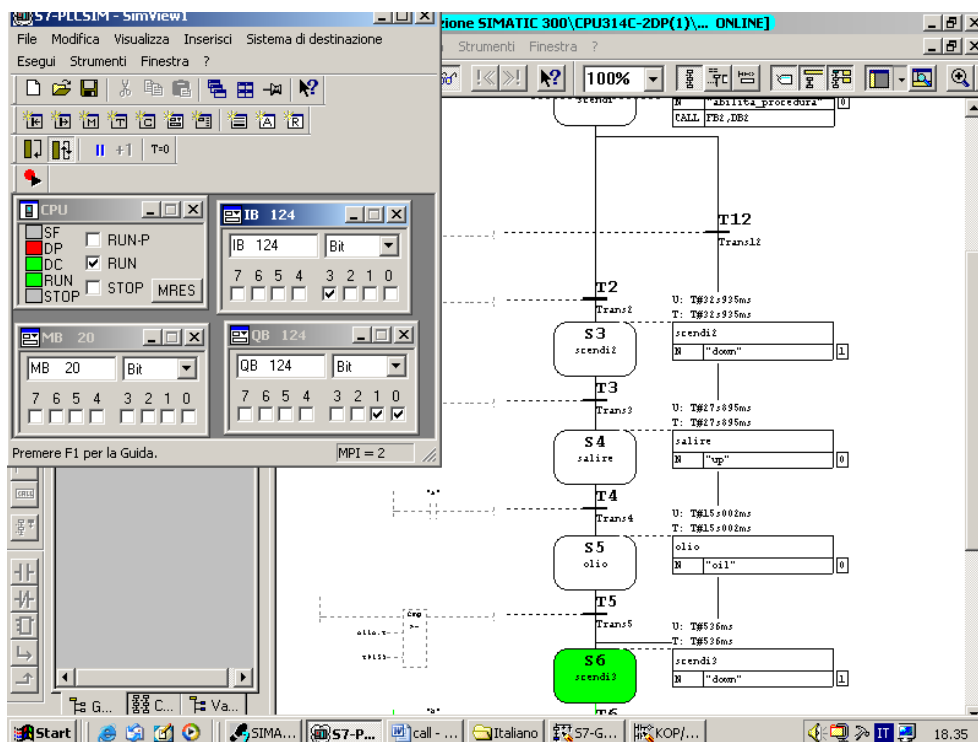
- Si attiva l'ingresso m



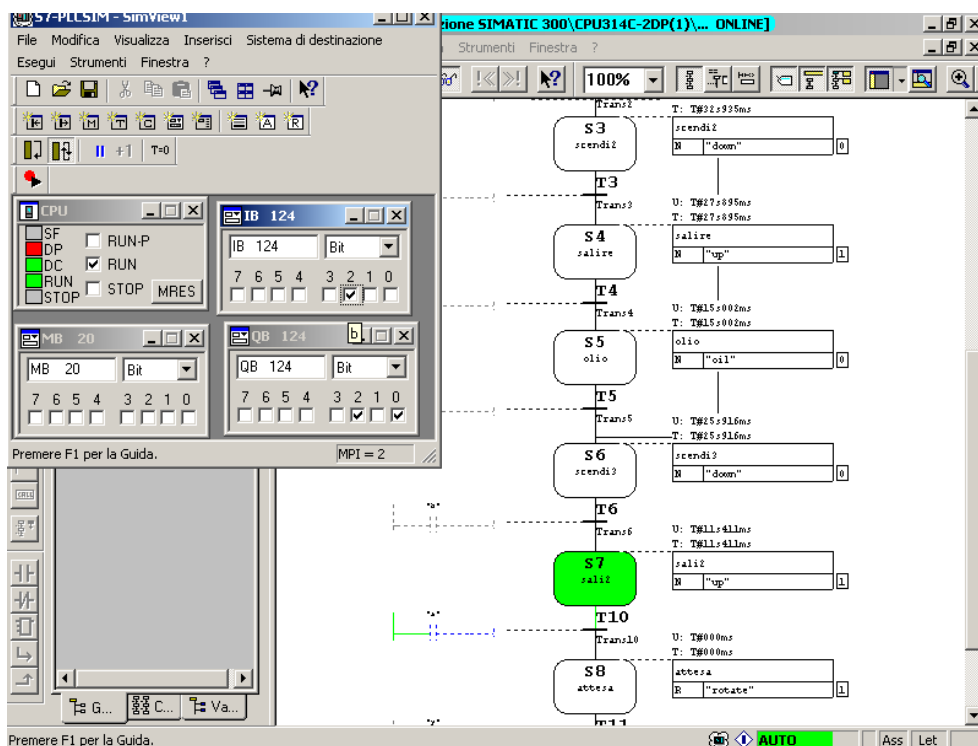
- Si attiva l'ingresso a



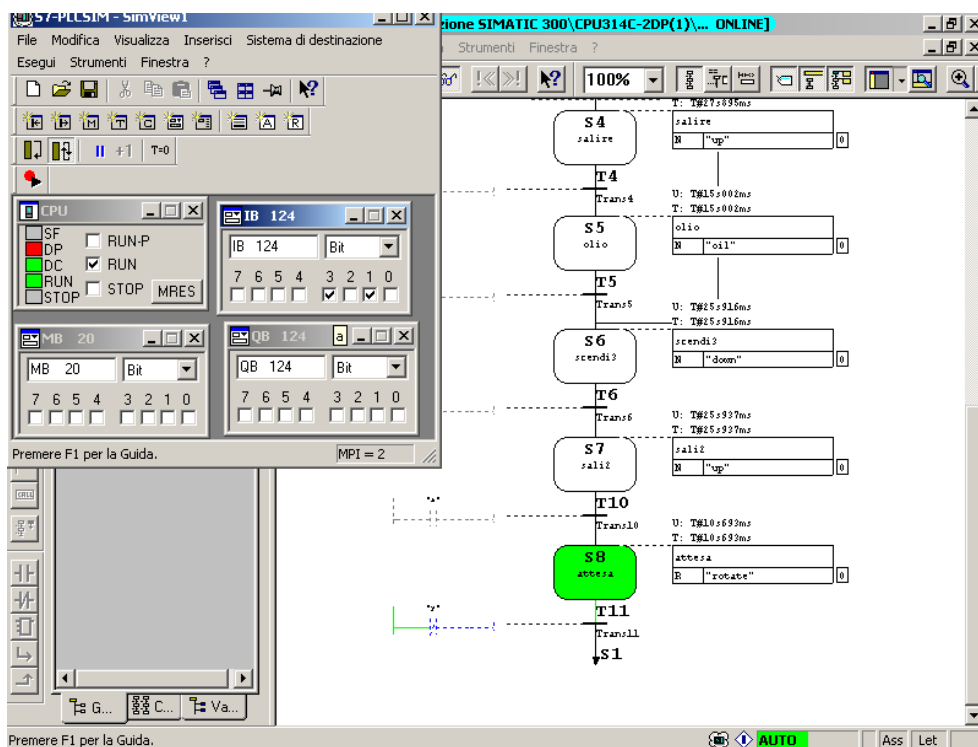
- Dopo la fase di olio, che dura 15s, si passa allo stato scendi3



- Si attiva l'ingresso b

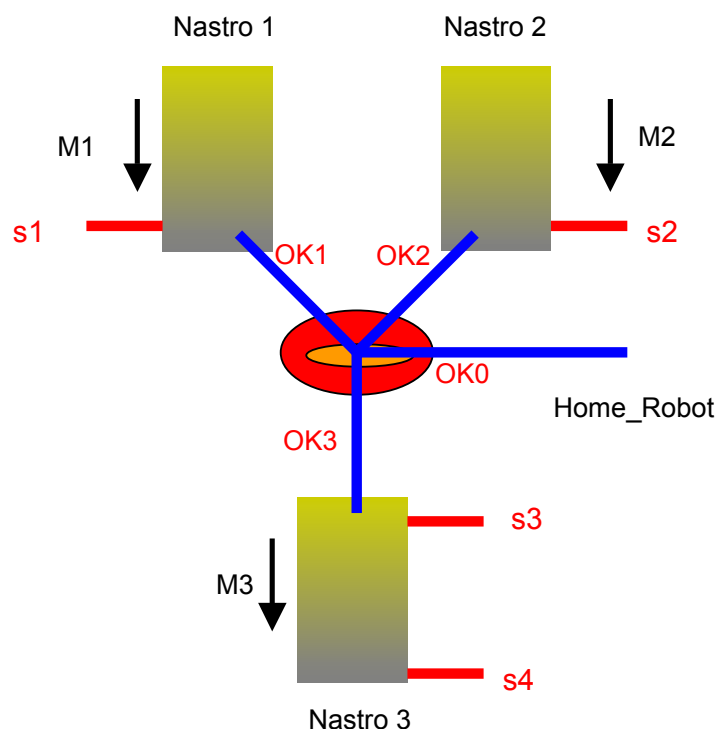


- Si attiva l'ingresso a



2.2 Svolgimento del progetto - Nastri Trasportatori e Robot

Il sistema è costituito da tre nastri trasportatori aventi ognuno un motore che li fa scorrere in avanti e da un robot con braccio meccanico mobile che si muove in senso orario e antiorario. Il robot è munito di manipolatore con motore che gli permette di aprirsi e chiudersi. Il movimento del robot è gestito da un sistema di coordinate ovvero il robot ha memorizzate come posizioni di sosta OK1, OK2 e OK3 in corrispondenza dei tre nastri, e OK0 in corrispondenza della posizione di home. Il robot dopo aver preso il pezzo dai nastri uno o due e averlo depositato nel nastro tre si riporta sempre alla posizione OK0. Il problema consiste nel trasporto di pezzi posti sui nastri uno e due al nastro tre, tramite il braccio meccanico. I nastri uno e due si considerano sempre in moto. Alla fine dei nastri uno e due sono presenti dei sensori che indicano se il pezzo è arrivato alla fine del nastro in tal caso il nastro si ferma fino a quando il braccio non preleva il pezzo. Il nastro tre è dotato invece di due sensori, uno all'inizio del nastro che permette di farlo partire quando il braccio vi rilascia il pezzo e uno alla fine che ferma il nastro quando il pezzo è giunto alla fine. Al nastro uno è data maggiore priorità qualora i pezzi giungano contemporaneamente alla fine dei nastri uno e due.



Analisi del problema

Individuiamo le variabili del sistema:

Uscite del PLC

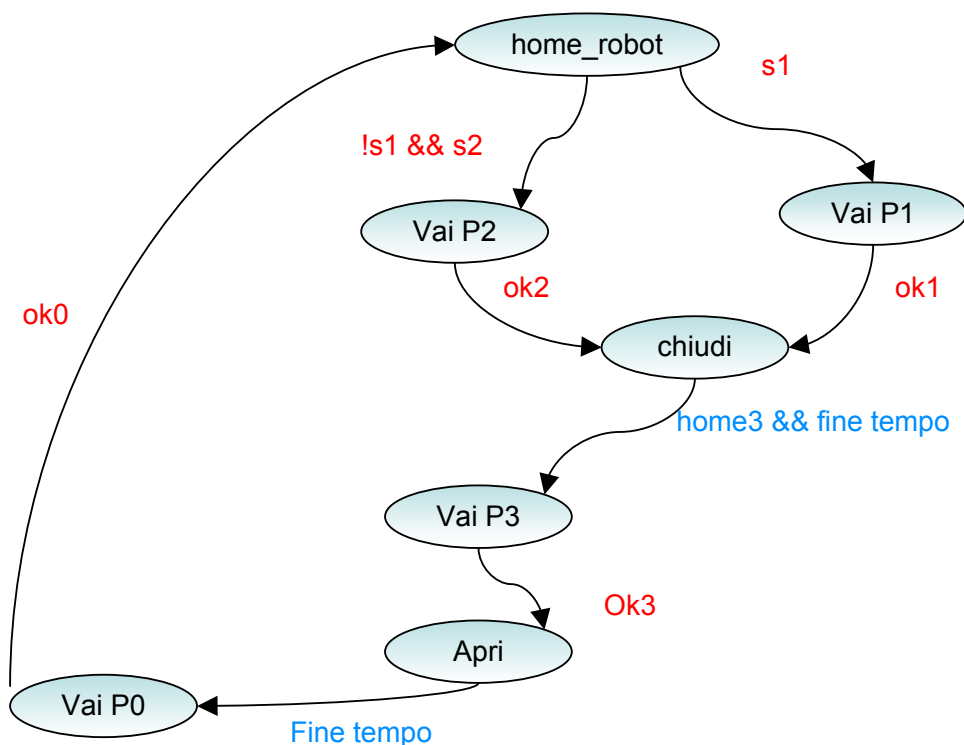
m1, m2, m3: Uscite digitali che permettono di muovere i rispettivi nastri

go0: comando digitale dato al robot per andare alla posizione home

go1: comando digitale dato al robot per andare alla posizione OK1 sul nastro1
 go2: comando digitale dato al robot per andare alla posizione OK2 sul nastro2
 go3: comando digitale dato al robot per andare alla posizione OK3 sul nastro3
 close: comando digitale per gestire l'apertura e la chiusura del manipolatore

Ingressi del PLC

s1, s2, s3 : sensori ON-OFF che indicano la presenza di un oggetto sui nastri
 s4: sensore di fine corsa sul nastro3
 ok0: sensore che indica che il robot è arrivato alla posizione P0
 ok1: sensore che indica che il robot è arrivato alla posizione P1
 ok2: sensore che indica che il robot è arrivato alla posizione P2
 ok3: sensore che indica che il robot è arrivato alla posizione P3



Descrizione macchina a stati robot

1) Allo stato di home_robot non è associata alcuna azione.

Da questo stato è possibile seguire due distinte diramazioni:

Se !s1&&s2 si passa allo stato vai_P2

2) vai_P2: go2 = 1

Se il pezzo è arrivato alla fine del nastro2 il braccio meccanico si sposta verso il nastro2. Da notare che la scelta dello stato da attivare viene effettuata tenendo conto che il nastro1 ha priorità sul nastro2.

Il passaggio allo stato successivo avviene tramite l'ingresso digitale OK2 sensore che indica che la posizione desiderata è stata raggiunta

3) CHIUDI : close = 1

In questo stato viene chiuso il manipolatore che afferra il pezzo.

fine_tempo && home3.

Prima di effettuare la transizione allo stato successivo si attende un tempo per garantire che il manipolatore abbia effettivamente afferrato il pezzo; questo è gestito da un timer. Inoltre viene verificata anche home3, variabile di sincronizzazione col nastro3. Quando il robot ha afferrato l'oggetto non può portarlo sul nastro3 se prima questo non è libero.

Se s1 si passa allo stato vai_P1

2) vai_P1: go1 = 1

Se il pezzo è arrivato alla fine del nastro1 il braccio meccanico si sposta verso il nastro1.

Il passaggio allo stato successivo avviene tramite l'ingresso digitale OK1 sensore che indica che la posizione desiderata è stata raggiunta

3) CHIUDI : close = 1

In questo stato viene chiuso il manipolatore che afferra il pezzo.

fine_tempo && home3.

4) vai_P3: go3 = 1

Il braccio trasportando il pezzo si sposta nella posizione relativa al nastro3

Il passaggio allo stato successivo avviene tramite l'ingresso digitale OK3 sensore che indica che la posizione desiderata è stata raggiunta.

5) APRI: close = 0

Viene lasciato cadere il pezzo sul nastro3

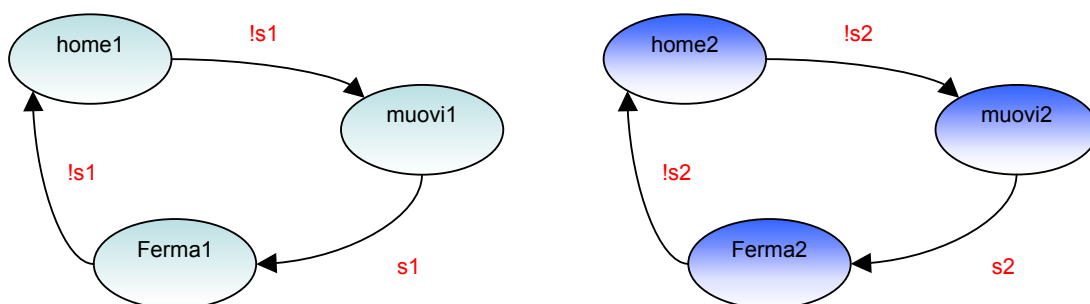
Il passaggio allo stato successivo avviene allo scadere del timer fine_tempo

6) vai_P0: go0 = 1

Il braccio si muove verso la posizione di home

Il ritorno allo stato home_robot avviene tramite l'ingresso digitale OK0 sensore che indica che la posizione desiderata è stata raggiunta.

Descrizione macchina a stati nastro1 e nastro2



1) **home 1/2**: Ci troviamo in questo stato quando il pezzo è arrivato alla fine del nastro.

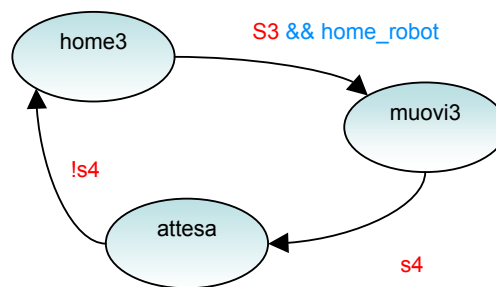
Il passaggio allo stato successivo avviene tramite !s1!/s2. Questo garantisce che il nastro sia sempre in movimento tranne quando il pezzo ha raggiunto la fine.

2) **muovi 1/2**: m1 = 1 / m2 = 1 Mette in moto il nastro.

Il passaggio allo stato successivo avviene tramite s1/s2.

3) **ferma 1/2** Quando il pezzo arriva a fine nastro il robot deve prendere il pezzo per questo il nastro si ferma. Il passaggio allo stato successivo avviene tramite !s1 / !s2.
Osservazione: La macchina a stati del nastro1/nastro2 ha apparentemente uno stato in più, ma in realtà tale stato è fondamentale per l'esecuzione del programma in Ladder. Il sistema senza questo stato di transizione comincerebbe ad oscillare.

Descrizione macchina a stati nastro3



1) **home 3**

Il passaggio allo stato successivo avviene tramite **s3 && home_robot**.
Per motivi di sicurezza si è supposto che il nastro3 dipenda dal robot, esso si muoverà solo quando il robot è tornato ad home.

2) **muovi3**: m3=1

Mette in moto il nastro

Il passaggio allo stato successivo avviene tramite **s4**. Indica che l'oggetto ha raggiunto la fine del nastro

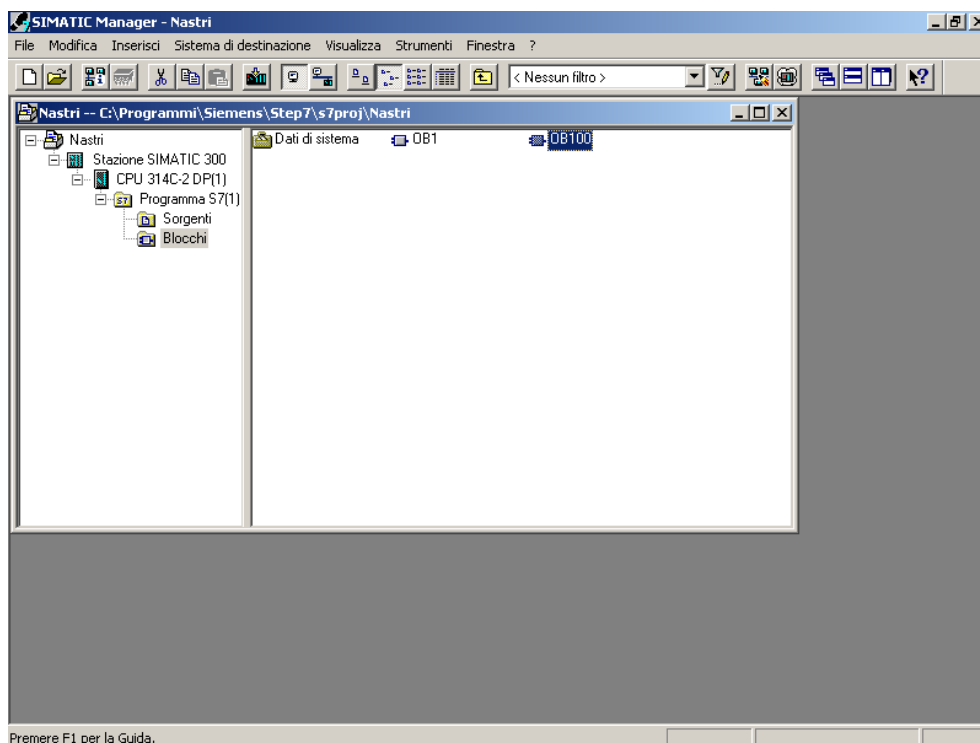
3) **attesa**

Il passaggio allo stato successivo avviene tramite **!s4**. Indica che l'oggetto è stato rimosso.

2.2.1 Progetto dei Nastri Trasportatori e del Robot realizzato in linguaggio KOP (Ladder)

Aprire il Simatic Manager e avviare la procedura **File > Assistente "Nuovo progetto"**. Scegliere la CPU314C-2DP (si ricordi che nella configurazione hardware bisogna impostare la versione firmware 2.0), selezionare OB1 e OB100, scegliere come linguaggio di programmazione KOP e indicare il nome del progetto "Nastri".

Si otterrà:



Per effettuare la programmazione con nomi simbolici bisogna creare una tabella dei simboli.

Aprire la cartella "Simboli" che si trova dentro **Programma S7(1)** e inserire le varie righe, tenendo conto di quanto detto nel paragrafo 1.3 *Indirizzi e tipi di dati ammessi nella tabella dei simboli* (vedere anche pag. 33).

Quello che si ottiene è:

Editor di simboli - [Programma S7(1) (Simboli) -- Nastri\Stazione SIMATIC 300\CPU 314C-2 DP(1)]					
Tabella Modifica Inserisci Visualizza Strumenti Finestra ?					
Tutti i simboli					
	Stato	Simbolo	Indirizz	Tipo di da	Commento
1		s1	I 124.0	BOOL	seniore primo nastro
2		s2	I 124.1	BOOL	seniore secondo nastro
3		ok1	I 124.2	BOOL	posizione robot 1
4		ok2	I 124.3	BOOL	posizione robot 2
5		ok3	I 124.4	BOOL	posizione robot 3
6		ok0	I 124.5	BOOL	posizione robot 0
7		s3	I 124.6	BOOL	seniore iniziale terzo nastro
8		s4	I 124.7	BOOL	seniore finale terzo nastro
9		home_robot	M 20.0	BOOL	stato di home del robot
10		vai_p1	M 20.1	BOOL	il robot va a prendere il pezzo sul primo nastro
11		vai_p2	M 20.2	BOOL	il robot va a prendere il pezzo sul secondo nastro
12		chiudi	M 20.3	BOOL	il robot afferra il pezzo
13		vai_p3	M 20.4	BOOL	il robot porta il pezzo sul nastro tre
14		apri	M 20.5	BOOL	il robot rilascia il pezzo
15		vai_p0	M 20.6	BOOL	il robot ritorna nella sua posizione di home
16		home1	M 20.7	BOOL	posizione di home del nastro 1
17		muovi1	M 21.0	BOOL	nastro 1 in movimento
18		ferma1	M 21.1	BOOL	nastro 1 fermo
19		home2	M 21.2	BOOL	posizione di home del nastro 2
20		muovi2	M 21.3	BOOL	nastro 2 in movimento
21		ferma2	M 21.4	BOOL	nastro 2 fermo
22		muovi3	M 21.5	BOOL	nastro 3 in movimento
23		attesa	M 21.6	BOOL	in questo stato si attende che il pezzo venga prelevato dal nastro 3
24		home3	M 21.7	BOOL	posizione di home del nastro 3
25		Cycle Execution	OB 1	OB 1	
26		COMPLETE REST...	OB 100	OB 100	Complete Restart
27		go0	Q 124.0	BOOL	comando che aziona il robot per andare in posizione di home
28		go1	Q 124.1	BOOL	comando che aziona il robot per andare in posizione 1
29		go2	Q 124.2	BOOL	comando che aziona il robot per andare in posizione 2
30		go3	Q 124.3	BOOL	comando che aziona il robot per andare in posizione 3
31		m1	Q 124.4	BOOL	comando che aziona il motore del primo nastro
32		m2	Q 124.5	BOOL	comando che aziona il motore del secondo nastro
33		m3	Q 124.6	BOOL	comando che aziona il motore del terzo nastro
34		close	Q 124.7	BOOL	comando che fa chiudere il braccio del robot
35		tempo1	T 0	TIMER	
36		tempo2	T 1	TIMER	
37					

Premere F1 per accedere alla Guida.

NUM

Per scrivere il programma in KOP(Ladder) cliccare su OB1. Si aprirà la maschera “KOP/AWL/FUP” dove si inseriranno le varie istruzioni.

Il programma si articola in segmenti. Quindi bisogna selezionare “inserisci segmento” per cominciare a introdurre i simboli del linguaggio ladder. Sulla sinistra si trova la finestra “elementi di programma”. Questa contiene tutta la simbologia relativa al linguaggio KOP(ladder).

Ogni qualvolta si inserisce un elemento esso presenterà dei punti interrogativi rossi. Cliccando su di essi è possibile inserire l'indirizzo assoluto oppure se si è costruita la tabella dei simboli è possibile scegliere il nome nella casella di riepilogo visualizzata.

Il programma in KOP sarà composto da tre fasi. A partire dalla macchina di stato bisogna scrivere questa parti di codice:

- stato attuale – ingresso – nuovo stato
- nuovo stato – resettaggio vecchio stato
- stato attuale – condizioni che devono essere compiute

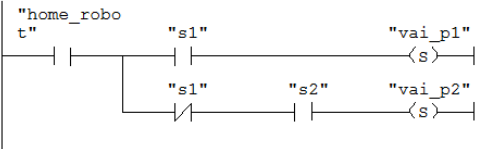
OB1 - <offline>

"Cycle Execution"
Nome: Famiglia:
Autore: Versione: 0.1
Versione blocco: 2
Data e ora Codice: 18/02/2005 17.35.17
Interfaccia: 15/02/1996 16.51.12
Lunghezza (blocco / codice / dati) 00462 00266 00022

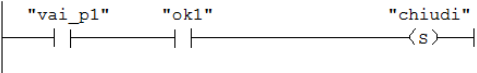
Nome	Tipo di dati	Indirizzo	Commento
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Blocco:OB1 "Main Program Sweep (Cycle)"

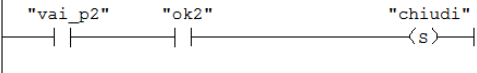
Segmento: 1 Programma di funzionamento del ROBOT



Segmento: 2 il robot afferra il pezzo



Segmento: 3 il robot afferra il pezzo

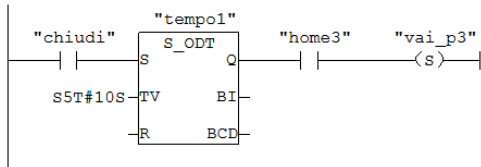


SIMATIC

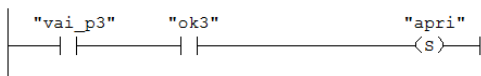
OB1

22/02/2005 12.29.24

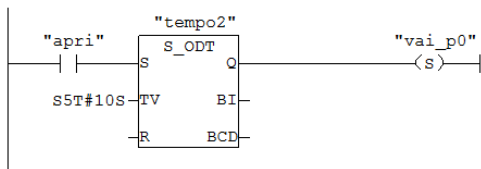
Segmento: 4



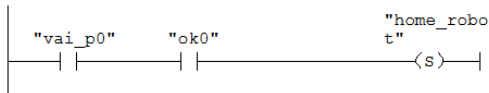
Segmento: 5 il robot rilascia il pezzo



Segmento: 6



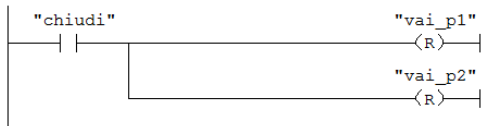
Segmento: 7 stato di home del robot



Segmento: 8 stato di home del robot



Segmento: 9 il robot va a prendere il pezzo sul primo nastro



SIMATIC

OB1

22/02/2005 12.29.24

Segmento: 10 il robot afferra il pezzo

"vai_p3"	"chiudi"
	—(R)—

Segmento: 11 il robot porta il pezzo sul nastro tre

"apri"	"vai_p3"
	—(R)—

Segmento: 12 il robot rilascia il pezzo

"vai_p0"	"apri"
	—(R)—

Segmento: 13 il robot ritorna nella sua posizione di home

"home_robot"	"vai_p0"
	—(R)—

Segmento: 14 comando che aziona il robot per andare in posizione 1

"vai_p0"	"go1"
	—(<)—

Segmento: 15 comando che aziona il robot per andare in posizione 2

"vai_p2"	"go2"
	—(<)—

Segmento: 16 comando che fa chiudere il braccio del robot

"chiudi"	"close"
	—(S)—

SIMATIC

OB1

22/02/2005 12.29.24

Segmento: 17 comando che aziona il robot per andare in posizione 3



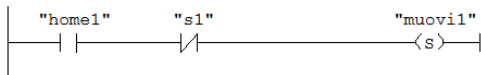
Segmento: 18 comando che fa chiudere il braccio del robot



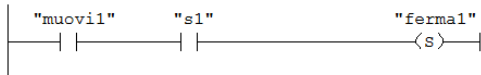
Segmento: 19 comando che aziona il robot per andare in posizione di home



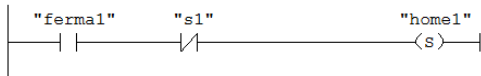
Segmento: 20 programma di funzionamento del NASTRO 1



Segmento: 21 nastro 1 fermo



Segmento: 22 posizione di home del nastro 1



Segmento: 23 posizione di home del nastro 1



SIMATIC

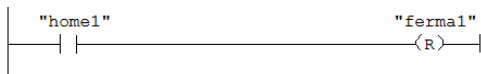
OB1

22/02/2005 12.29.24

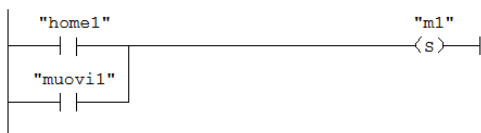
Segmento: 24 nastro 1 in movimento



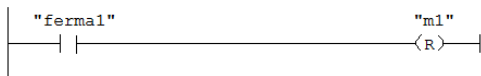
Segmento: 25 nastro 1 fermo



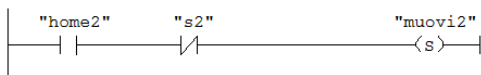
Segmento: 26 comando che aziona il motore del primo nastro



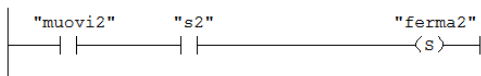
Segmento: 27 comando che aziona il motore del primo nastro



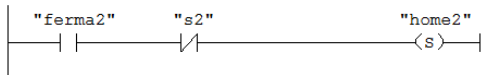
Segmento: 28 programma di funzionamento del NASTRO 2



Segmento: 29 nastro 2 fermo



Segmento: 30 posizione di home del nastro 2



SIMATIC

OB1

22/02/2005 12.29.24

Segmento: 31 posizione di home del nastro 2



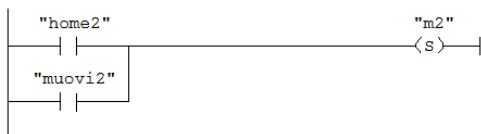
Segmento: 32 nastro 2 in movimento



Segmento: 33 nastro 2 fermo



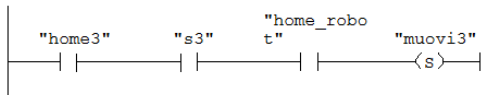
Segmento: 34 comando che aziona il motore del secondo nastro



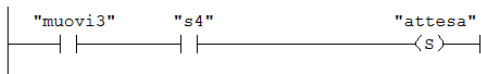
Segmento: 35 comando che aziona il motore del secondo nastro



Segmento: 36 programma di funzionamento del NASTRO 3



Segmento: 37 in questo stato si attende che il pezzo venga prelevato dal nast

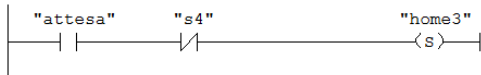


SIMATIC

OB1

22/02/2005 12.29.24

Segmento: 38 posizione di home del nastro 3



Segmento: 39 posizione di home del nastro 3



Segmento: 40 nastro 3 in movimento



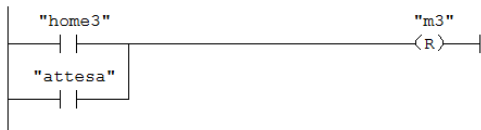
Segmento: 41 in questo stato si attende che il pezzo venga prelevato dal nast



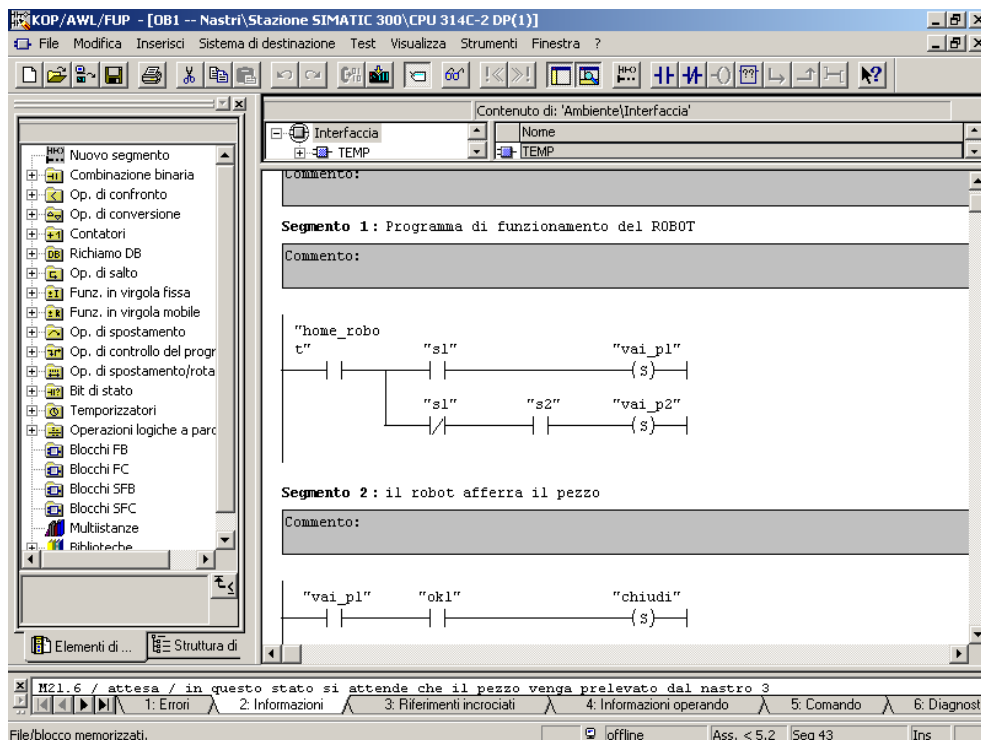
Segmento: 42 comando che aziona il motore del terzo nastro



Segmento: 43 comando che aziona il motore del terzo nastro



Di seguito viene riportato un esempio dell'interfaccia "KOP/AWL/FUP"



Dopo aver salvato il programma, aprire il blocco di avviamento OB100 per inserire le istruzioni che permettono di settare le variabili: home_Robot, home1, home2, home3. Scegliere dal menu **Visualizza > AWL** e inserire la seguente istruzione, che al momento dell'avviamento setta le variabili a true:

SET

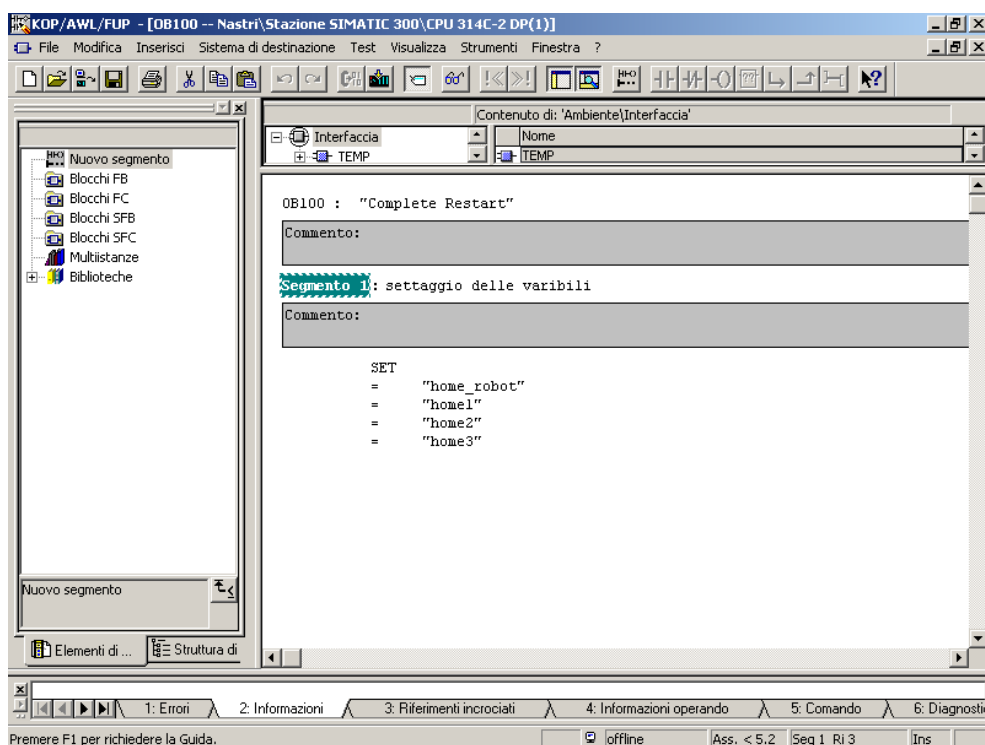
= "home_robot"

= "home1"

= "home2"

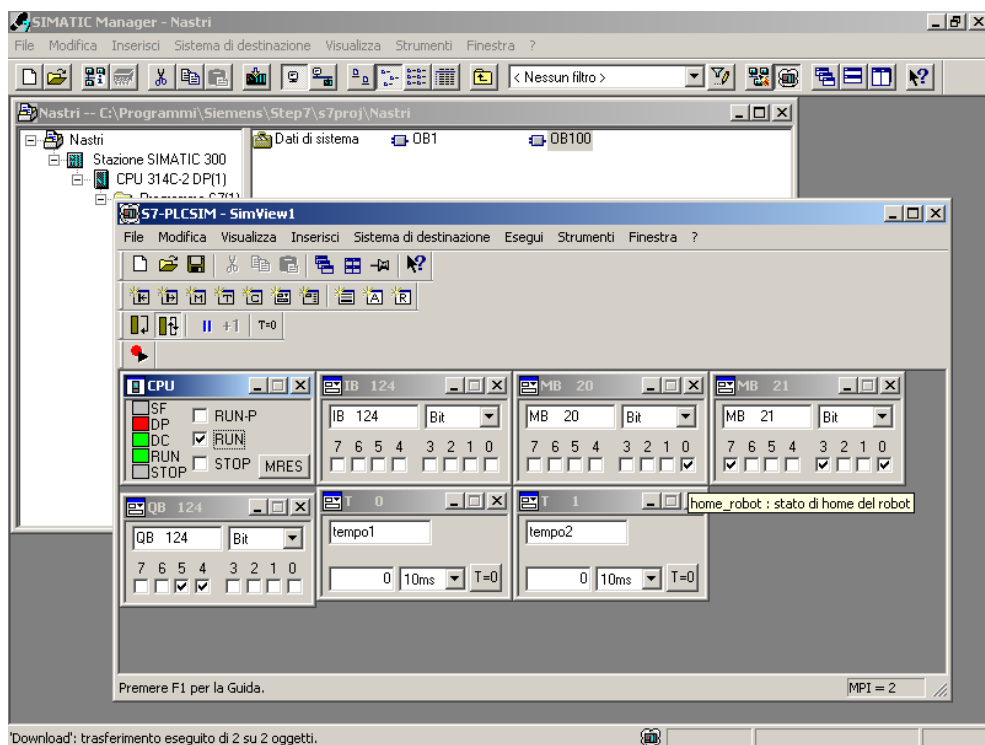
= "home3"

Si ricorda che AWL (lista istruzioni) è un linguaggio di programmazione testuale vicino al linguaggio macchina. Quando si crea un programma in AWL, le singole istruzioni corrispondono in larga misura alle operazioni con le quali la CPU elabora il programma.



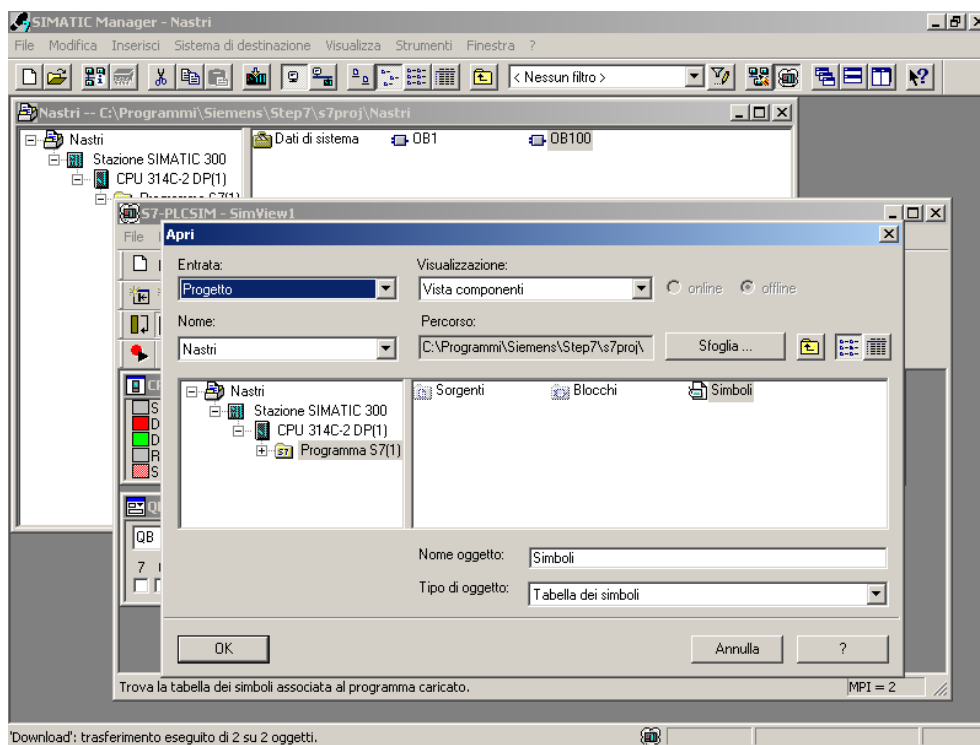
Per simulare il progetto seguire i passi dei paragrafi:

- 1.8 Simulazione con S7-PLCSIM
- 1.9 Come utilizzare STEP 7 per controllare la simulazione del programma



Come si vede dall'immagine è possibile inserire ingressi, uscite, merker e temporizzatori. Inoltre è possibile associare ad ogni elemento (IB,MB,QB, T0) il corrispondente simbolo della tabella dei simboli.

Basta andare su **Strumenti > opzioni > aggiungi simboli**. Si aprirà la maschera "apri" che permette di selezionare la tabella dei simboli relativa al progetto in esame.



2.2.2 Progetto dei Nastri Trasportatori e del Robot realizzato in linguaggio SFC

Nella prima fase bisogna aprire il Simatic Manager e richiamare il comando di menu **File > Assistente "Nuovo progetto"**. Creare un nuovo progetto "Nastri_sfc".

Per usare la notazione simbolica è necessario creare la tabella dei simboli. Si ricorda che in figura è riportata la tabella dei simboli ottenuta dopo il salvataggio e la compilazione del file S7GRAPH. Inizialmente è sufficiente inserire solo ingressi, uscite e i merker.

	Stato	Simbolo	Indirizzo	Tipo di dati	Commento
1		G7_STD_3	FC 72	FC 72	
2		s1	I 124.0	BOOL	sensore primo nastro
3		s2	I 124.1	BOOL	sensore secondo nastro
4		ok1	I 124.2	BOOL	posizione robot 1
5		ok2	I 124.3	BOOL	posizione robot 2
6		ok3	I 124.4	BOOL	posizione robot 3
7		ok0	I 124.5	BOOL	posizione robot 0
8		s3	I 124.6	BOOL	sensore iniziale terzo nastro
9		s4	I 124.7	BOOL	sensore finale terzo nastro
10		robot_home	M 20.0	BOOL	
11		nastro3_home	M 20.1	BOOL	
12		INIT_SQ	M 20.2	BOOL	
13		Cycle Execution	OB 1	OB 1	
14		go0	Q 124.0	BOOL	comando che aziona il robot per andare in posizione di home
15		go1	Q 124.1	BOOL	comando che aziona il robot per andare in posizione 1
16		go2	Q 124.2	BOOL	comando che aziona il robot per andare in posizione 2
17		go3	Q 124.3	BOOL	comando che aziona il robot per andare in posizione 3
18		m1	Q 124.4	BOOL	comando che aziona il motore del primo nastro
19		m2	Q 124.5	BOOL	comando che aziona il motore del secondo nastro
20		m3	Q 124.6	BOOL	comando che aziona il motore del terzo nastro
21		close	Q 124.7	BOOL	comando che fa chiudere il braccio del robot
22		TIME_TCK	SFC 64	SFC 64	Read the System Time
23					

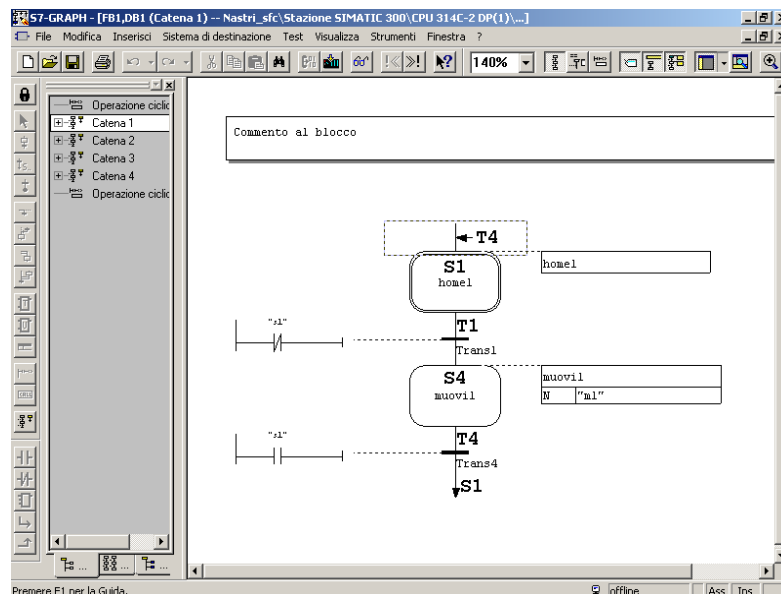
Dopo aver creato il progetto e aver definito la tabella dei simboli portarsi dentro "Blocchi" e con il tasto destro del mouse selezionare **Inserisci nuovo oggetto > Blocco funzionale**. Si aprirà la finestra "Proprietà- Blocco funzionale" e nella casella linguaggio scegliere "GRAPH".

Cliccare su FB1 e si aprirà S7_GRAPH.

Il progetto può essere composto da 4 catene sequenziali, ognuna delle quali individua rispettivamente la

- CATENA nastro1,
- CATENA nastro2,
- CATENA nastro3
- CATENA robot.

➤ Catena Nastro1



Step home1:

Allo step iniziale non è associata alcuna azione

Transizione T1:

la transizione è composta dal contatto normalmente chiuso s1

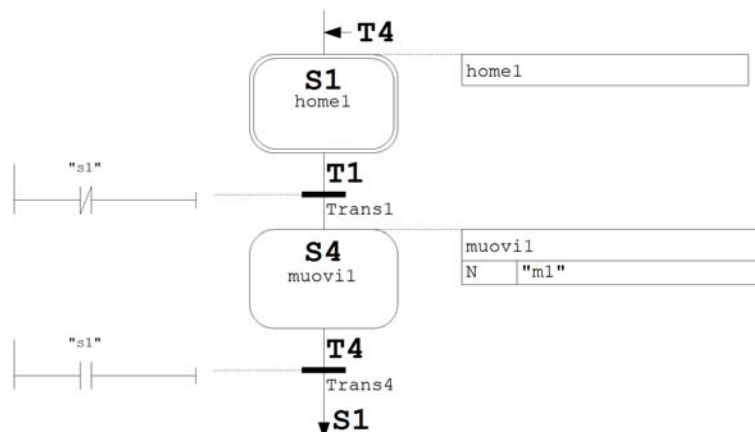
Step muovi1:

Allo step è associata la seguente azione: N "m1" questa indica che finchè il passo è attivo m1 è 1

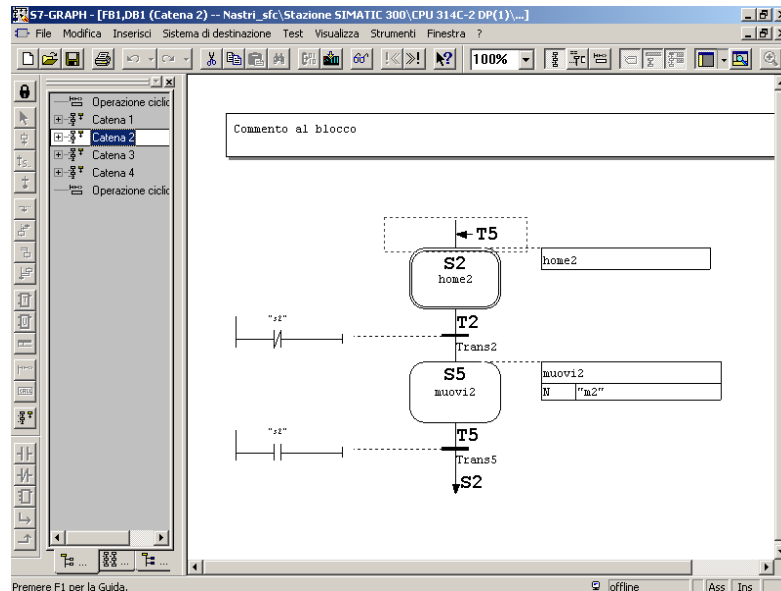
Transizione T4:

la transizione è composta da un contatto normalmente aperto s1.

Dopodichè si ritorna allo step Home1



➤ Catena Nastro2



Step home2:

allo step iniziale non è associata alcuna azione

Transizione T2:

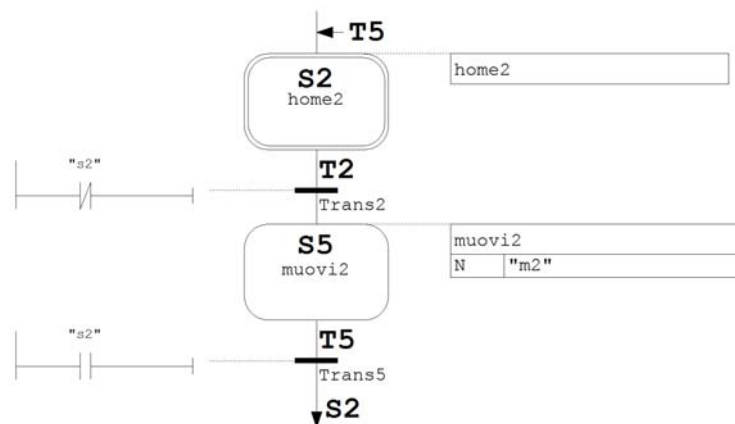
la transizione è composta dal contatto normalmente chiuso s2

Step muovi2:

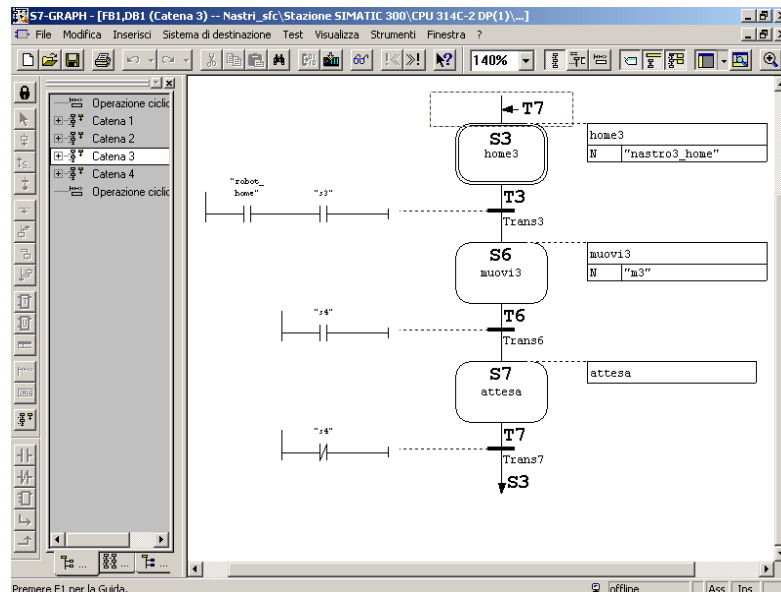
allo step è associata la seguente azione: N "m2"
questa indica che finché il passo è attivo m2 è 1

Transizione T5:

la transizione è composta da un contatto normalmente aperto s2.
Dopodiché si ritorna allo step Home2.



➤ Catena Nastro3



Step home3:

allo step iniziale è associata la seguente azione: N "nastro3_home" questa indica che finché il passo è attivo nastro3_home è 1. Questa variabile è necessaria per sapere quando il nastro 3 non ha più pezzi.

Transizione T3:

la transizione è composta dal contatto normalmente aperto Robot_home e dal contatto normalmente aperto s3.

Step muovi3:

allo step è associata la seguente azione: N "m3" questa indica che finché il passo è attivo m3 è 1

Transizione T6:

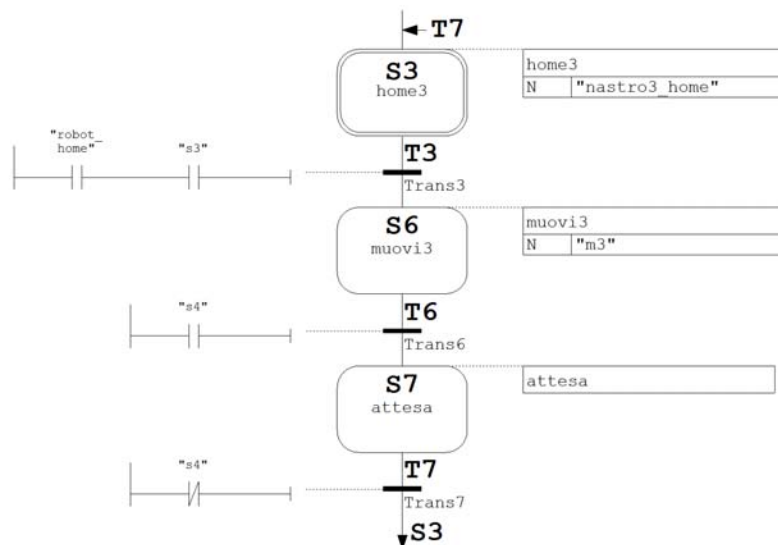
la transizione è composta da un contatto normalmente aperto s4.

Step attesa:

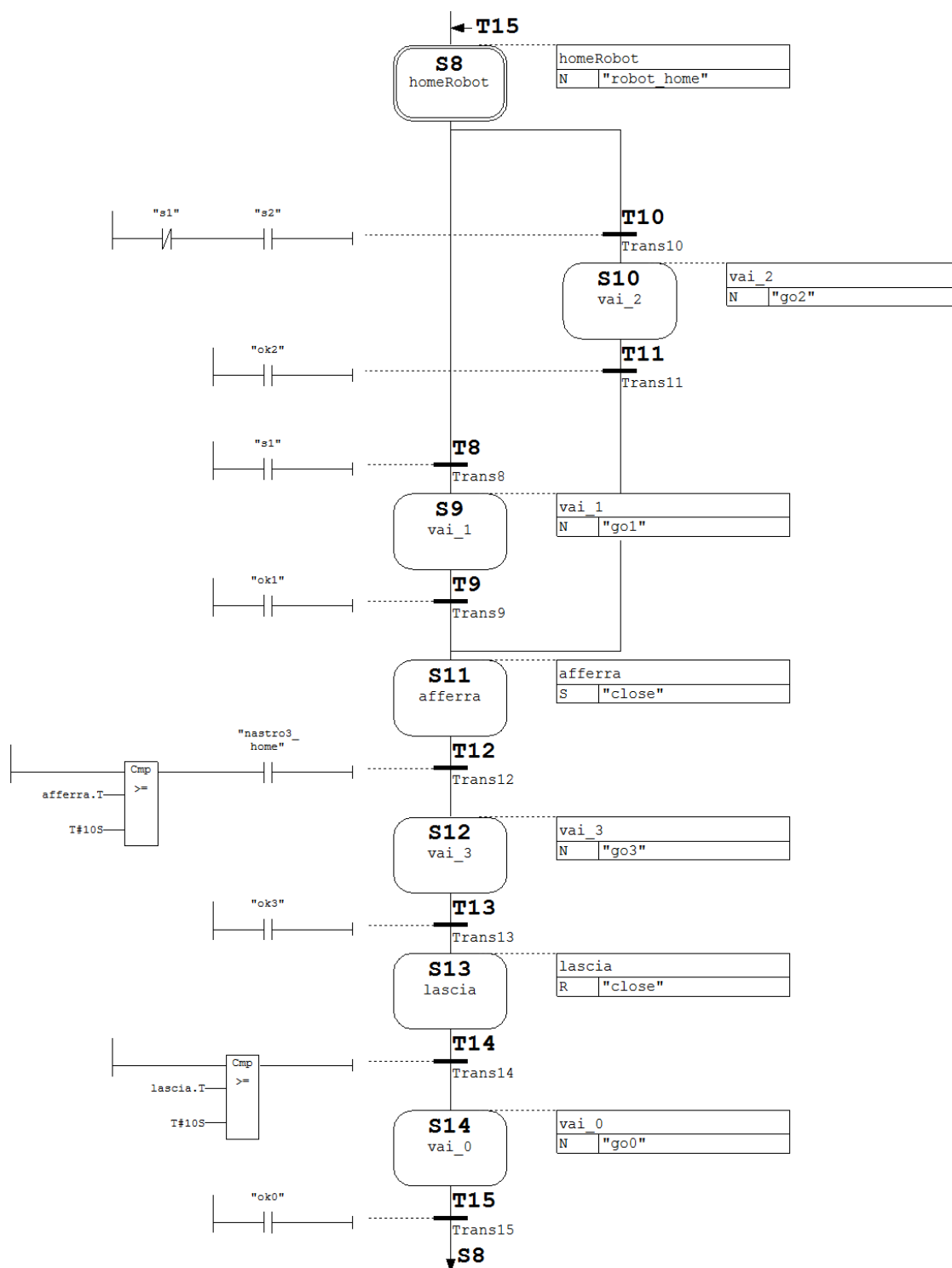
allo step non è associata alcuna azione

Transizione T7:

la transizione è composta da un contatto normalmente chiuso s4. Dopodichè si ritorna allo step Home3.



➤ Catena Robot



Step homeRobot:

allo step iniziale è associata la seguente azione: N “robot_home” questa indica che finché il passo è attivo robot_home è 1.

Transizione T8:

la transizione è composta dal contatto normalmente aperto s1.

Step vai_1:

allo step è associata la seguente azione: N “go1” questa indica che finché il passo è attivo go1 è 1

Transizione T9:

la transizione è composta da un contatto normalmente aperto ok1.

Esaminiamo adesso l'altro percorso.

Transizione T10:

la transizione è composta da un contatto normalmente chiuso s1 e da un contatto normalmente aperto s2.

Step vai_2:

allo step è associata la seguente azione: N “go2” questa indica che finché il passo è attivo go2 è 1

Transizione T11:

la transizione è composta da un contatto normalmente aperto ok2.

A questo punto si ha un “percorso convergente” in quanto le due sequenze alternative convergono.

Step afferra:

allo step è associata la seguente azione: S “close” questa indica che appena il passo è attivo close viene impostato a 1 e rimane così impostato

Transizione T12:

la transizione è composta un comparatore dove confrontiamo afferra.T, che rappresenta il tempo di attivazione corrente o più recente del passo, con un tempo prefissato di 10 secondi (T#10S). Questo equivale a fissare il tempo necessario per portare il pezzo sul nastro3.

Step vai_3:

allo step è associata la seguente azione: N “go3” questa indica che finché il passo è attivo go3 è 1

Transizione T13:

la transizione è composta da un contatto normalmente aperto ok3.

Step lascia:

allo step è associata l'azione R “close”. Appena il passo è attivo close viene impostato a 0 e rimane così impostato

Transizione T14:

la transizione è composta da un comparatore dove confrontiamo lascia.T, che rappresenta il tempo di attivazione corrente o più recente del passo, con un tempo prefissato di 10 secondi (T#10S).

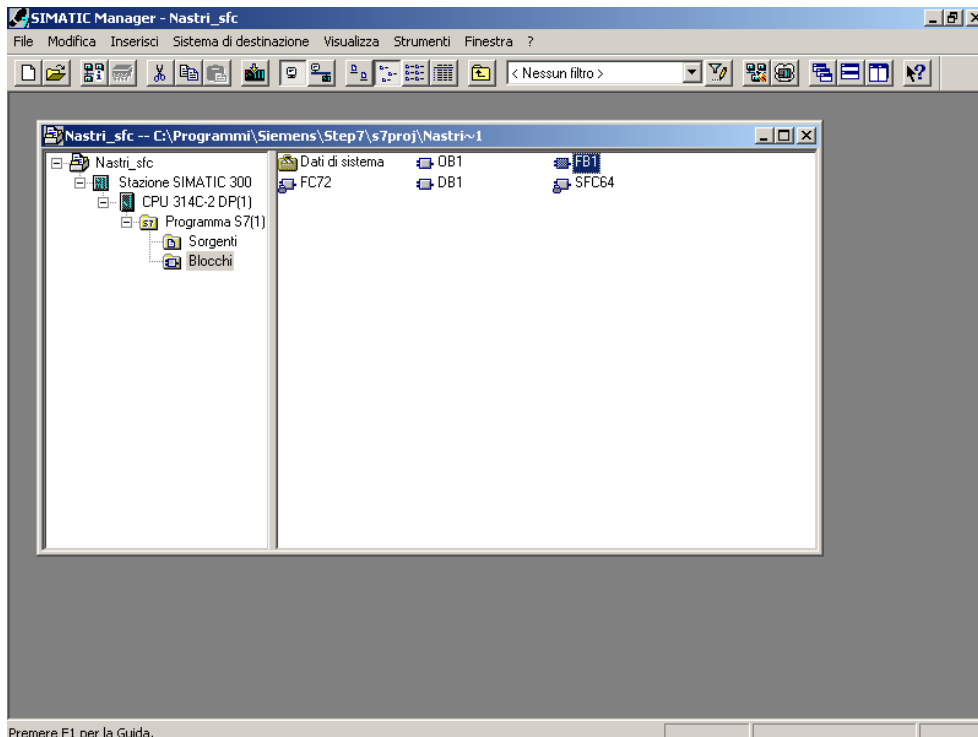
Step vai_0:

allo step è associata la seguente azione: N “go0” questa indica che finché il passo è attivo go0 è 1

Transizione T15:

la transizione è composta da un contatto normalmente aperto ok0.
Dopodichè si ritorna allo step HomeRobot

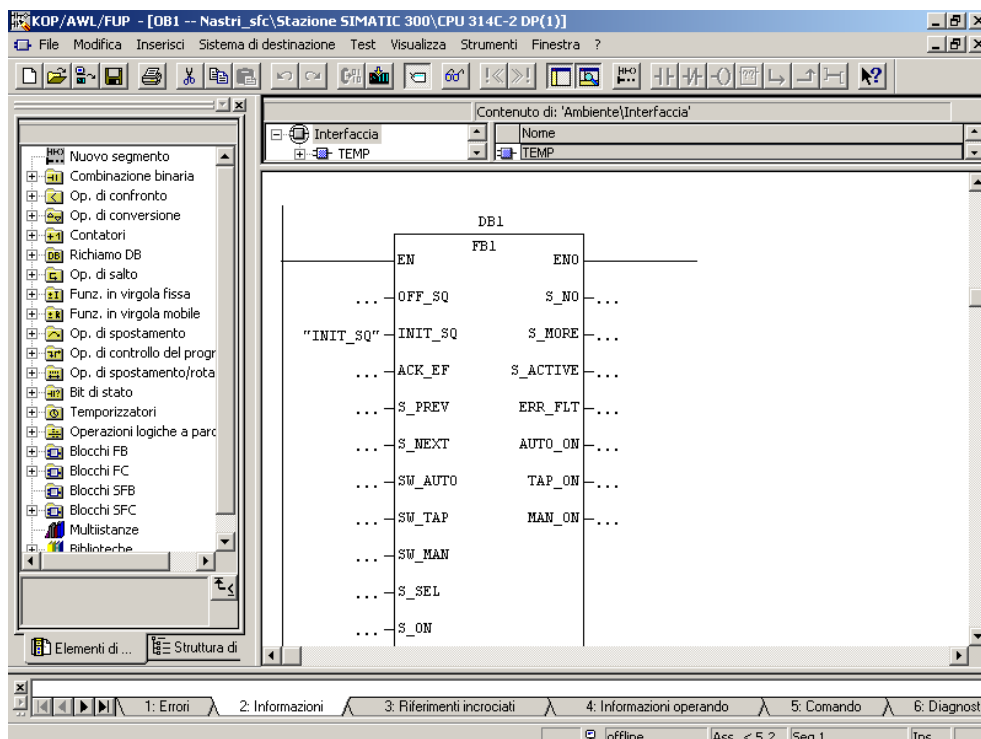
Dopo aver salvato e compilato il progetto realizzato in S7_GRAPH, si troveranno nella cartella Blocchi i seguenti elementi generati dalla compilazione




Per realizzare il richiamo di FB1 all'interno dell'OB1. E' necessario aprire OB1 e dalla finestra "elementi di programma", posizionarsi su blocchi e scegliere FB1.

Introdurre al posto dei punti interrogativi, DB1. Si ricorda che i blocchi dati di istanza(DB), generati automaticamente nella compilazione, vengono assegnati al blocco quando viene richiamato un FB.

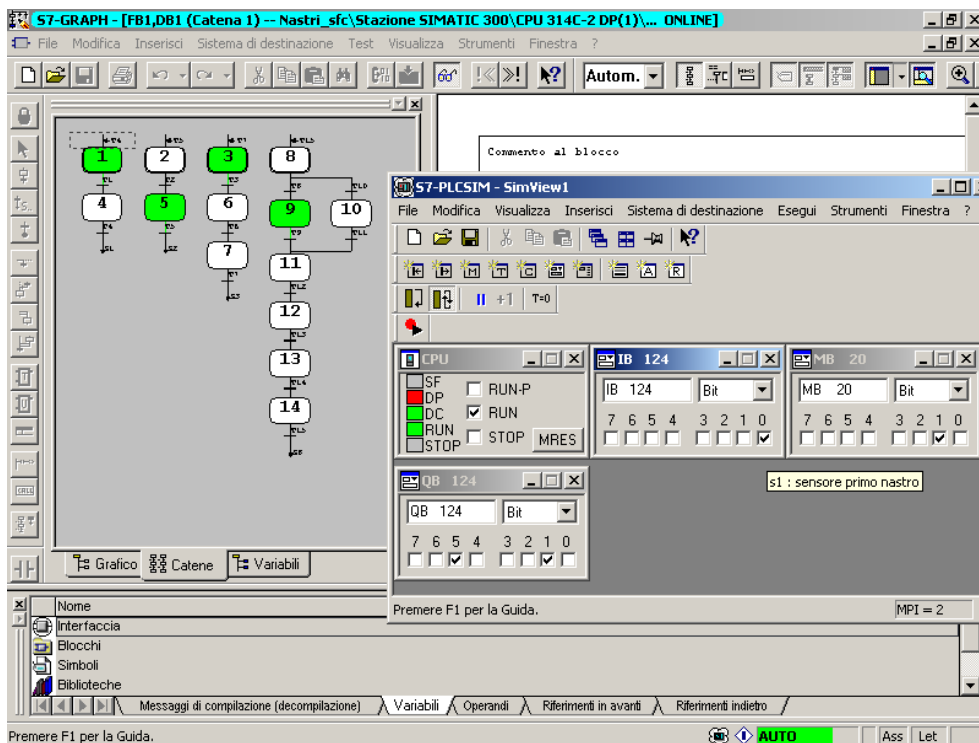
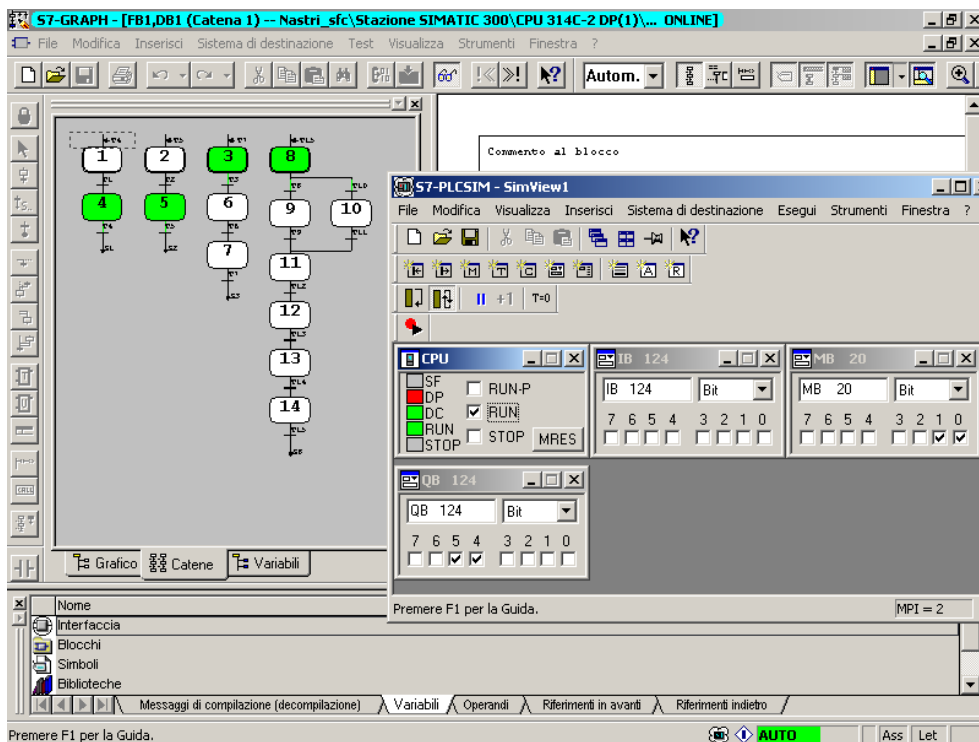
Associare a INIT_SQ (init_sequence) la variabile creata nella tabella dei simboli (INIT_SQ M20.2). Questa consente di inizializzare la catena.

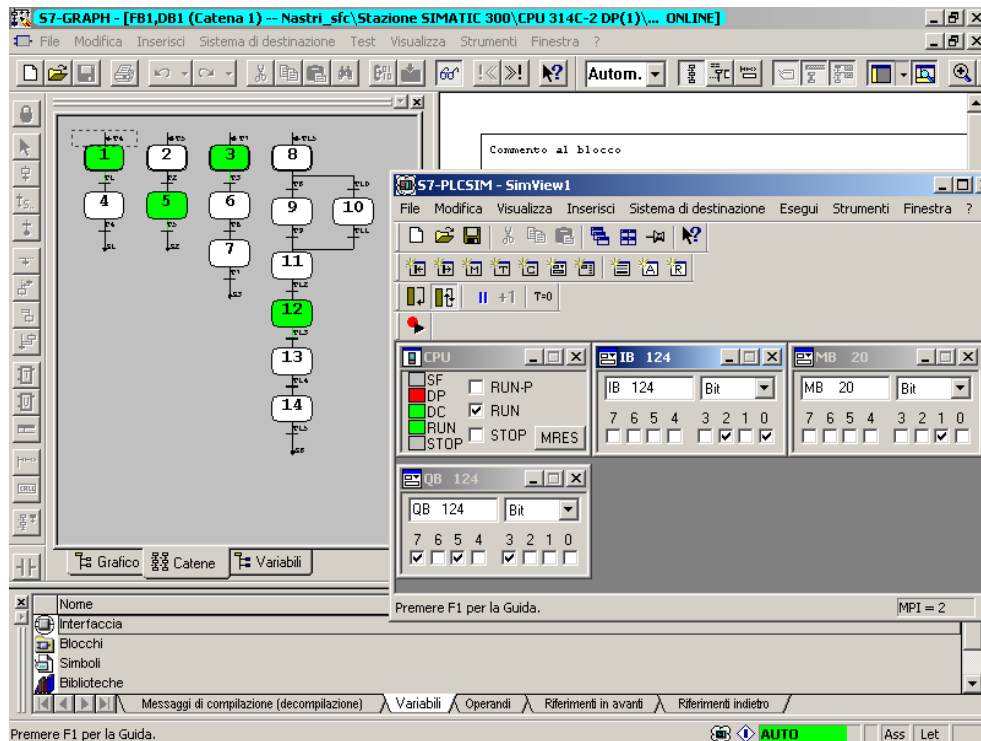
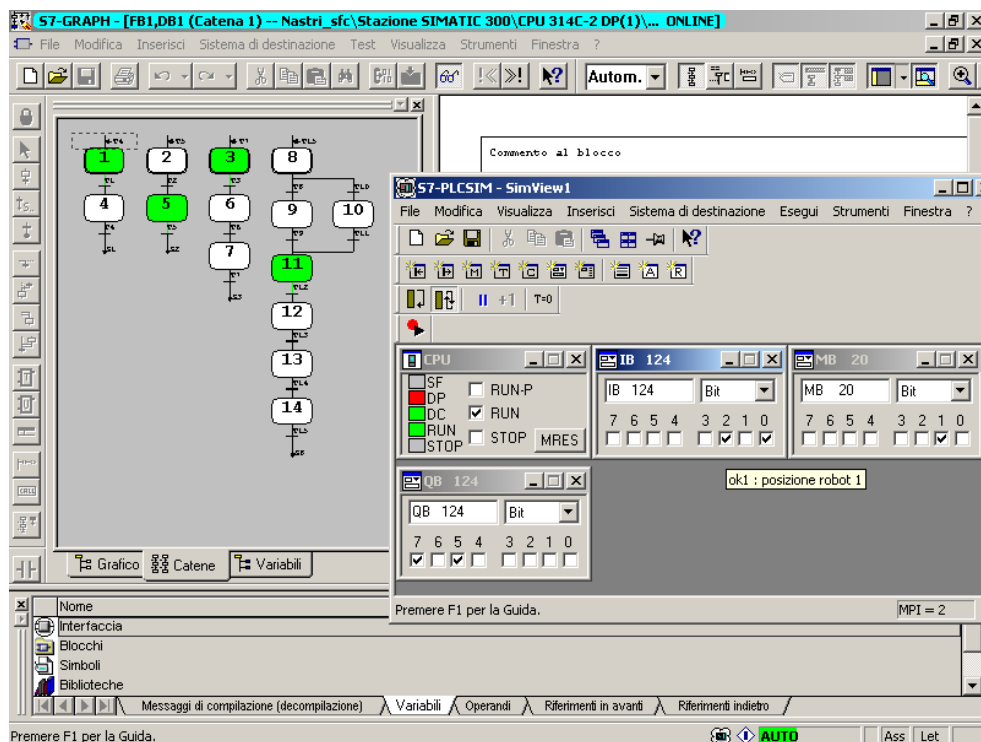


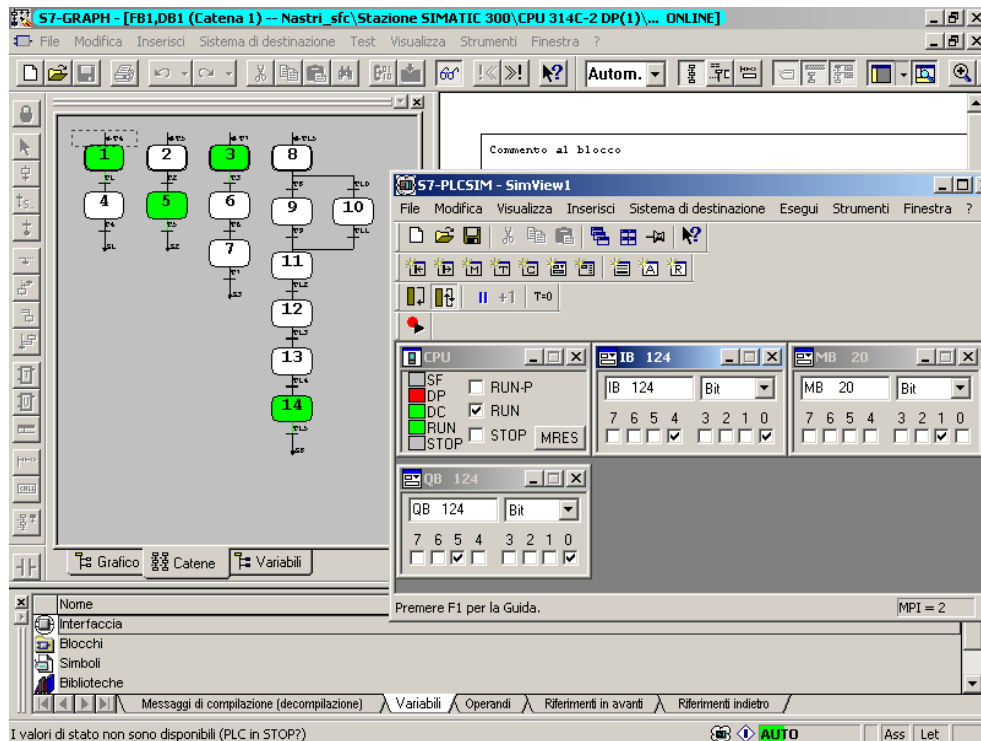
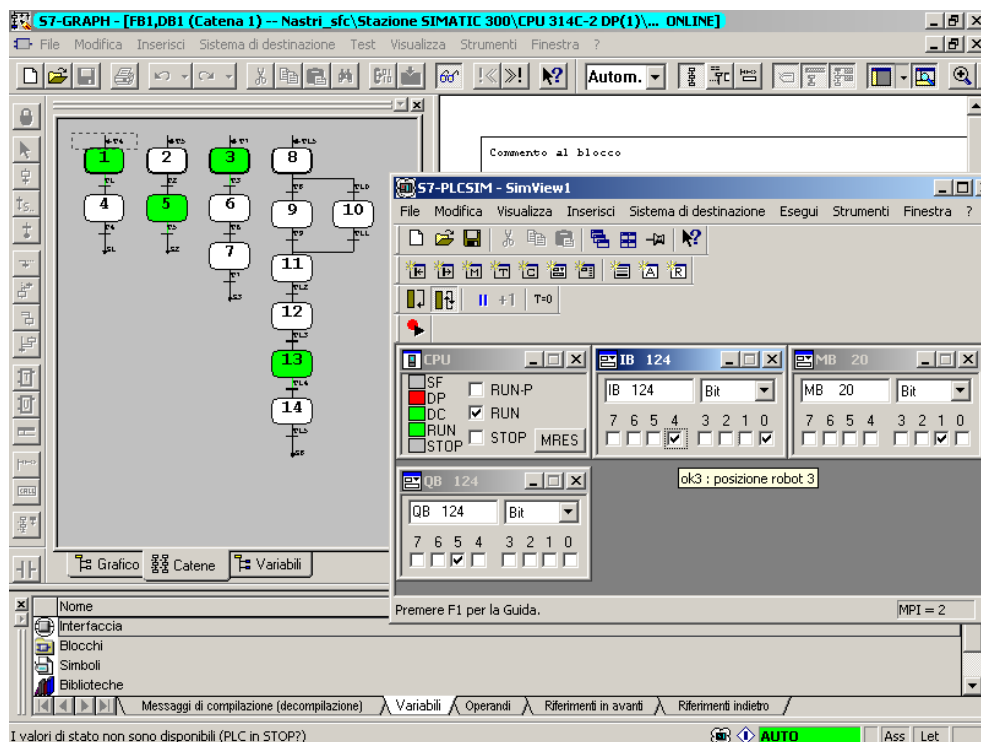
Per eseguire la simulazione del progetto:

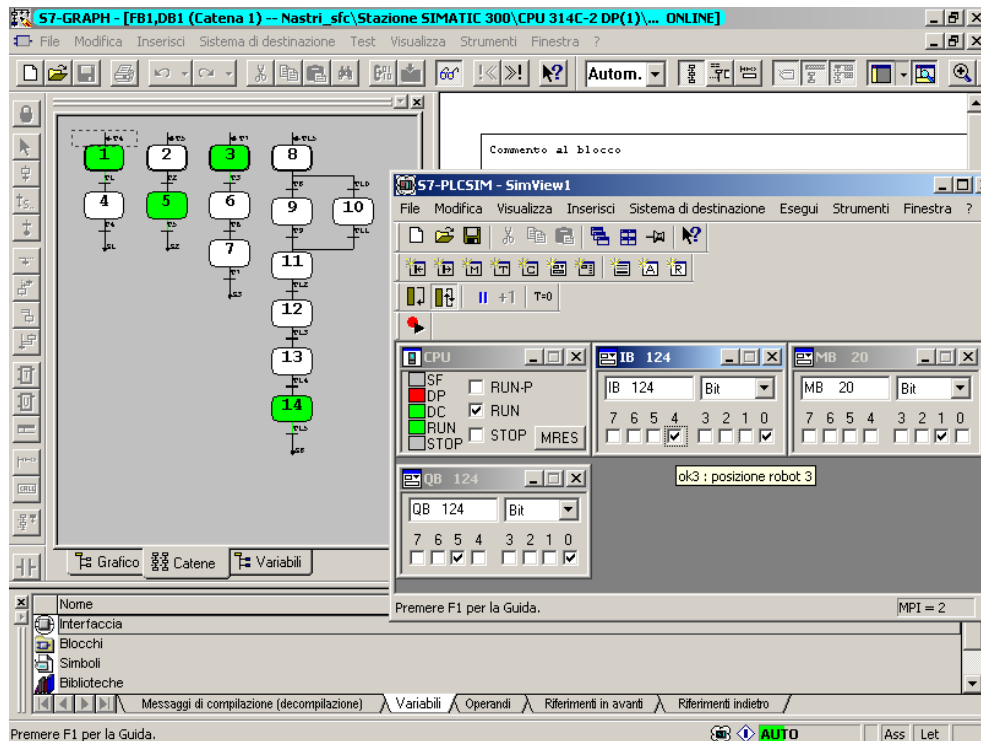
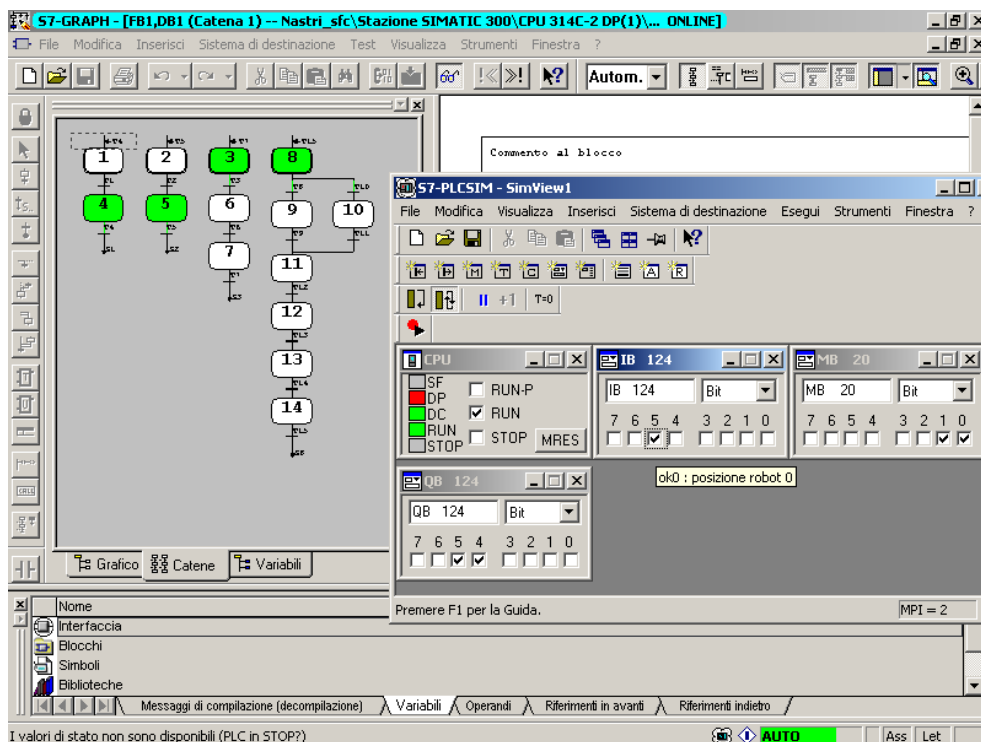
- seguire gli stessi passi di **1.8 Simulazione con S7-PLCSIM**
- cliccare su  e avviare FB1 (non OB1)
- Commutare la CPU o il simulatore in RUN e attivare i vari ingressi.

Di seguito si riportano alcune immagini che sintetizzano alcuni passi della simulazione.









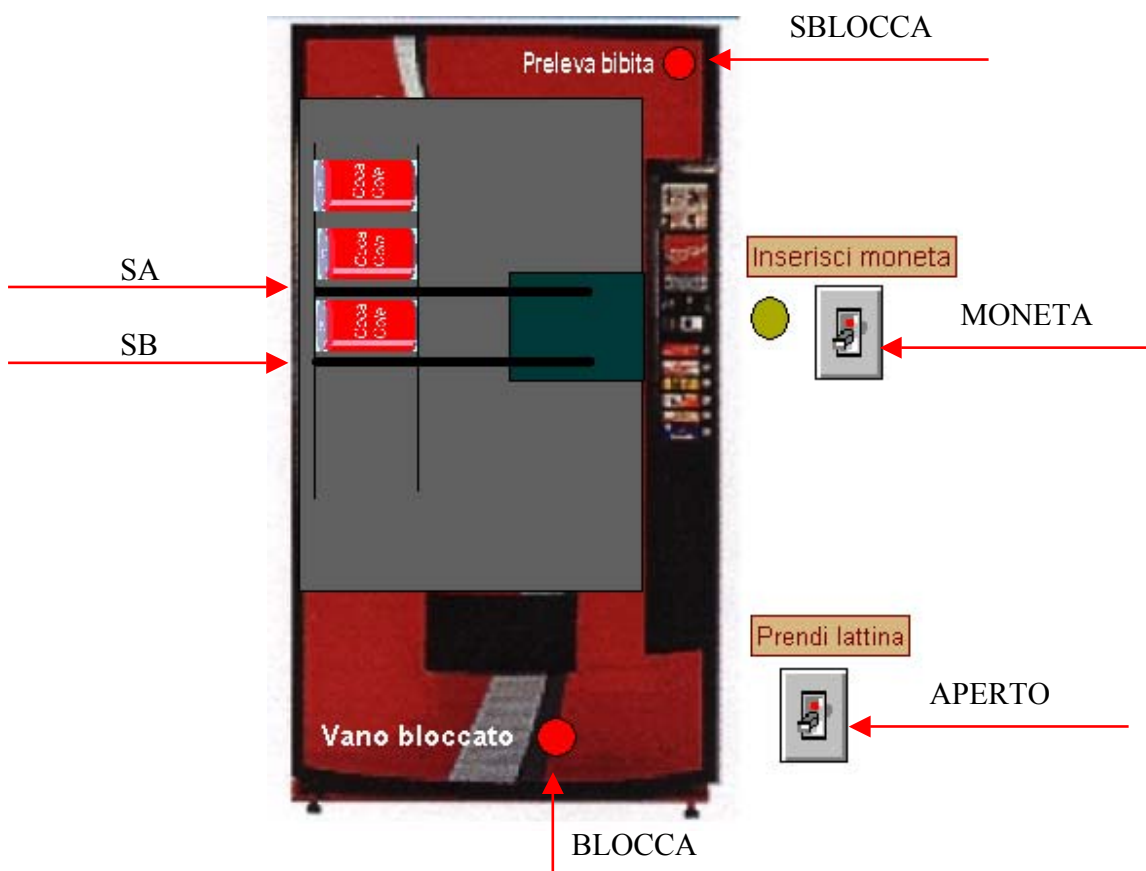
2.3 Svolgimento del progetto - Distributore di Bibite

Il sistema da analizzare è un distributore automatico di bibite. Come mostrato in figura le bibite sono contenute in un vano cilindrico; due pistoni SA, SB trattengono la prima bibita che verrà distribuita. Il sistema distribuisce le bibite a pagamento, per semplicità si suppone che è sufficiente l'inserimento di una sola moneta per il rilascio di una lattina. Un sensore segnala l'inserimento della moneta nella gettoniera.

Quando una moneta viene inserita il pistone SB che trattiene la bibita si ritrae e lascia cadere la lattina nel vano di prelievo, per consentire la caduta della lattina il pistone rimane ritratto per un intervallo di tempo t . Il vano di prelievo è provvisto di uno sportello con un controllo di sicurezza che sblocca lo sportello solo quando la bibita è pronta per essere prelevata. Superato l'intervallo t il pistone SB ritorna nella posizione iniziale. A questo punto è necessario che una nuova lattina si posizioni fra i due pistoni; per fare ciò il pistone SA si ritrae e lascia cadere la successiva lattina, rimanendo ritratto un intervallo t per poi tornare nella posizione iniziale.

Il distributore è dotato anche di un sistema controllo-utente che permette di sapere tramite sensore sullo sportello quando l'utente lo apre per prelevare la lattina.

Gli ingressi della parte di controllo sono il segnale MONETA dalla gettoniera e il segnale APERTO che indica che il vano di prelievo è stato aperto. Le uscite sono i segnali SBLOCCA e BLOCCA che agiscono sulla porta del vano di prelievo e i segnali SA, SB per il comando dei due pistoni.



Analisi del problema

Individuiamo le variabili del sistema:

Uscite del PLC

SA, SB : variabili digitali per il controllo dei pistoni.

SA, SB = 0 il pistone non è ritratto.

SA, SB = 1 il pistone è ritratto.

BLOCCA, SBLOCCA : variabili digitali che agiscono sullo sportello del distributore

Possibili configurazioni:

Sportello bloccato: BLOCCA = 1, SBLOCCA = 0;

Sportello aperto: BLOCCA = 0, SBLOCCA = 1;

Ingressi del PLC

MONETA : variabile binaria

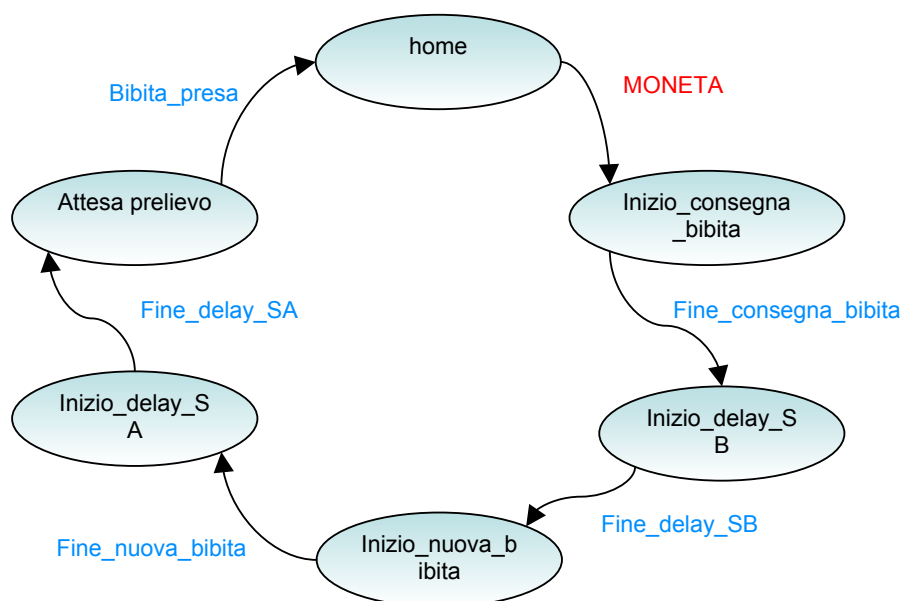
MONETA = 1 moneta inserita;

MONETA = 0 assenza di moneta;

APERTO : mi permette di verificare se lo sportello è stato realmente aperto

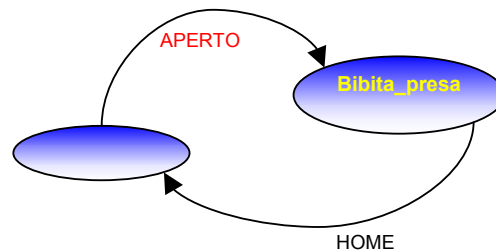
APERTO = 1 la bibita è stata presa

Il sistema inizia il controllo quando è inserita la moneta (MONETA = 1) a questo punto si ritrae il pistone SB che lascia cadere la bibita. Si introduce un timer per gestire l'intervallo t in cui il pistone rimane ritratto; contemporaneamente viene sbloccato lo sportello (BLOCCA = 0, SBLOCCA = 1) e si attende che l'utente prelevi la bibita (APERTO = 1). Dopo di che bisogna fare scendere la prossima lattina di una posizione ritraendo il pistone SA e attivando un secondo timer per gestire l'intervallo t in cui il pistone rimane ritratto.

**Prima Parte**

- 1) **Home**: stato iniziale; SA = SB = 0 BLOCCA = 1 SBLOCCA = 0 Bibita_presa = 0
Bibita_presa è una variabile locale che viene resettata nello stato di home. Questo

segnale viene introdotto perché l'azione dell'utente che preleva la bibita è indipendente dalle azioni del PLC. Per tale motivo sarà gestita una macchina a stati parallela che setta la variabile Bibita_presa non appena APERTO = 1; la variabile locale non cambia il suo stato.



Il passaggio allo stato successivo avviene se la variabile d'ingresso MONETA = 1.

2) Inizio consegna bibite

SA = 0

SB = 1

BLOCCA = 0

SBLOCCA = 1

Il passaggio di stato si ha mediante il segnale fine_consegna_bibite fornito dal timer che descrive il tempo in cui il pistone SB rimane ritratto.

3) Inizio_delay SB

SA = 0

SB = 0

BLOCCA = 0

SBLOCCA = 1

Il passaggio di stato si ha tramite il segnale fine_delay_SB che diventa uno allo scadere del timer.

4) Inizio nuova bibita :

SA = 1

SB = 0

Si fanno scendere le altre bibite solo dopo che si è sicuri che il pistone SB sia tornato al suo posto.

Il passaggio allo stato successivo si ha mediante il segnale fine_nuova_bibita che sarà messo a uno quando il timer indica che il pistoncino è stato ritratto.

5) Inizio_delay SA:

SA = 0

SB = 0

Il passaggio di stato si ha tramite il segnale fine_delay_SA che diventa uno allo scadere del timer.

6) Attesa Prelievo:

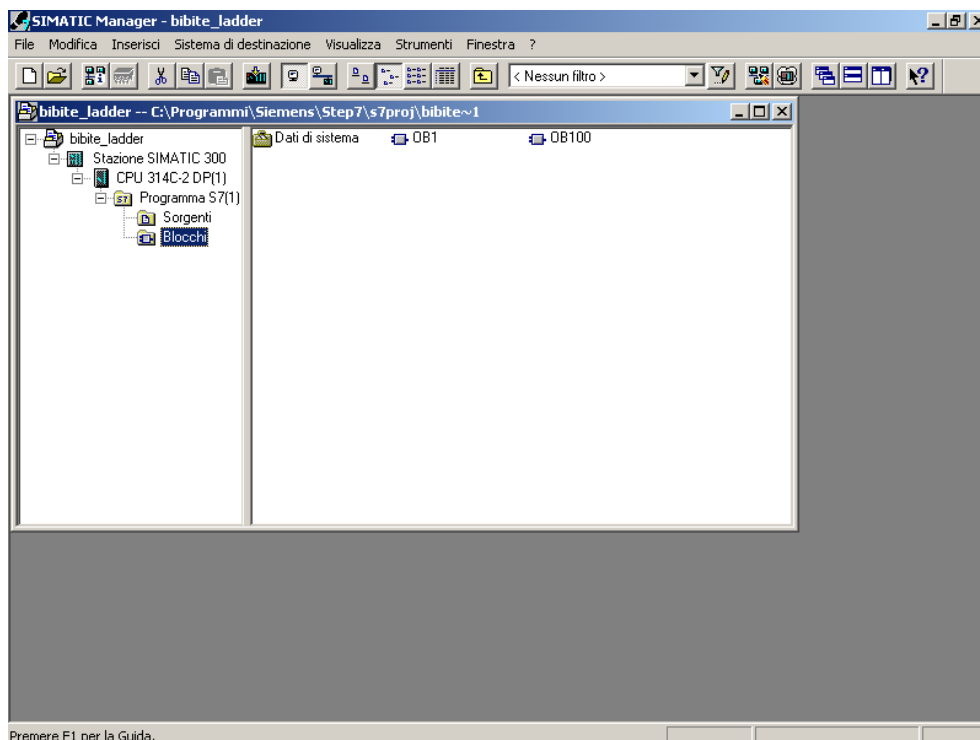
Stato che ci permette di sincronizzarci con la macchina a stati parallela.

Si ritorna a home quando bibita_presa è uno.

2.3.1 Progetto dei Nastri Trasportatori e del Robot realizzato in linguaggio KOP (Ladder)

Aprire il Simatic Manager e avviare la procedura **File > Assistente "Nuovo progetto"**. Scegliere la CPU314C-2DP (si ricordi che nella configurazione hardware bisogna impostare la versione firmware 2.0), selezionare OB1 e OB100, scegliere come linguaggio di programmazione KOP e indicare il nome del progetto "bibite_ladder".

Si otterrà:



Per effettuare la programmazione con nomi simbolici bisogna creare una tabella dei simboli.

Aprire la cartella "Simboli" che si trova dentro **Programma S7(1)** e inserire le varie righe, tenendo conto di quanto detto nel paragrafo 1.3 *Indirizzi e tipi di dati ammessi nella tabella dei simboli* (vedere anche pag. 33).

Quello che si ottiene è:

Stato	Simbolo	Indirizzo	Tipo di dati	Commento
	MONETA	I 124.0	BOOL	
	APERTO	I 124.1	BOOL	
	home	M 20.0	BOOL	
	inizio_consegna_bibita	M 20.1	BOOL	
	inizio_delay_SB	M 20.2	BOOL	
	inizio_nuova_bibita	M 20.3	BOOL	
	inizio_delay_SA	M 20.4	BOOL	
	attesa_prelievo	M 20.5	BOOL	
	fine_consegna_bibita	M 20.6	BOOL	
	fine_delay_SB	M 20.7	BOOL	
	fine_nuova_bibita	M 21.0	BOOL	
	fine_delay_SA	M 21.1	BOOL	
	bibita_presa	M 21.2	BOOL	
	Cycle Execution	OB 1	OB 1	
	COMPLETE RESTART	OB 100	OB 100	Complete Restart
	SA	Q 124.0	BOOL	
	SB	Q 124.1	BOOL	
	BLOCCA	Q 124.2	BOOL	
	SBLOCCA	Q 124.3	BOOL	
	timer_SB	T 0	TIMER	
	timer_delay_SB	T 1	TIMER	
	timer_SA	T 2	TIMER	
	timer_delay_SA	T 3	TIMER	

Premere F1 per accedere alla Guida.

Per scrivere il programma in KOP (Ladder) cliccare su OB1. Si aprirà la maschera “KOP/AWL/FUP” dove si inseriranno le varie istruzioni.

Il programma si articola in segmenti. Quindi bisogna selezionare “inserisci segmento” per cominciare a introdurre i simboli del linguaggio ladder. Sulla sinistra si trova la finestra “elementi di programma”. Questa contiene tutta la simbologia relativa al linguaggio KOP (Ladder).

Ogni qualvolta si inserisce un elemento dentro il segmento esso presenterà dei punti interrogativi rossi. Cliccando su di essi è possibile inserire l'indirizzo assoluto oppure se si è costruita la tabella dei simboli è possibile scegliere il nome nella casella di riepilogo visualizzata.

Il programma in KOP sarà composto da tre fasi:

- stato attuale – ingresso – nuovo stato
- nuovo stato – resettaggio vecchio stato
- stato attuale – condizioni che devono essere compiute

SIMATIC

OB1

22/02/2005 12.30.53

OB1 - <offline>

"Cycle Execution"

Nome: Famiglia:

Autore: Versione: 0.1

Versione blocco: 2

Data e ora Codice: 17/02/2005 09.26.53

Interfaccia: 15/02/1996 16.51.12

Lunghezza (blocco / codice / dati) 00340 00194 00022

Nome	Tipo di dati	Indirizzo	Commento
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started

Blocco:OB1 "Main Program Sweep (Cycle)"

Segmento: 1

```

"home"      "MONETA"      "inizio_co
nsegna_bib
ita"      <s>

```

Segmento: 2

```

"inizio_co  "fine_cons
nsegna_bib egna_bibit
ita"      a"      "inizio_de
lay_SB"      <s>

```

Segmento: 3

```

"inizio_de  "fine_dela
lay_SB"      y_SB"      "inizio_nu
ova_bibita
"      <s>

```

SIMATIC

OB1

22/02/2005 12.30.53

Segmento: 4

```
"inizio_nu  
ova_bibita" "fine_nuov  
"           a_bibita"           "inizio_de  
lay_SA"           <S>
```

Segmento: 5

```
"inizio_de" "fine_dela"           "attesa_pr  
lay_SA"     y_SA"           elievo"           <S>
```

Segmento: 6

```
"attesa_pr" "bibita_pr"           "home"  
elievo"     esa"           <S>
```

Segmento: 7

```
"inizio_co  
nsegna_bib  
ita"           "home"  
               <R>
```

Segmento: 8

```
"inizio_de"           "inizio_co  
lay_SB"           nsegna_bib  
               ita"           <R>
```

Segmento: 9

```
"inizio_nu  
ova_bibita"           "inizio_de  
"           lay_SB"           <R>
```

SIMATIC

OB1

22/02/2005 12.30.53

Segmento: 10



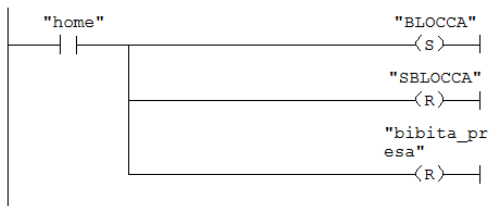
Segmento: 11



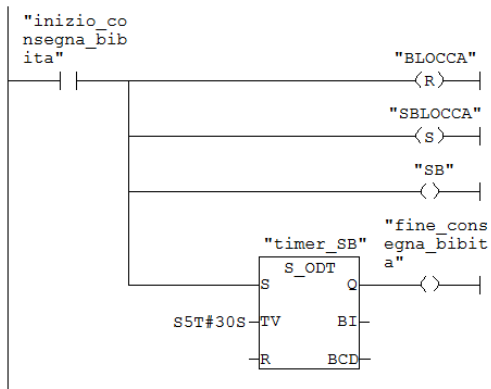
Segmento: 12



Segmento: 13



Segmento: 14

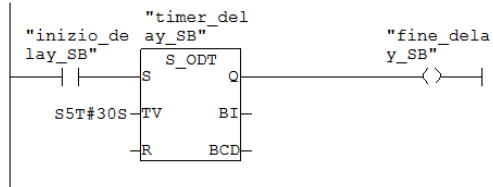


SIMATIC

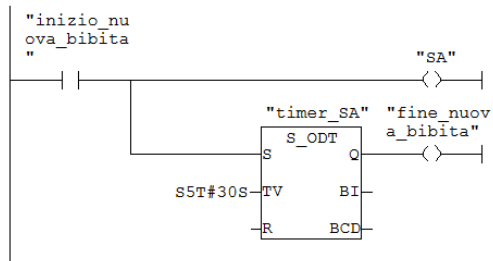
OB1

22/02/2005 12.30.53

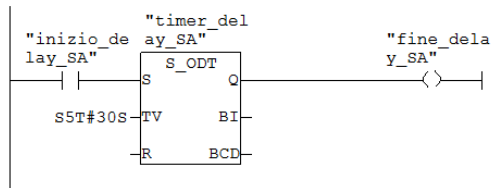
Segmento: 15



Segmento: 16



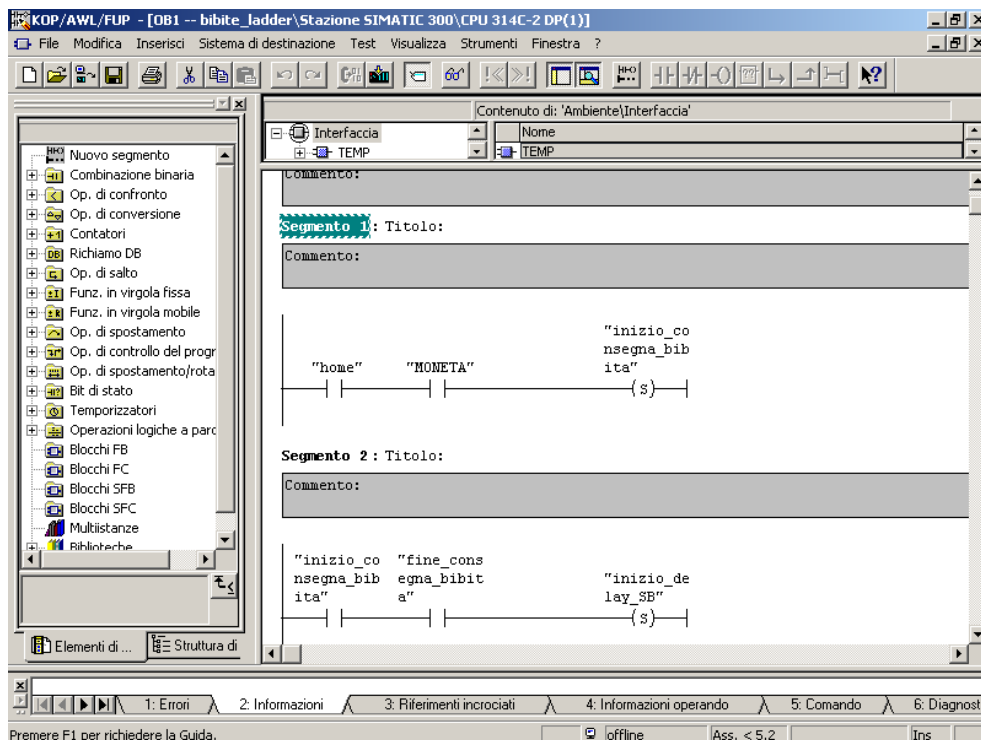
Segmento: 17



Segmento: 18



Di seguito viene riportato un esempio dell'interfaccia "KOP/AWL/FUP"



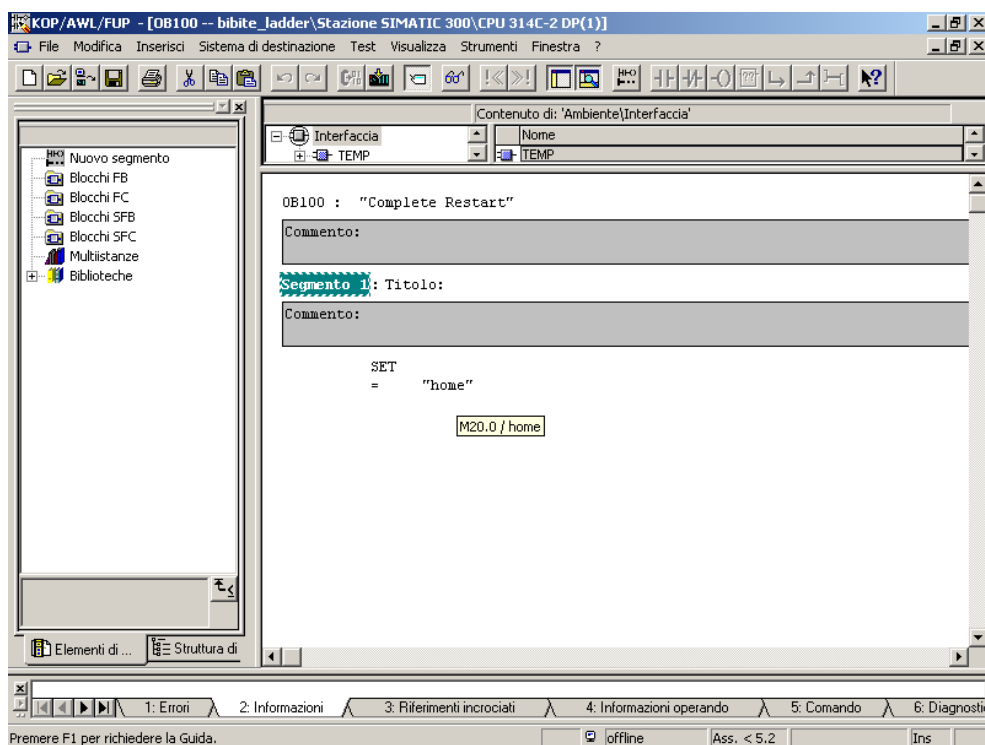
Dopo aver salvato il programma, aprire il blocco di avviamento OB100 per settare la variabile home a true.

Scegliere dal menu **Visualizza > AWL** e inserire la seguente istruzione:

SET

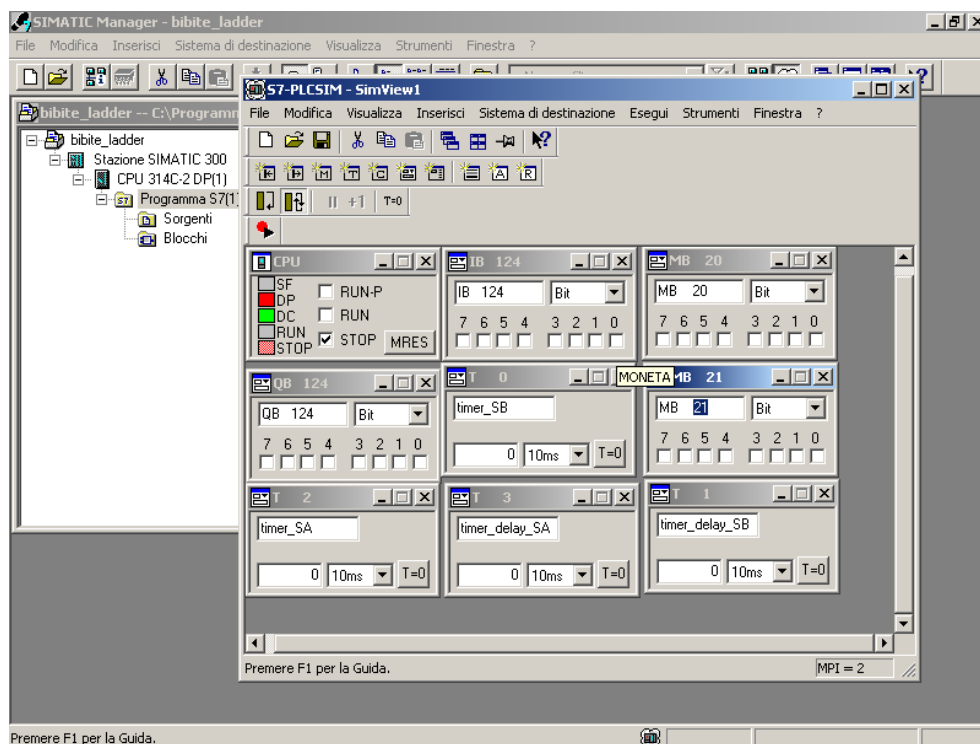
= "home"

Si ricorda che AWL (lista istruzioni) è un linguaggio di programmazione testuale vicino al linguaggio macchina. Quando si crea un programma in AWL, le singole istruzioni corrispondono in larga misura alle operazioni con le quali la CPU elabora il programma.



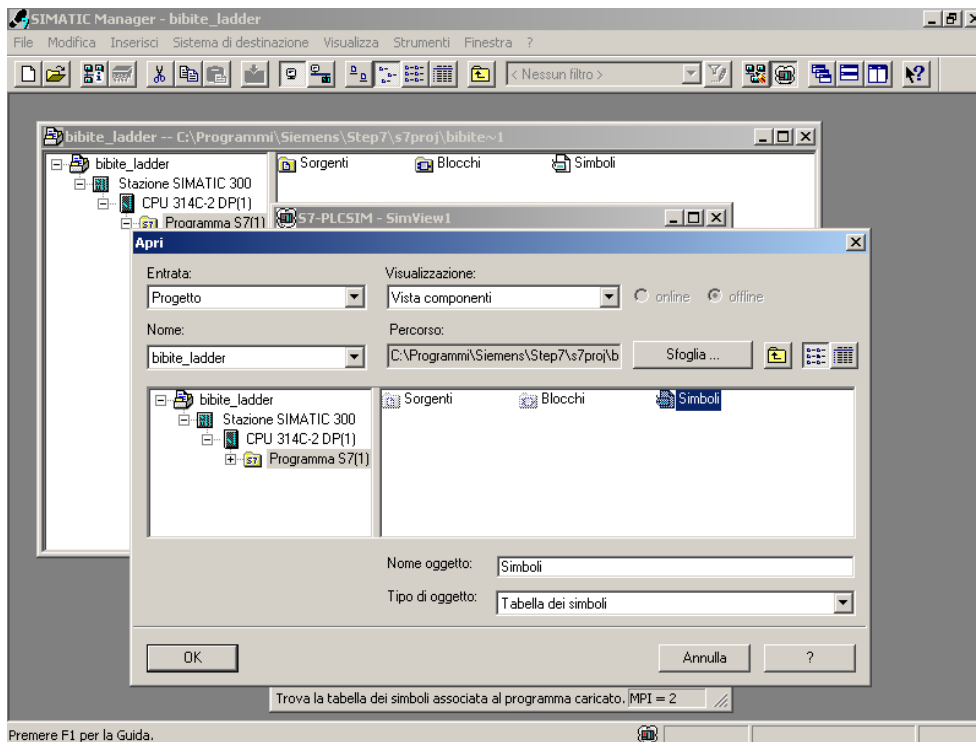
Per simulare il progetto seguire i passi dei paragrafi:

- 1.8 Simulazione con S7-PLCSIM
- 1.9 Come utilizzare STEP 7 per controllare la simulazione del programma



Come si vede dall'immagine è possibile inserire ingressi, uscite, merker e temporizzatori. Inoltre è possibile associare ad ogni elemento (IB,MB,QB, T0) il corrispondente simbolo della tabella dei simboli.

Basta andare su **Strumenti > opzioni > aggiungi simboli**. Si aprirà la maschera "Apri" che permette di selezionare la tabella dei simboli relativa al progetto in esame.



2.3.2 Progetto del Distributore di Bibita realizzato in linguaggio SFC

Nella prima fase bisogna aprire il Simatic Manager e richiamare il comando di menu **File > Assistente "Nuovo progetto"**. Creare un nuovo progetto "Nastri_sfc".

Per usare la notazione simbolica è necessario creare la tabella dei simboli.

Si ricorda che in figura è riportata la tabella dei simboli ottenuta dopo il salvataggio e la compilazione del file S7GRAPH.

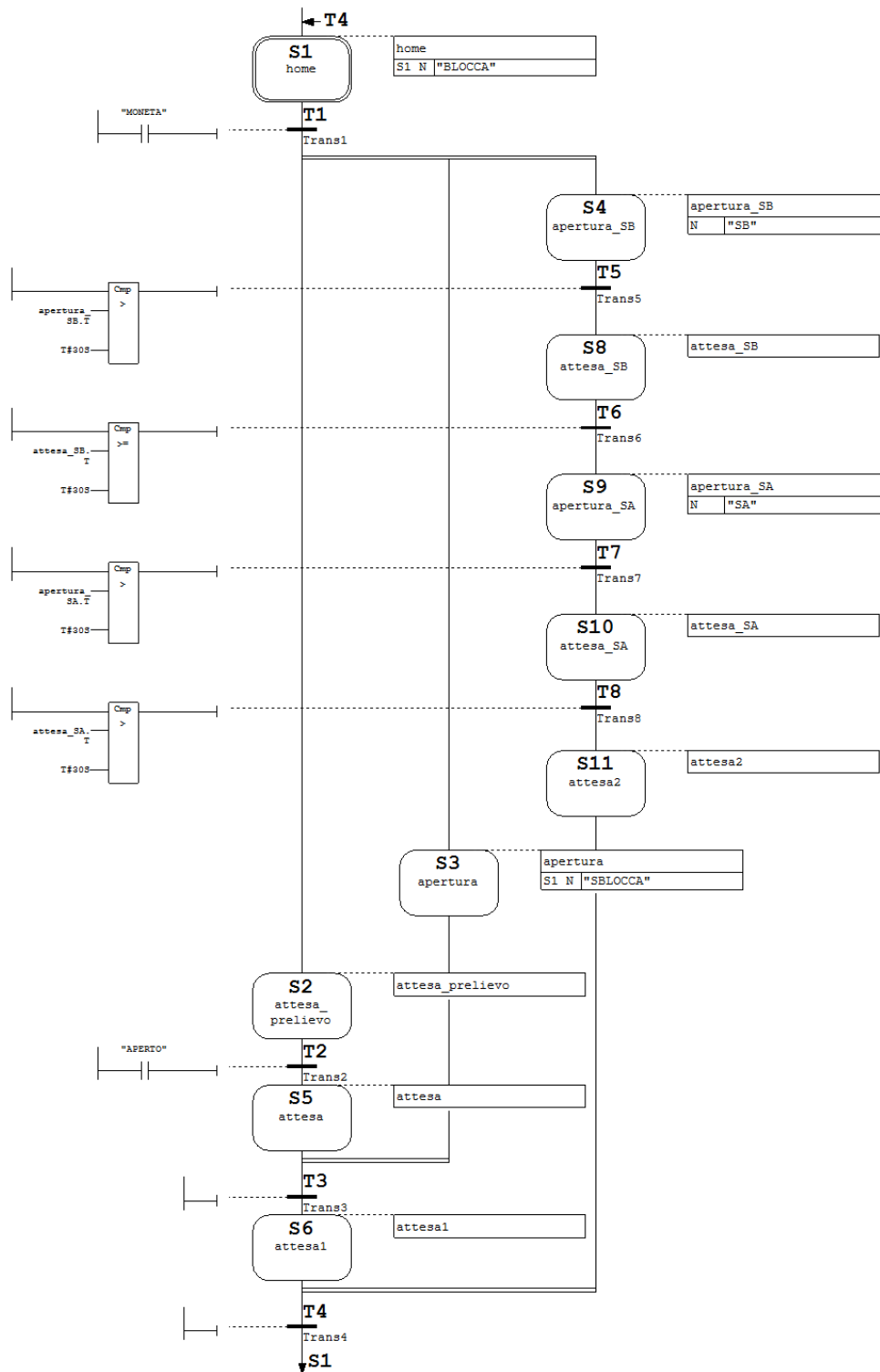
Inizialmente è sufficiente inserire solo gli ingressi e le uscite.

Stato	Simbolo	Indirizzo	Tipo di dati	Commento
1	G7_STD_3	FC 72	FC 72	
2	MONETA	I 124.0	BOOL	
3	APERTO	I 124.1	BOOL	
4	INIT_SQ	M 20.0	BOOL	
5	Cycle Execution	OB 1	OB 1	
6	SA	Q 124.0	BOOL	
7	SB	Q 124.1	BOOL	
8	BLOCCA	Q 124.2	BOOL	
9	SBLOCCA	Q 124.3	BOOL	
10	TIME_TCK	SFC 64	SFC 64	Read the System Time
11				

Dopo aver creato il progetto e aver definito la tabella dei simboli portarsi dentro "Blocchi" e con il tasto destro del mouse selezionare **Inserisci nuovo oggetto > Blocco funzionale**. Si aprirà la finestra "Proprietà- Blocco funzionale" e nella casella linguaggio dobbiamo scegliere "GRAPH".

Cliccare su FB1 e si aprirà S7_GRAPH.

Di seguito viene mostrato il codice SFC del progetto in esame.



Step Home :

Allo step iniziale è associata la seguente azione: S1 N "BLOCCA". Dopo che si è verificato l'evento (S1=step attivo) , per il ciclo del passo successivo vale quanto segue: il segnale dell'operando è per una volta = 1 (N)

Transizione T1:

Alla prima transizione si associa un contatto normalmente aperto MONETA.

A questo punto si ha una diramazione simultanea.

Step attesa_prelievo:

Allo step attesa_prelievo non è associata alcuna azione

Transizione T2:

Alla seconda transizione associamo un contatto normalmente aperto "APERTO"

Step apertura :

A tale step associamo l'azione S1 N "SBLOCCA" Dopo che si è verificato l'evento (S1=step attivo) , per il ciclo del passo successivo vale quanto segue: il segnale dell'operando è per una volta = 1 (N)

A questo punto si ha un "percorso convergente" in quanto le due sequenze alternative convergono.

Transizione T3 è vuota.**Step attesa1 :**

A tale step non è associata alcuna azione

Step apertura_SB :

A tale step associamo l'azione N "SB" questa indica che finchè il passo è attivo SB è 1.

Transizione T5:

la transizione è composta un comparatore dove confrontiamo apertura_SB.T, che rappresenta il tempo di attivazione corrente o più recente del passo, con un tempo prefissato di 30 secondi (T#30S).

Step attesa_SB :

A tale step non è associata alcuna azione.

Transizione T6:

la transizione è composta un comparatore dove confrontiamo attesa_SB.T, che rappresenta il tempo di attivazione corrente o più recente del passo, con un tempo prefissato di 30 secondi (T#30S).

Step apertura_SA :

A tale step associamo l'azione N "SA" questa indica che finchè il passo è attivo SA è 1

Transizione T7:

la transizione è composta un comparatore dove confrontiamo apertura_SA.T, che rappresenta il tempo di attivazione corrente o più recente del passo, con un tempo prefissato di 30 secondi (T#30S).

Step attesa_SA :

A tale step non è associata alcuna azione.

TransizioneT8:

la transizione è composta un comparatore dove confrontiamo attesa_SA.T, che rappresenta il tempo di attivazione corrente o più recente del passo, con un tempo prefissato di 30 secondi (T#30S).

Step attesa2 :

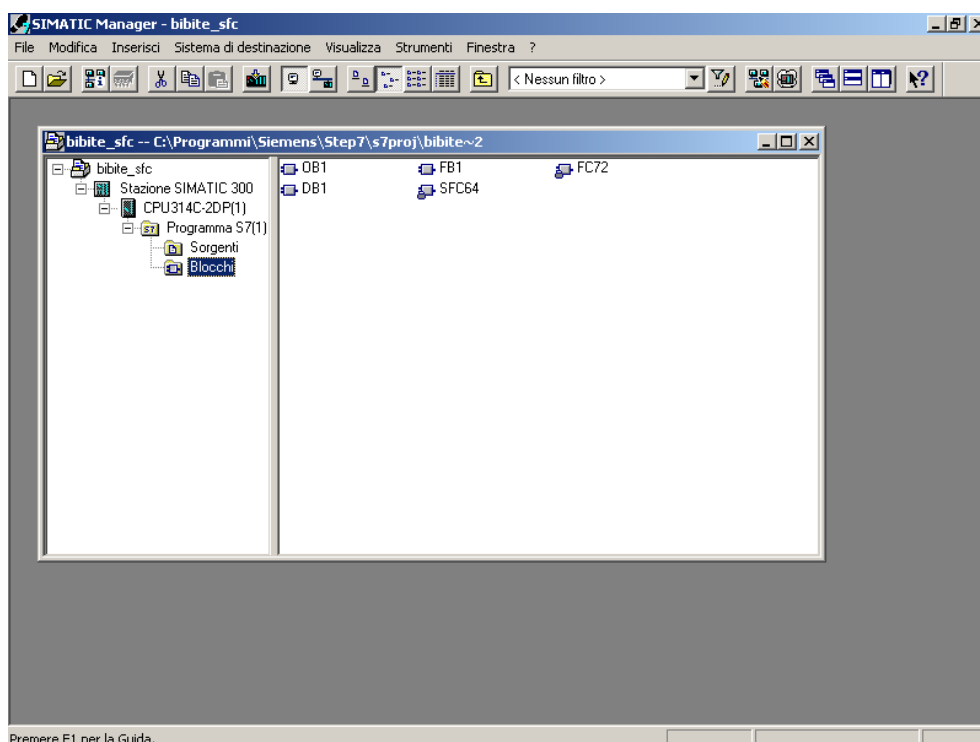
A tale step non è associata alcuna azione

A questo punto si ha un “percorso convergente” in quanto le due sequenze alternative convergono.

Transizione T4 è vuota.

Dopodichè si ritorna allo step Home

Dopo aver salvato e compilato il progetto realizzato in S7_GRAPH si troveranno nella cartella “Blocchi” i seguenti elementi generati dalla compilazione

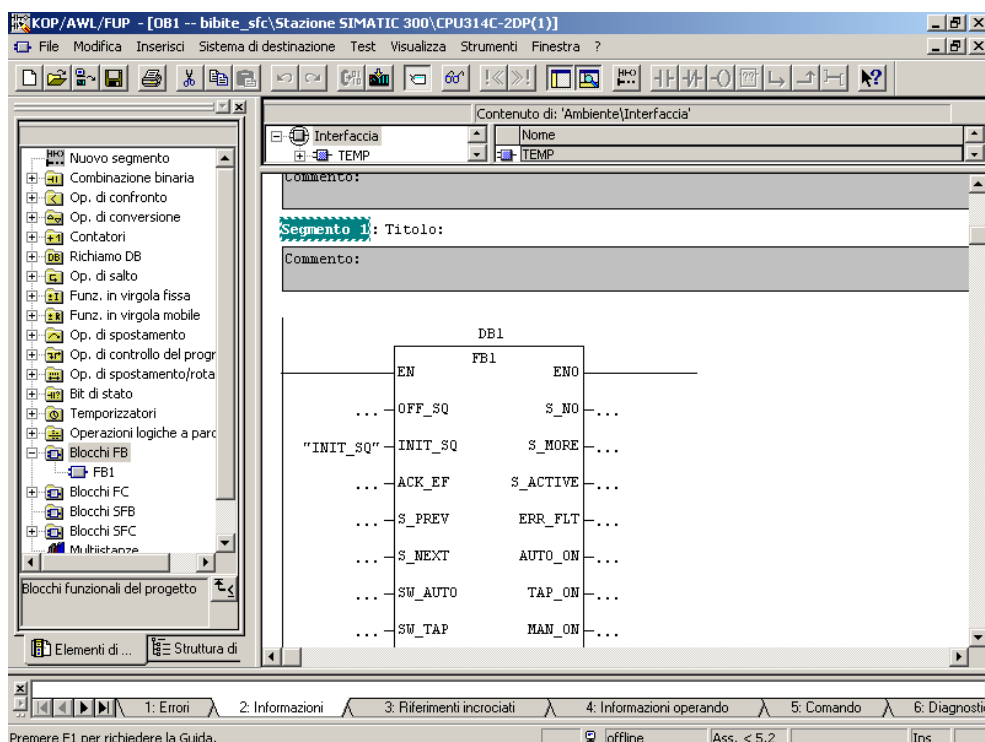


Per realizzare il richiamo di FB1 all'interno del blocco organizzativo 1 è necessario aprire OB1 e dalla finestra “elementi di programma”, posizionarsi su “Blocchi” e scegliere FB1.


Introdurre al posto dei punti interrogativi DB1.

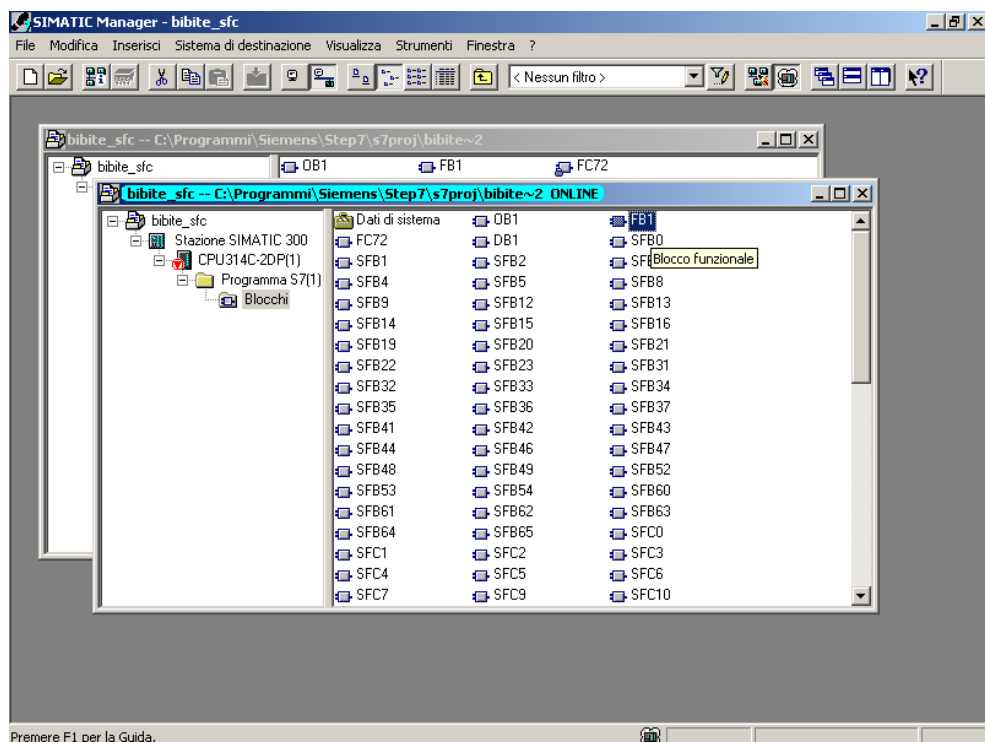
Si ricorda che i blocchi dati di istanza(DB), generati automaticamente nella compilazione, vengono assegnati al blocco quando viene richiamato un FB.

Associare a INIT_SQ (init_sequence) la variabile creata nella tabella dei simboli (INIT_SQ M20.0). Questa consente di inizializzare la catena.



Per eseguire la simulazione del progetto

- seguire gli stessi passi di **1.8 Simulazione con S7-PLCSIM**
- cliccare su  e avviare FB1 (non OB1)



- Commutare la CPU o il simulatore in RUN e attivare i vari ingressi.

Di seguito si riportano alcune immagini che sintetizzano alcuni passi della simulazione.

