Dipartimento di Elettronica e Informazione
POLITECNICO DI MILANO

Automazione industriale
dispense del corso (a.a. 2008/2009)
A. Introduzione a ISaGRAF

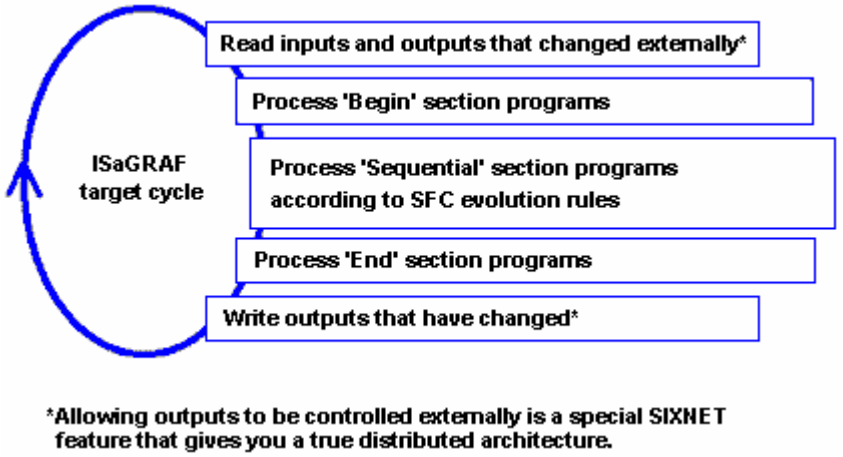
Luigi Piroddi
piroddi@elet.polimi.it

Introduzione

ISaGRAF è un ambiente di sviluppo per sistemi di controllo logico, basato su linguaggi standard di programmazione di PLC.

Un progetto ISaGRAF è una collezione di programmi, sotto-programmi e funzioni che formano un’applicazione di controllo completa (eseguibile su un controllore target). Ogni programma controlla una parte specifica dell’applicazione.

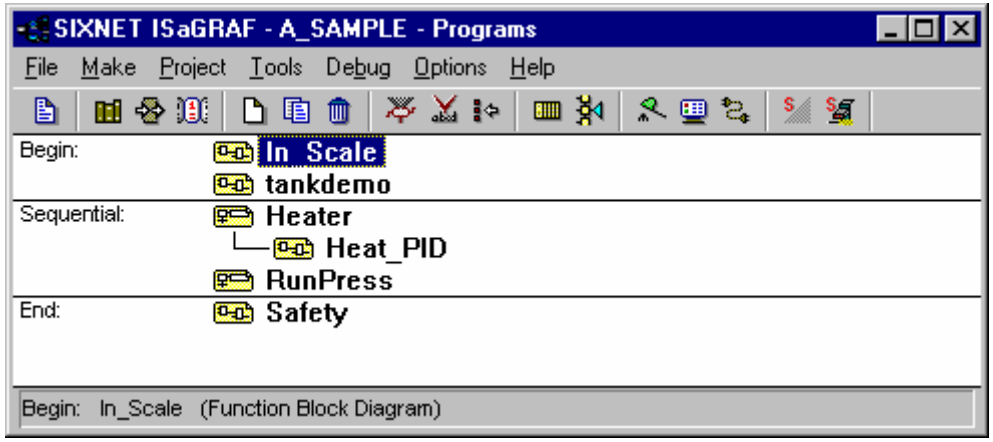
Programmi e funzioni sono suddivisi in 4 sezioni a seconda della loro posizione nell’ISaGRAF Target Cycle.



Le 4 sezioni sono:

- *Iniziale*
  Programmi ciclici che non dipendono dal tempo. Vengono eseguiti sistematicamente all’inizio del ciclo dopo la lettura degli ingressi. Tali programmi non possono essere scritti in SFC.
- *Sequenziale*
  I programmi di questa sezione sono scritti in SFC e supportano parallelismo e gerarchia.
- *Finale*
  Simili ai programmi della sezione iniziale, vengono eseguiti alla fine del ciclo prima della scrittura delle uscite.
- *Funzioni / blocchi funzione*
  Le funzioni sono sotto-programmi richiamabili da programmi nelle altre tre sezioni. Non possono essere scritte in SFC.

Ogni sezione può contenere più di un programma o anche nessuno. Le sezioni compaiono nella finestra Program Management separate da una barra orizzontale.



ISaGRAF implementa tutti e 5 i linguaggi IEC 1131-3, Function Block Diagram/Ladder Diagram (FBD/LD), Quick Ladder (Quick LD), versione ridotta del Ladder Diagram, Sequential Function Charts (SFC), Structured Text (ST), Instruction List (IL).

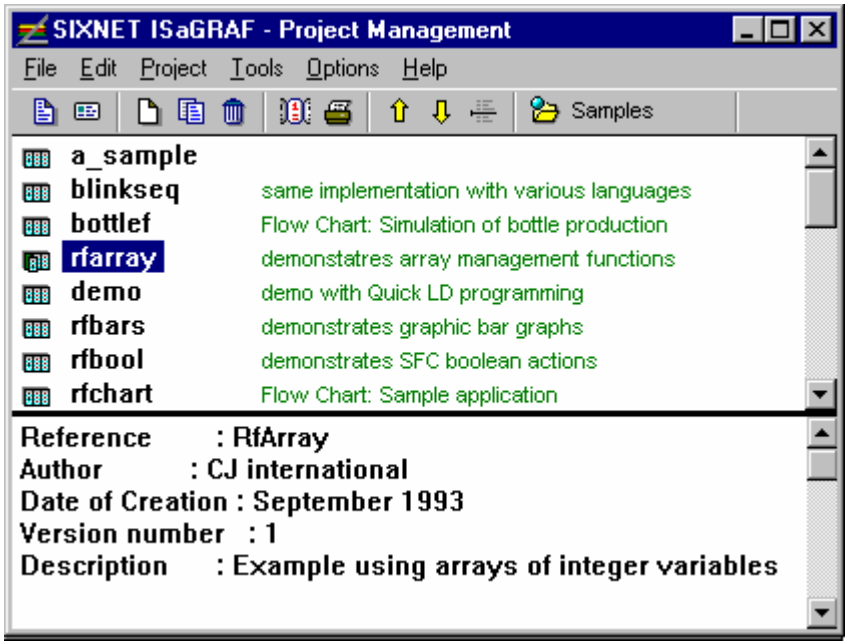
Tutti e 5 i linguaggi possono essere utilizzati nella stessa applicazione.

Gestione progetti

Le icone principali di ISaGRAF sono le seguenti:

Il programma principale è la Gestione Progetti a cui si accede cliccando sull’icona Progetti.

La sezione superiore contiene la lista dei progetti, quella inferiore una descrizione del progetto selezionato.



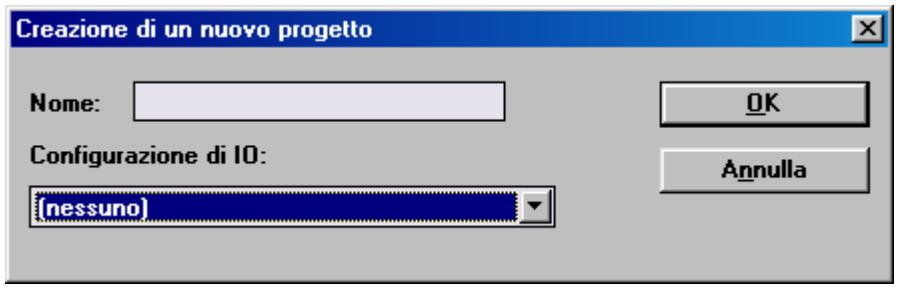
Di default, ISaGRAF conserva i progetti nella directory \isawin\smp sul medesimo disco su cui è installato il programma principale.

E’ possibile creare una directory progetti diversa selezionando File > Selezione gruppo progetti.

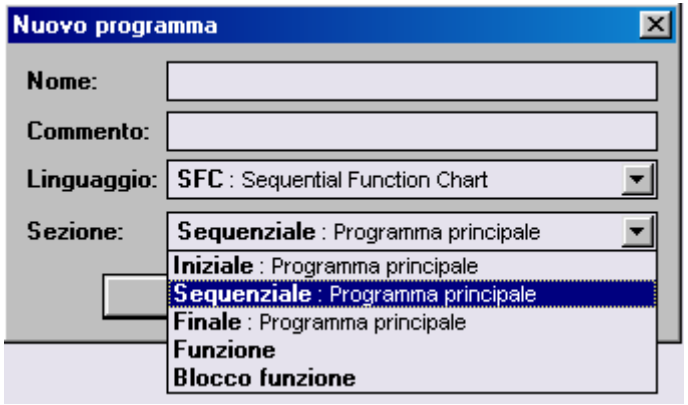
Creazione di un progetto

Per creare un progetto si devono operare i seguenti passi:

1. Aprire la finestra di Gestione Progetti.
2. Selezionare File > Selezione gruppo progetti
3. Selezionare Default c:\isawin\apl.
4. Dalla finestra di Gestione Progetti selezionare File > Nuovo.
5. Assegnare un nome al programma (massimo 8 caratteri, di cui il primo deve essere una lettera).



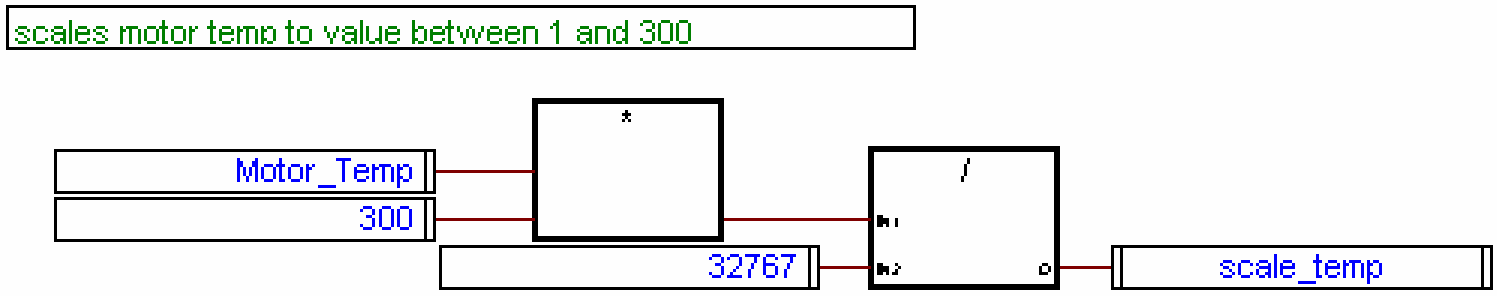
Cliccando sull’icona associata ad un progetto si apre la finestra Programmi, nella quale si possono creare i vari moduli e programmi che costituiscono il progetto, usando i 5 linguaggi disponibili. Per ogni modulo occorre specificare a quale sezione appartiene e il linguaggio con cui lo si intende sviluppare.



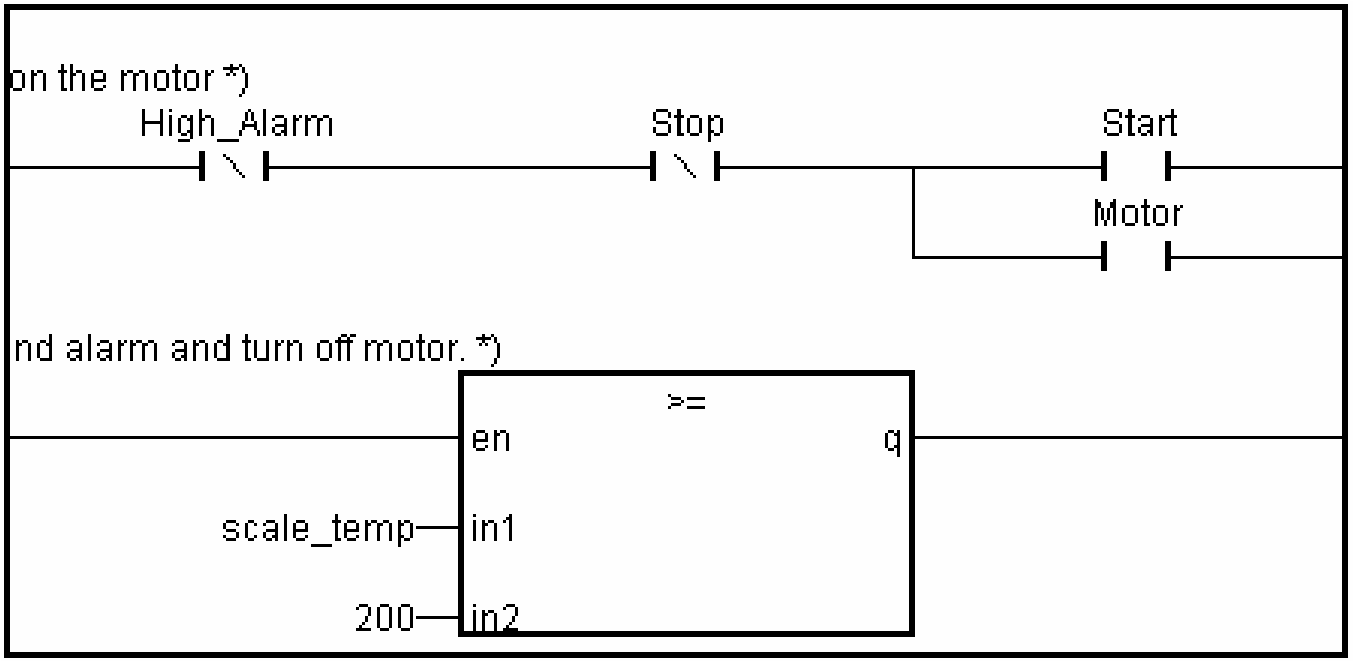
Cliccando sull’icona di un modulo si apre una finestra di editing del programma nel linguaggio IEC 1131-3 prescelto.

Le variabili utilizzate all’interno di ogni programma vanno definite nel dizionario.

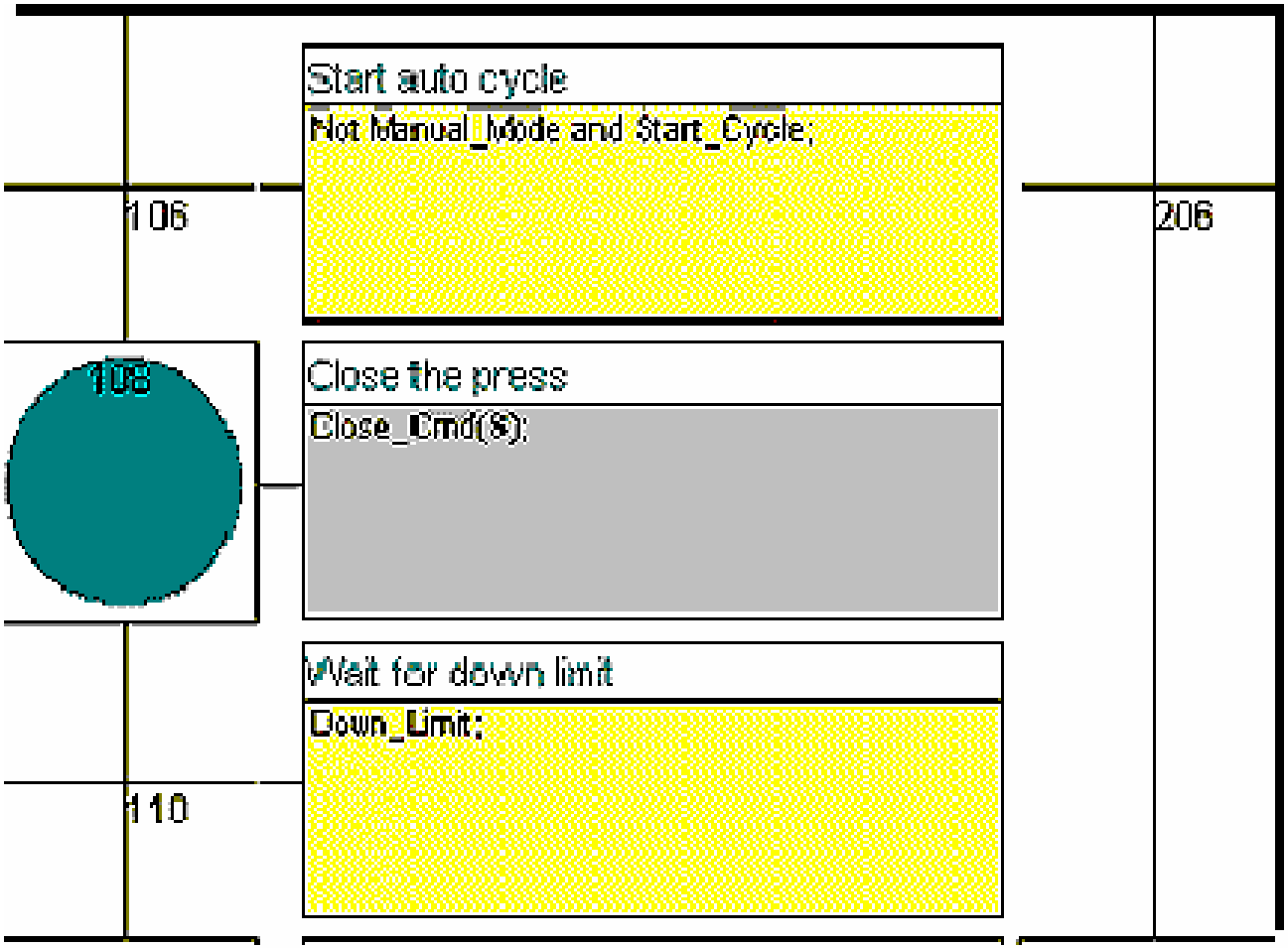
Editor di programma: Function Block Diagram/Ladder Diagram (FBD/LD)



Editor di programma: Quick Ladder (Quick LD)



Editor di programma: Sequential Function Charts (SFC)



Si noti che le azioni nei passi possono essere dettagliate usando gli altri linguaggi (ST, IL, LD, e FBD).

Editor di programma: Structured Text (ST)

Il testo strutturato è un linguaggio di programmazione simile al Pascal. Esso fornisce un modo efficiente per programmare operazioni complesse. Si usa spesso per creare blocchi funzione definiti dall’utente, e per dettagliare passi e transizioni in schemi SFC.

```
(* signal: the sequence is changed *)
ResetOrder := TRUE;

(* search for digits in the input message *)
nbchr := mlen (command);
NbSeq := 0;
for chr := 1 to nbchr do
  code := ascii (command, chr);
  (* allowed numbers = ['1' .. '5'] - Ascii codes = [49 .. 53] *)
  if (nbseq < MAX_SEQ) & (code >= 49) & (code <= 53) then
    rc := ArWrite (ARIDT, NbSeq, code-48);
    NbSeq := NbSeq + 1;
  end_if;
end_for;
```

Editor di programma: Instruction List (IL)

```
(* Calling program : converts an analog value into a time value *)

Main:       LD          bi0
            SUBPRO      bi1,bi2     (* call sub-program to get analog value *)
            ST          result      (* result := value returned by sub-program *)
            GT          vmax        (* test value overflow *)
            RETC                    (* return if overflow *)
            LD          result
            MUL         1000        (* converts seconds in milliseconds *)
            TMR                     (* converts to a timer *)
            ST          tmval       (* stores converted value in a timer *)

(* Called sub-program named 'SUBPRO' : evaluates the analog value *)
(* given as a binary value on three boolean inputs: in0, in1, in2 are the three boole
input parameters of the sub-program *)
            LD      in2
            ANA                 (* result = ana (in2); *)
            MUL     2           (* result := 2*ana (in2); *)
```

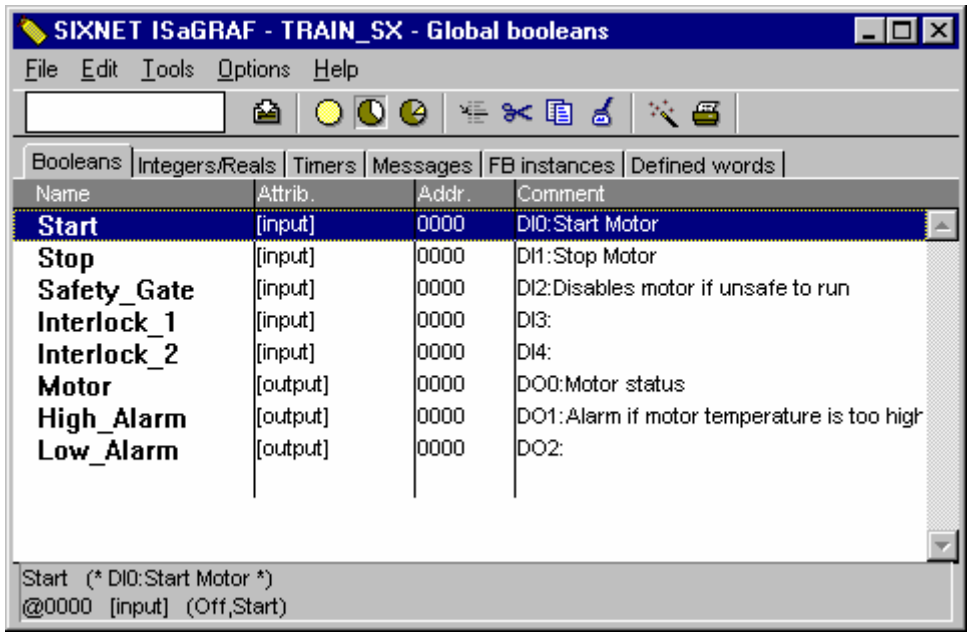
Il linguaggio IL è simile ad un linguaggio proprietario della Siemens.

Dizionario

Il dizionario è la collezione di tutte le variabili interne, di ingresso o di uscita usate nei programmi di un progetto.

Le variabili di ingresso e uscita sono associate a variabili del controllore o a ingressi e uscite virtuali.

È possibile definire delle costanti scegliendo come attributo Costante (ed in tal caso occorre specificarne il valore).



Costanti e variabili possono essere create nel dizionario durante il processo di progettazione e sono specificate:

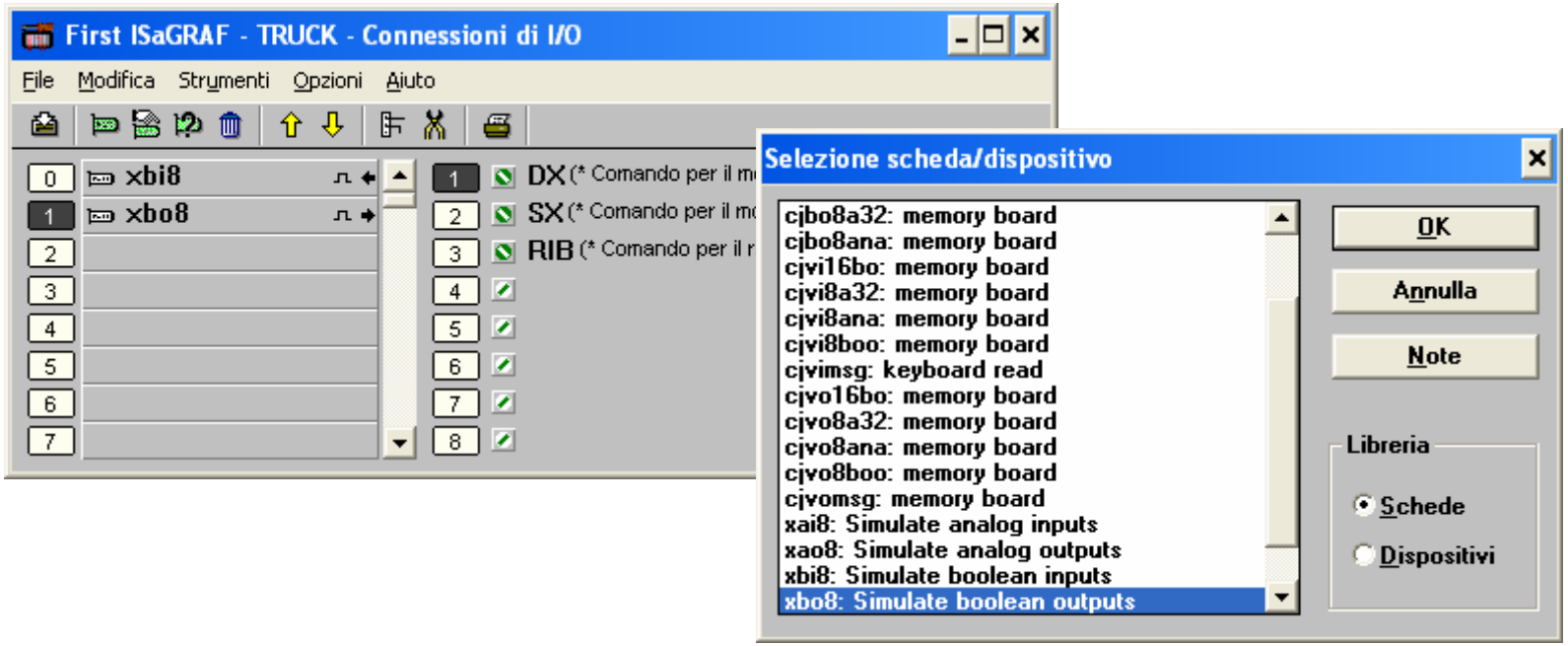
- *locali* quando sono specifiche di un programma
- *globali* quando possono essere usate da qualunque programma all’interno di un progetto
- *comuni* quando possono essere usate da qualunque progetto

Connessioni di I/O

Tutte le variabili definite nel dizionario come variabili di ingresso e uscita devono poi essere mappate su dei dispositivi fisici oppure su delle “schede virtuali”.

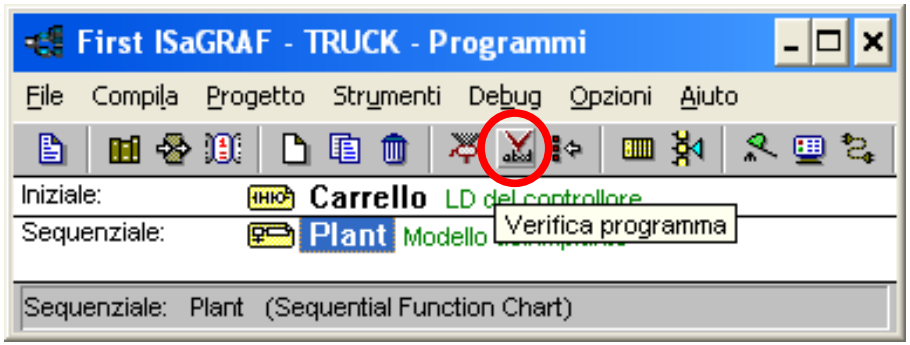
La finestra per le connessioni si apre selezionando Progetto > Connessioni di I/O.

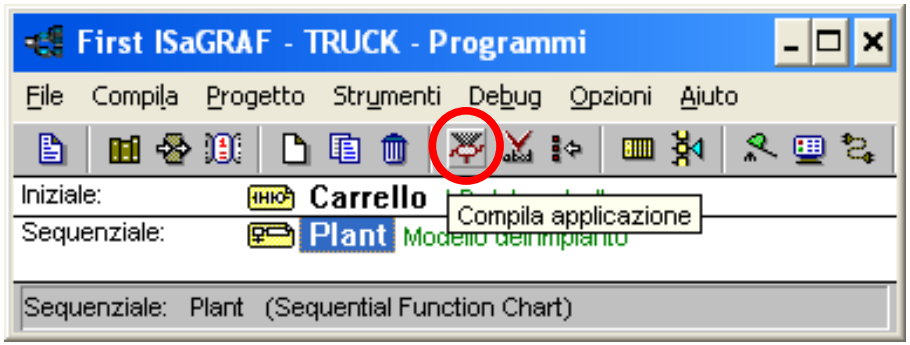
È necessario inserire delle schede di ingresso (digitale o analogico) su cui mappare gli ingressi, e delle schede di uscita (digitale o analogico) su cui mappare le uscite.



Verifica, compilazione, debugging e simulazione

Prima di essere eseguito, il codice deve essere verificato e compilato.



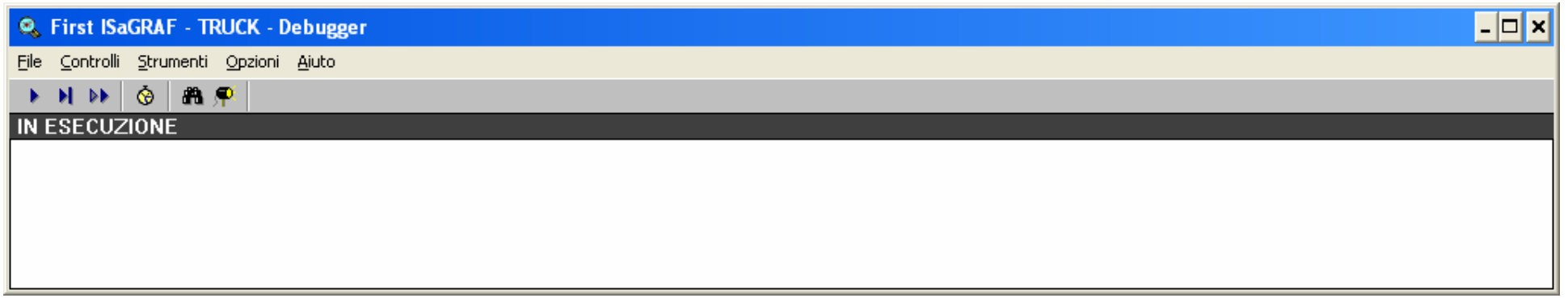


Una volta compilato il codice è possibile eseguire una simulazione oppure un debug del programma utilizzando il comando Debug > Simulazione.

Di seguito sono brevemente descritte le principali finestre utilizzate durante la simulazione.

Debugger

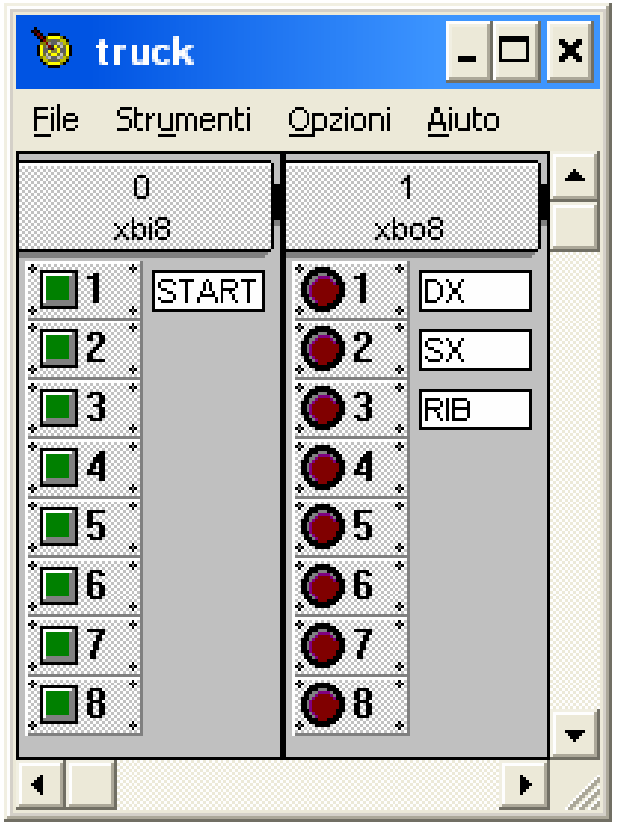
Finestra principale utilizzata per commutare tra la modalità di esecuzione in tempo reale e quella passo passo (utile in fase di debug).



Attraverso questa finestra è poi possibile accedere alle finestre secondarie.

Pannello di controllo

Rappresenta le variabili mappate sulle schede virtuali di ingresso ed uscita e viene utilizzata per controllare lo stato degli ingressi e per visualizzare lo stato delle uscite.



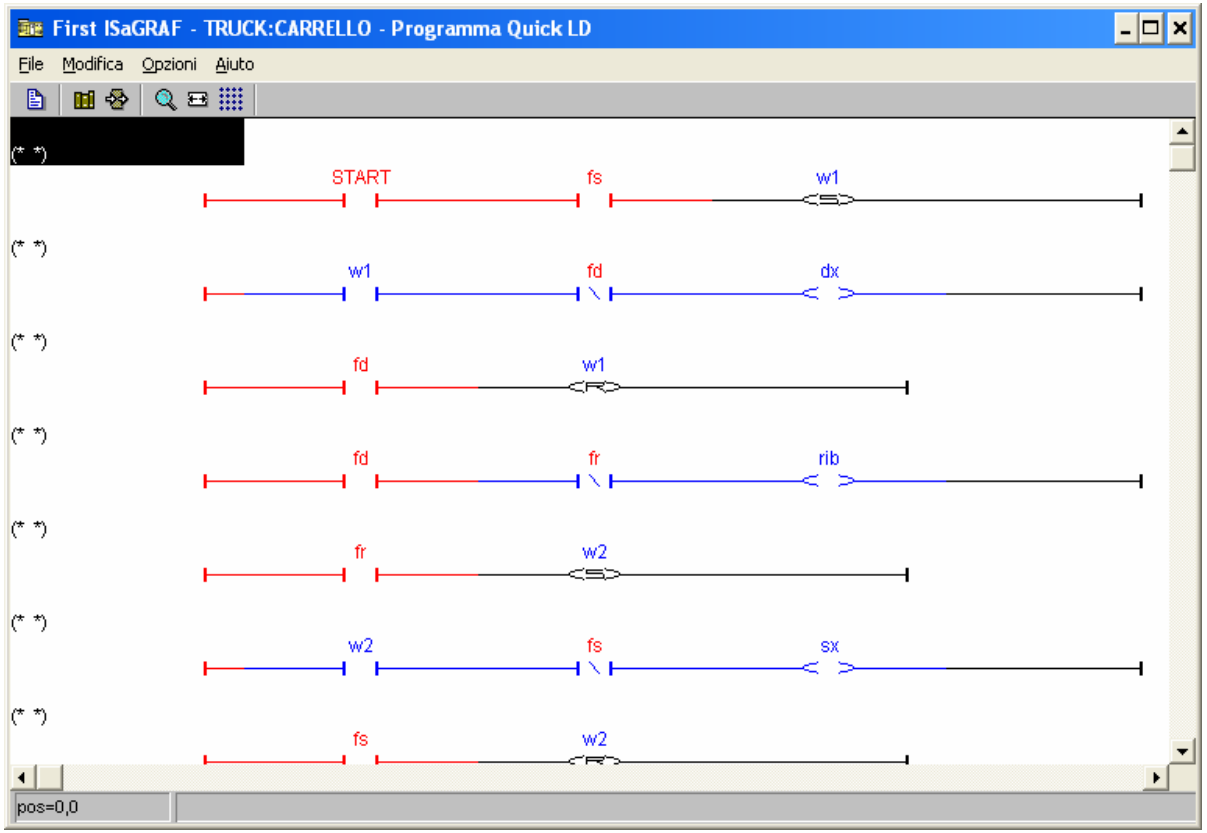
I pulsanti verdi rappresentano gli ingressi del PLC (uscite dell’impianto controllato), mentre le spie rosse rappresentano le uscite del PLC (ingressi dell’impianto controllato).

Lista di variabili

Questa finestra rappresenta una lista delle variabili “spiate” e si apre dal menù Strumenti > Lista variabili spiate. Essa permette di visualizzare il valore delle variabili interne o di modificarne il valore.

Accesso all’editor in simulazione

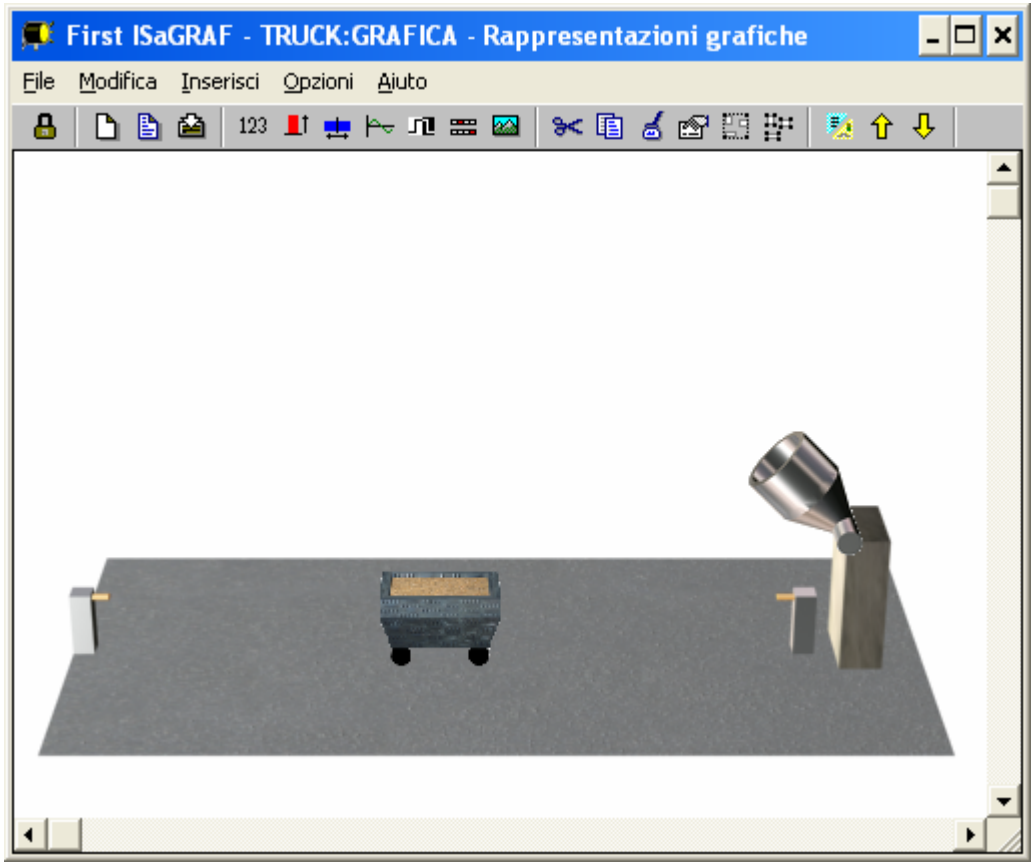
Durante la fase di simulazione è possibile aprire le finestre del codice in modo da seguirne l’esecuzione. Nei programmi LD viene evidenziato lo stato delle variabili (rosso = vero, blu = falso), dei contatti e delle bobine (rosso = aperto, blu = chiuso). Nei programmi SFC vengono invece evidenziati i passi attivi.



Rappresentazioni grafiche

È infine possibile creare una rappresentazione grafica dell’impianto.

La finestra della grafica si apre dalla finestra del Debugger attraverso il menù Strumenti > Rappresentazioni grafiche.



Per la costruzione di una rappresentazione grafica si possono utilizzare dei semplici oggetti, come etichette di testo, barre unipolari o bipolari, oppure icone booleane.