

STEP 7

L'ambiente di sviluppo **STEP 7** è un pacchetto software creato per sviluppare *progetti di automazione* basati sui prodotti Siemens della serie SIMATIC.

Le informazioni relative alle caratteristiche di questo ambiente vengono desunte dai manuali della casa costruttrice e fanno particolare riferimento alla versione 5.4.

L'**ambiente di sviluppo** è composto da:

- un software di base;
- un numero di componenti aggiuntivi, eseguibili su piattaforme Windows.

I **progetti** sono costituiti dall'insieme dei dati e dei programmi.

In particolare un progetto contiene:

- dati di configurazione relativi alla struttura hardware e dati di parametrizzazione per le unità;
- dati di progettazione per le comunicazioni di rete;
- programmi per le unità programmabili.

Con il pacchetto base di STEP 7 è possibile:

- creare e gestire progetti;
- configurare e parametrizzare l'hardware e le comunicazioni dei sistemi di automazione;
- creare programmi per i PLC delle serie SIMATIC S7;
- caricare i programmi nei sistemi di destinazione;
- validare il software sviluppato.

Per la programmazione STEP 7 mette a disposizione dell'utente i seguenti linguaggi:

- KOP (schema a contatti);
- FUP (schema funzionale);
- AWL (lista di istruzioni AWL);
- SCL (testo strutturato).

La versione Siemens di questi linguaggi ha come corrispondente quelli specificati dallo standard IEC 61131.

I linguaggi KOP, FUP e AWL consentono di scrivere blocchi di codice direttamente caricabili nel controllore, senza dover effettuare alcuna compilazione.

I sorgenti SCL vanno invece compilati prima di poter essere caricati nel controllore.

Sviluppo di una soluzione di automazione

Facendo ricorso ad un semplice esempio suggerito direttamente dalla Siemens viene fatto cenno ad un possibile criterio che può essere utilizzato per la *progettazione di soluzioni di automazione* e per lo *sviluppo di programmi utente per sistemi di automazione*.

Viene analizzato in particolare un **processo di miscelazione industriale** che presenta le seguenti caratteristiche:

- due pompe di alimentazione immettono due sostanze A e B in un serbatoio di miscelazione;

- le valvole di aspirazione e di alimentazione consentono o impediscono l'afflusso delle sostanze nel serbatoio di miscelazione;
- un motore viene utilizzato per miscelare le due sostanze immesse nel serbatoio;
- la miscela passa allo stadio successivo del processo attraverso una valvola di scarico;
- per bloccare la pompa di alimentazione e il motore del miscelatore vengono utilizzate le informazioni fornite sul livello di riempimento nel serbatoio fornite da appositi interruttori.

Suddivisione in aree funzionali

Il sistema che viene schematizzato in figura 1 risulta suddiviso in quattro *aree funzionali* rispettivamente dedicate a:

- sostanza A;
- sostanza B;
- serbatoio di miscelazione;
- scarico.

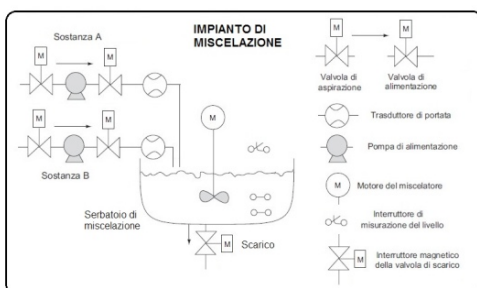


Figura 1 Rappresentazione schematica del processo di miscelazione industriale

La suddivisione del processo in aree funzionali semplifica la realizzazione del sistema.

Descrizione delle aree funzionali

Viene fatto ora riferimento al funzionamento dei componenti che fanno parte delle singole aree.

Le **pompe di alimentazione** vengono attivate o disattivate utilizzando i relativi comandi che si trovano all'interno di un quadro posto nelle vicinanze del serbatoio di miscelazione.

Il numero degli avvii viene contato per la manutenzione; contatore e indicatore possono essere resettati tramite un interruttore comune.

Sono condizioni per l'attivazione:

- il serbatoio di miscelazione non è pieno;
- la valvola di scarico del serbatoio di miscelazione è chiusa;
- non sono presenti situazioni di emergenza che non garantiscono più la sicurezza dell'impianto.

Sono condizioni per la disattivazione:

- il trasduttore di portata non registra alcuna portata dopo sette secondi dall'avvio del motore della pompa;
- il trasduttore di portata non registra più alcuna portata durante il funzionamento.

Le **valvole di aspirazione e di alimentazione**, controllate da un programma utente, sono provviste di un interruttore elettromagnetico dotato di una molla di richiamo.

Il funzionamento previsto è il seguente:

- la valvola è aperta quando l'interruttore elettromagnetico viene attivato;
- la valvola è chiusa quando l'interruttore elettromagnetico viene disattivato.

Devono essere rispettate le seguenti condizioni:

- per l'attivazione il motore della pompa di alimentazione deve funzionare per almeno un secondo;
- per la disattivazione il trasduttore di portata non deve registrare alcuna portata.

Il **miscelatore** viene attivato o disattivato utilizzando i comandi che si trovano all'interno di un quadro di comando posto nelle vicinanze del serbatoio di miscelazione.

Il numero degli avvii viene contato per la manutenzione; contatore e indicatore possono essere resettati tramite un interruttore comune.

Sono condizioni per l'attivazione:

- il misuratore di livello indica un valore inferiore al minimo;
- la valvola di scarico del serbatoio di miscelazione è chiusa;
- non sono presenti situazioni di emergenza che non garantiscono più la sicurezza dell'impianto.

Condizione per la disattivazione è che il trasduttore di velocità segnali, al più tardi dopo 10 s dall'avvio del motore, il mancato raggiungimento del numero di giri nominale.

La **valvola di scarico** incorpora un interruttore elettromagnetico dotato di molla di richiamo.

Il funzionamento previsto è il seguente:

- la valvola è aperta quando l'interruttore elettromagnetico viene attivato;
- la valvola è chiusa quando l'interruttore elettromagnetico viene disattivato.

La valvola di scarico viene aperta o chiusa utilizzando il relativo comando che si trova all'interno di un quadro posto nelle vicinanze del serbatoio di miscelazione.

Sono condizioni per l'apertura della valvola di scarico:

- il motore dell'agitatore è disinserito;
- il misuratore di livello non indica che il serbatoio di miscelazione è vuoto;
- non sono presenti situazioni di emergenza che non garantiscono più la sicurezza dell'impianto.

Condizione per la disattivazione è che il misuratore del livello indichi che il serbatoio di miscelazione è vuoto.

Programmazione

Noto il funzionamento del sistema è possibile definire ingressi, uscite e variabili di controllo per ciascun settore o apparecchiatura che di esso fa parte considerati come blocchi a se stanti.

Si possono nel caso dell'esempio considerare i blocchi motore e i blocchi valvole.

Con la programmazione viene definito il legame che sussiste tra queste variabili.

Considerazioni sulla sicurezza

I dispositivi che si ritengono adatti a garantire i requisiti necessari per la sicurezza del processo vengono scelti facendo riferimento alle vigenti norme di legge e alle procedure aziendali.

Tali apparecchiature richiedono cablaggi particolari in quanto i circuiti elettrici destinati alla sicurezza devono funzionare indipendentemente dal sistema di automazione; tuttavia è spesso presente un'interfaccia che consente l'interazione con il programma utente.

In fase progettuale si rileva la necessità di:

- determinare gli interblocchi logici e meccanici/elettrici tra i singoli compiti di automazione;
- progettare i circuiti in modo tale che, in caso di necessità, sia possibile azionare manualmente le apparecchiature coinvolte nel processo;
- stabilire requisiti di sicurezza supplementari affinché il processo possa essere eseguito senza problemi.

Nell'esempio del processo di miscelazione industriale si utilizza un *interruttore di emergenza* collocato sul quadro di comando che disinserisce entrambe le pompe di alimentazione, il motore del miscelatore e le valvole.

Un ingresso del controllore registra lo stato dell'interruttore di emergenza.

Quadro di comando

Ogni processo richiede un sistema di servizio e supervisione che consenta all'operatore di intervenire manualmente nel corso del processo stesso agendo sulle apparecchiature presenti in un quadro di comando opportunamente progettato.

In figura 2 viene rappresentato un esempio di quadro di comando adatto per il processo di miscelazione industriale.

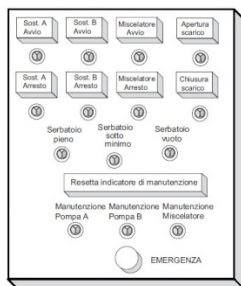


Figura 2 Quadro di comando

Si evidenzia in particolare che:

- l'avvio o l'arresto di ciascuna apparecchiatura avviene utilizzando degli interruttori opportunamente predisposti;
- le informazioni relative allo stato di funzionamento del sistema sono garantite dalla presenza di indicatori;
- sono presenti anche i LED di segnalazione delle apparecchiature che devono essere periodicamente controllate, e l'interruttore di emergenza e un interruttore per il reset dell'indicatore di manutenzione dei tre motori (utilizzato per spegnere i LED dei motori che

Selezionando una cartella del progetto e di seguito il menù *Insert* è possibile definire alcune specifiche del progetto stesso inserendovi:

- una stazione SIMATIC;
- un'altra rete di comunicazione;
- un programma.

Si evidenzia in particolare che:

- si può inserire un programma senza aver definito la stazione sulla quale andrà in esecuzione;
- è necessario definire la stazione di lavoro se si vuole caricare il software sviluppato sul PLC o se si vuole utilizzare il simulatore.

Se ad esempio si desidera inserire un PLC della serie S7-400 si deve selezionare il comando *Station* dal menù *Insert* e di seguito **SIMATIC 400 Station**; con l'assegnazione del nome alla nel progetto comparirà il simbolo della stazione.

Con un procedimento analogo è possibile inserire altre reti di comunicazione che fanno parte del sistema di controllo (per esempio reti Profibus e Industrial Ethernet).

In figura 5 viene riportata la videata iniziale a seguito delle scelte di progetto.

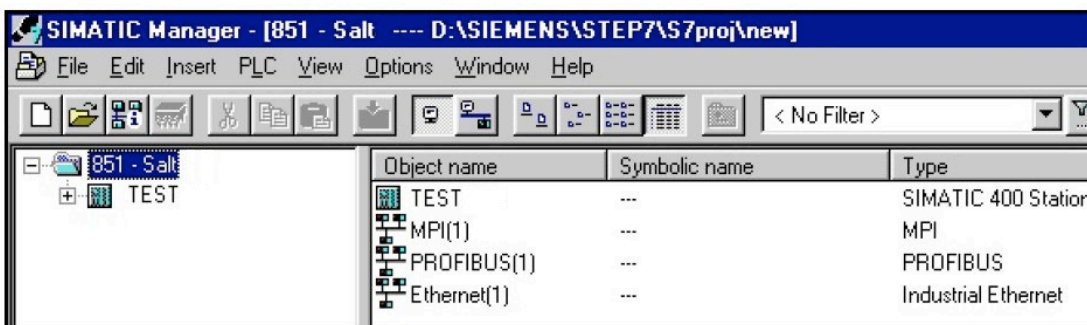


Figura 5 Videata iniziale a seguito delle scelte di progetto

Configurazione hardware

Se si seleziona la stazione SIMATIC compare l'icona *Hardware* riprodotta in figura 6.



Figura 6 Icona hardware della stazione SIMATIC

Selezionando l'icona viene richiamata l'applicazione *HW Config* con la quale è possibile configurare l'hardware di una stazione SIMATIC.

Dalla finestra di HW Config riprodotta in figura 7 si ha accesso ad *Hardware catalog* (finestra di destra) da cui è possibile selezionare i componenti della stazione.

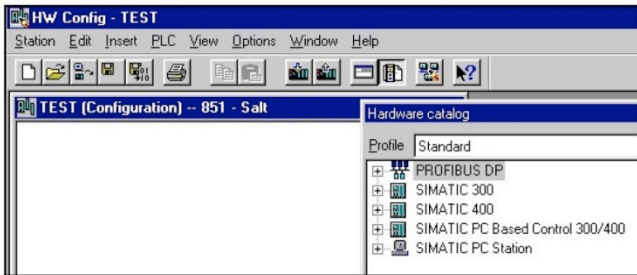


Figura 7 Finestra di HV Config

In corso di definizione nella finestra di sinistra (*Configuration*) comparirà anche la struttura della stazione.

La sequenza di operazioni necessarie per procedere alla configurazione è poi la seguente:

- disporre i telai di montaggio (rack);
- disporre/inserire le unità;
- definire le proprietà delle unità;
- salvare e compilare la configurazione;
- caricare la configurazione nel sistema di automazione.

In STEP 7 le unità che compongono la stazione vengono collocate su *telai di montaggio (rack)*, proprio come in un impianto reale.

L'unica differenza consiste nel fatto che i telai vengono rappresentati mediante *tabelle di configurazione*, caratterizzate da un numero di righe pari a quello delle unità inseribili nel telaio di montaggio reale.

Il primo passo necessario per definire la struttura della stazione è la scelta del *rack* (ad esempio UR1 o *Universal Rack*).

Per l'inserimento delle unità negli slot del telaio una possibile scelta è la seguente:

- slot 1 e 2, Alimentatore 10 A 120-230 V AC;
- slot 3 e 4, Unità Centrale CPU 414-2-DP;
- slot 5, Interfaccia seriale CP 441-1;
- slot 6, Interfaccia Ethernet CP 443-1;
- slot da 7 a 13, Moduli con 32 Ingressi Digitali 24 V DC;
- slot da 14 a 18, Moduli con 32 Uscite Digitali 24 V DC.

Selezionando ciascuna unità è possibile impostarne i parametri premendo il tasto destro del mouse e selezionando *Object Properties*.

Una volta completata la configurazione hardware, dovrà essere compilata per verificarne la correttezza.

Se non sono presenti errori di compilazione e il PC è collegato con il PLC, si può caricare la configurazione fatta nella CPU.

Concluse queste operazioni, come viene evidenziato in figura 8, sono presenti in SIMATIC Manager i moduli processore definiti in precedenza con, al loro interno, dei programmi vuoti.

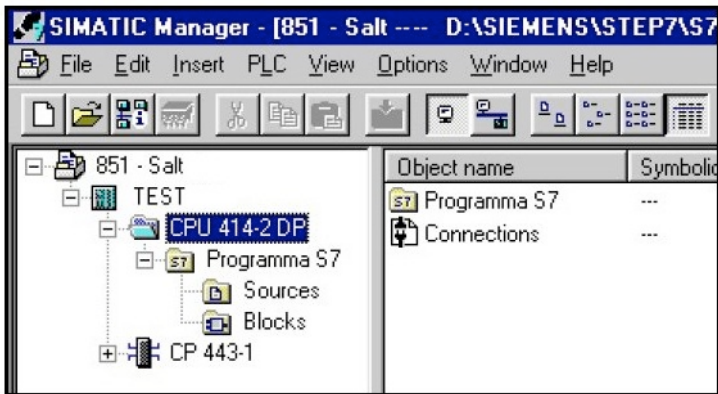


Figura 8 Videata di Simatic Manager al termine della configurazione

In ogni programma sono presenti due cartelle e precisamente:

- *Sources* con i sorgenti del codice;
- *Blocks* con i blocchi eseguibili che dovranno essere caricati nel controllore.

Struttura dei programmi

Nella CPU dei PLC vengono eseguiti due programmi distinti:

- il *sistema operativo*;
- il *programma utente*.

Sistema operativo

Il sistema operativo, presente in tutte le CPU, organizza tutte le funzioni e le procedure che non sono legate ad un compito di controllo specifico.

Il sistema operativo lavora rispettando una modalità di funzionamento particolare denominata **ciclo a copia massiva degli ingressi e delle uscite**.

Le operazioni che vengono svolte ciclicamente sono:

- aggiornare l'immagine di processo degli ingressi;
- eseguire il *programma utente*, operando sui valori della memoria e conservando i risultati sempre in memoria;
- eseguire i *programmi di gestione* del sistema;
- scrivere l'*immagine di processo delle uscite*.

L'**immagine di processo** è una copia degli ingressi e delle uscite che viene conservata in un'apposita area di memoria del controllore.

Il programmatore non accede direttamente alle unità di I/O ma a quest'area di memoria dedicata.

I **programmi di gestione** si occupano di:

- rilevare gli allarmi e richiamare i segmenti di codice da eseguire in caso di allarme;
- gestire la memoria;
- riconoscere e gestire gli errori;
- gestire la comunicazione con i dispositivi di comunicazione e gli altri nodi della rete.

Programma utente

Il **programma utente**, creato e caricato nella CPU, contiene tutte le funzioni necessarie per risolvere il problema di automazione considerato.

L'ambiente di sviluppo STEP 7 offre la possibilità di suddividere il programma utente in singole sezioni indipendenti ottenendo i seguenti vantaggi:

- le singole parti del programma possono essere standardizzate;
- l'organizzazione del programma viene semplificata;
- le modifiche del programma si possono eseguire più facilmente;
- la semplificazione del testing e della validazione del programma.

Il programmatore ha a disposizione dei blocchi elementari attraverso i quali può strutturare il software:

- i *blocchi di codice*;
- i *blocchi di dati*.

Esiste un numero massimo di blocchi inseribili per ogni tipo e questo limite varia a seconda del modulo processore utilizzato.

Un blocco può essere inserito come sorgente in un opportuno linguaggio di programmazione o come direttamente eseguibile e caricabile nella memoria del controllore.

Per inserire un blocco sorgente bisogna aprire la cartella *Sources* da SIMATIC Manager e selezionare *Insert e Software*.

Per inserire un blocco eseguibile in un programma bisogna aprire la cartella *Blocks* da SIMATIC Manager e selezionare *Insert e Block*.

Il procedimento viene illustrato in figura 9.

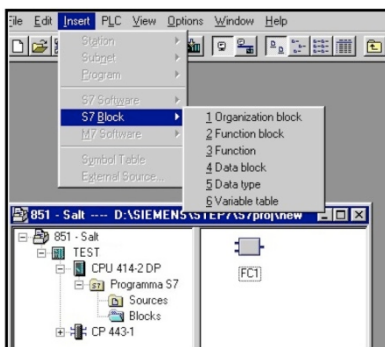


Figura 9 Inserzione di blocchi

Quando si inserisce un blocco eseguibile, compare l'icona corrispondente; selezionandolo il blocco viene aperto e SIMATIC Manager richiama l'editor *AWL, KOP e FUP* attraverso il quale l'utente può scrivere il codice e creare strutture dati.

Blocchi di codice

Vi sono diversi tipi di blocchi di codice che possono essere utilizzati all'interno dei programmi utente; si hanno:

- *blocchi organizzativi (OB)* che determinano la struttura del programma utente;

- *funzioni* (FC) che contengono routine di programma;
- *blocchi funzionali con memoria* (FB);
- *blocchi funzionali di sistema* (SFB) e *funzioni di sistema* (SFC) che sono integrati nella CPU e che rendono accessibili alcune importanti funzioni di sistema.

Quando viene aperto un blocco di codice eseguibile nell'editor compaiono due finestre; quella superiore contiene le dichiarazioni delle variabili, quella inferiore il codice; il codice può essere suddiviso in più segmenti (*network*), per aumentare la leggibilità.

I **blocchi organizzativi** rappresentano l'interfaccia tra il sistema operativo e il programma utente.

Richiamati dal sistema operativo comandano l'elaborazione ciclica del programma su interrupt, il comportamento del sistema di automazione all'avvio e la gestione degli errori.

Le **funzioni** contengono dei programmi che vengono eseguiti quando sono richiamate da un altro blocco di codice.

Per ogni funzione è possibile dichiarare dei parametri di ingresso, di uscita, di ingresso/uscita e delle variabili temporanee che, memorizzate in un'area opportunamente predisposta, vengono perse dopo l'elaborazione della funzione.

I **blocchi funzionali con memoria** si comportano in modo simile alle funzioni per quanto riguarda il codice che viene eseguito solo se il blocco è richiamato da un altro blocco di codice.

Si differenziano dalle funzioni in quanto possono essere memorizzati dei dati che non vanno persi una volta completata l'elaborazione del blocco stesso.

Anche in questo caso possono essere utilizzate variabili temporanee.

I **blocchi funzionali di sistema** e le **funzioni di sistema** sono blocchi integrati nelle CPU della serie S7.

Questi blocchi sono simili alle funzioni e ai blocchi funzionali con memoria ma realizzano delle funzioni predefinite come ad esempio la copia di un area di memoria in un'altra, l'impostazione dell'orologio di sistema, l'invio di un pacchetto di dati ad un nodo di una rete progettata.

Blocchi di dati

I blocchi di dati vengono memorizzati in un'apposita area di memoria.

In questo caso l'utente può definire all'interno di ogni blocco delle strutture dati come gli array e i record che nell'ambiente di STEP 7 vengono chiamati *struct*.

A differenza dei blocchi di codice non contengono istruzioni ma servono alla memorizzazione dei dati utente.

Esistono due tipi di blocchi di dati rispettivamente denominati:

- *DB globali*;
- *DB d'istanza*.

Le *DB* contengono dati che possono essere utilizzati da tutti i blocchi di codice.

La struttura di una *DB globale* viene determinata dall'utente e non dipende da nessun blocco di codice.

La struttura delle *DB d'istanza* rispecchia la dichiarazione delle variabili del blocco funzionale con memoria a cui fa riferimento.

Quando si apre un blocco di dati viene richiamato l'editor KOP, AWL e FUP e viene visualizzata la struttura del blocco stesso.

Un esempio di struttura di un blocco di dati viene riprodotta in figura 10.

Address	Variable Name	Data Type	Initial Value	Comment
+2.3	AreaD_InAnomalia	BOOL	FALSE	
+2.4	AreaE_InAnomalia	BOOL	FALSE	
+2.5	AreaF_InAnomalia	BOOL	FALSE	
+2.6	AreaG_InAnomalia	BOOL	FALSE	
+2.7	spare25	BOOL	FALSE	
+3.0	spare26	BOOL	FALSE	
+3.1	spare27	BOOL	FALSE	

Figura 10 Struttura di un blocco di dati

Nella prima colonna, per ogni variabile, è riportato l'indirizzo interno al blocco di dati.

Nelle altre colonne si devono inserire, rispettivamente, il nome del dato, il tipo, il valore iniziale e un commento.

L'utente ha la possibilità di definire dei tipi di dati utilizzando i blocchi *UDT* che si possono inserire uno di selezionando nell'ordine *Insert*, *S7 Block* e *Data Type*.

Una volta aperto, un blocco *UDT* è uguale a un blocco *DB*.

Una volta definito un tipo di dati con un blocco *UDT* può essere utilizzato all'interno di un blocco *DB* o all'interno della sezione di dichiarazione variabili di un blocco di codice.

Linguaggi di programmazione

I blocchi di codice possono contenere istruzioni nei diversi linguaggi di programmazione che il software di base di STEP 7 mette a disposizione (*KOP*, *FUP*, *AWL*, *SCL*).

Qualsiasi sia il linguaggio utilizzato all'interno di un blocco di codice, è sempre possibile richiamare, al suo interno, altri blocchi scritti in un linguaggio diverso.

Con STEP 7 è possibile la conversione del codice scritto da un linguaggio ad un altro semplicemente selezionando il linguaggio di visualizzazione nel menù *View*.

Il procedimento di conversione è sempre possibile se si vuole passare da *KOP* ad *AWL* o *FUP* oppure da *FUP* ad *AWL* o *KOP*; passare da *AWL* a *KOP* o *FUP*, invece, non è sempre possibile in quanto l'editor riesce nell'operazione solo per segmenti di codice semplici.

Tra i linguaggi di programmazione *S7*, l'*AWL* rappresenta quello più vicino al linguaggio macchina *MC7* delle CPU *S7*.

Per questa ragione il codice sviluppato in *AWL* è ottimale sia per quanto riguarda l'occupazione di memoria sia per quanto riguarda il tempo di esecuzione.

Sebbene l'editor *AWL* consenta di inserire vari tipi di commento nel codice, come tutti i linguaggi di tipo assemblativo la leggibilità dei programmi risulta difficoltosa.

Con l'editor *AWL*, *KOP* e *FUP*, selezionando *File_Generate_Source* è possibile generare i sorgenti dei blocchi scritti in uno di questi tre linguaggi.

Qualsiasi sia il linguaggio di partenza il sorgente che si ottiene è sempre un *AWL*.

Un sorgente è un documento in formato testo che può essere salvato in un file esterno al progetto selezionando *Edit_Export_Source*.

È possibile creare una libreria di funzioni da riutilizzare in progetti diversi.

Per poter inserire un sorgente esterno all'interno di un progetto si deve aprire la cartella *Sources* del programma *S7* e selezionare *Inser_External_Source*.

Una volta importato il blocco sorgente deve essere compilato per ottenere il blocco eseguibile, selezionando *Edit Compile* in SIMATIC Manager.

Linguaggio SCL

Il **linguaggio SCL**, di tipo testuale, viene definito di alto livello; è adatto per la programmazione dei sistemi di controllo *S7*.

Le espressioni linguistiche e la sintassi dell'*SCL* sono simili a quelle del linguaggio Pascal.

Con questo linguaggio si possono creare blocchi di diverse tipologie (OB, FC, FB, DB e UDT).

Per poter inserire in un programma un sorgente SCL, bisogna aprire la cartella *Sources* in SIMATIC Manager e selezionare di seguito *Insert_S7 Software_5 SCL Source*.

Con la conseguente comparsa dell'icona corrispondente al blocco inserito si può accedere all'editor SCL aprendo il blocco medesimo.

Una volta definito il blocco deve essere compilato per poter generare il *blocco eseguibile*.

Nella maggior parte dei casi, quando si compila un blocco SCL, il codice eseguibile ottenuto occuperà uno spazio maggiore di memoria (fino a dieci volte più grande) rispetto ad un blocco AWL che realizza la stessa funzione.

A volte, per codici molto complessi, questa differenza può raggiungere anche un fattore dieci.

Quando si compila un sorgente SCL per non incorrere in errori bisogna assicurarsi di aver già compilato tutti i blocchi richiamati al suo interno.

In figura 11 viene riportato un esempio di blocco funzione nel linguaggio SCL.

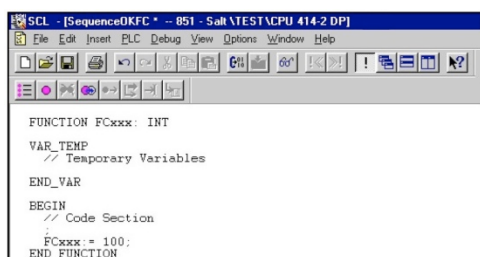


Figura 11 Esempio di blocco funzione nel linguaggio SCL

Simulazione

Il simulatore di controllori S7-PLCSIM permette di testare i programmi scritti anche quando non è disponibile un controllore reale da collegare al PC.

È possibile avviare S7-PLCSIM dalla cartella *Simatic_STEP 7* del menù avvio.

Una volta avviata l'applicazione si ha virtualmente a disposizione un PLC Siemens come quello illustrato in figura 12.

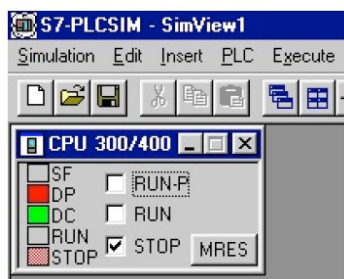


Figura 12 Aspetto del S7-PLCSIM

Il codice, per poter essere testato, deve essere prima caricato nel controllore selezionando la cartella *Blocks* e l'opzione *PLC Download*.

Selezionando RUN per dare inizio alla simulazione vengono visualizzati i blocchi di programma S7 in *modalità on-line*.

Con questa modalità vengono visualizzati i valori attuali dei parametri all'interno dei blocchi di dati.

Per i blocchi di codice la modalità on-line permette di visualizzare il valore contenuto nei registri interni del processore in corrispondenza di ogni istruzione eseguita.

Premendo il tasto destro nella finestra di destra compare un menù a tendina che consente di aggiungere o rimuovere i registri di cui si vuole visualizzare il contenuto; è anche possibile scegliere la rappresentazione dei dati (binaria, esadecimale, decimale).

Il simulatore mette inoltre a disposizione delle finestre attraverso le quali è possibile visualizzare il valore dei parametri di un blocco di dati, dei temporizzatori, dei contatori, dei registri interni.

Per inserire queste finestre bisogna utilizzare il menù *Insert*.

Un ulteriore strumento utile per la prova ed il debug dei programmi sono le *tabelle di variabili (VAT)* che si richiamano selezionando *Insert_S7 Block_6 Variable table* da *SIMATIC Manager*.