1 Funzioni ed FB

Funzioni

Le funzioni hanno numero di variabili in ingresso variabile e sempre una sola variabile in uscita. Per utilizzarle basta inserirle nei programmi LD ed FBD e connetterle alle variabili. Nei programmi IL devono essere chiamate con l'istruzione CAL, nei programmi ST basta indicarne il nome per essere eseguite.

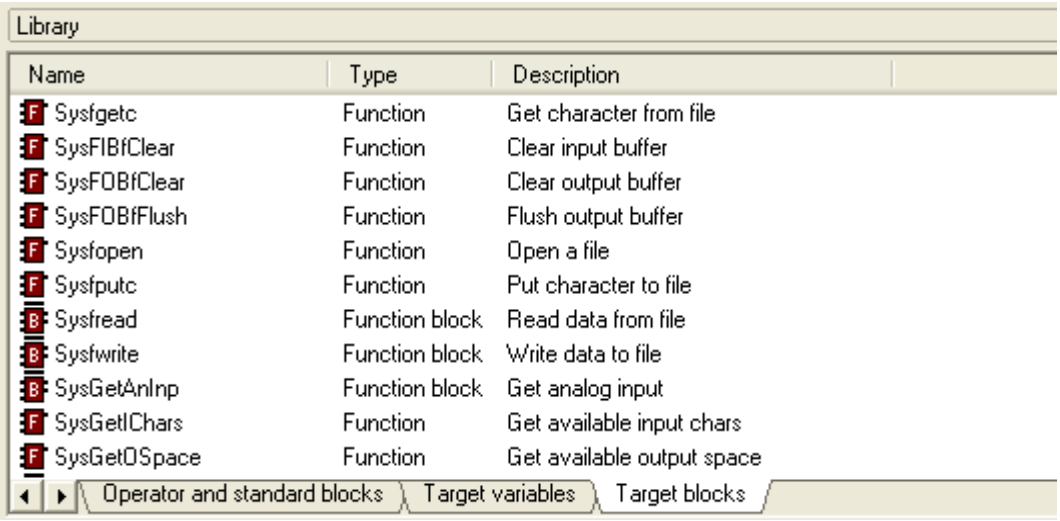
Function Blocks

Le FB a differenza delle funzioni allocano nel programma una variabile che contiene tutte le variabili di input e di output gestite dal blocco funzione. Per utilizzarle basta inserirle nei programmi LD ed FBD e connetterle alle variabili. Nei programmi IL devono essere chiamate con l'istruzione CAL, nei programmi ST basta indicarne il nome per essere eseguite.

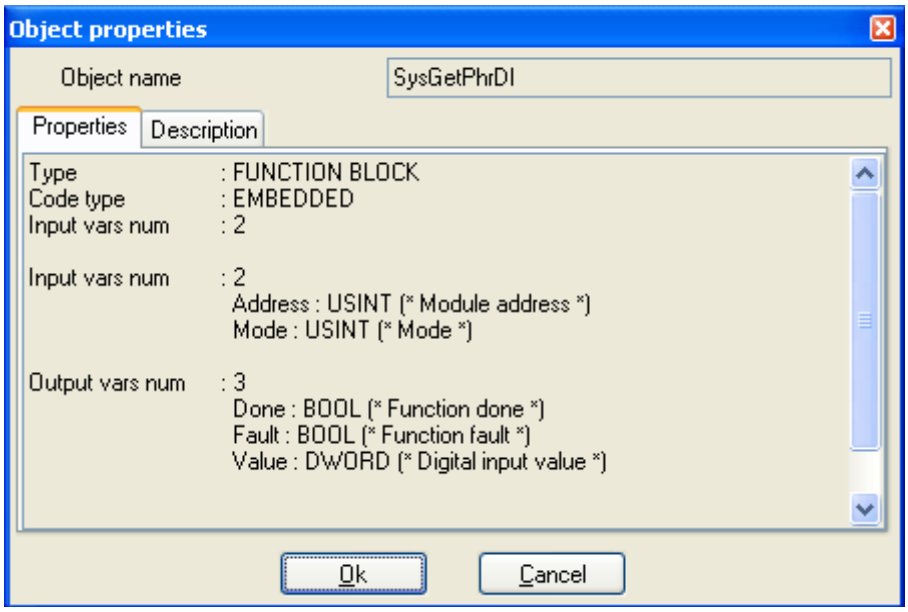
1.0.1 Funzioni ed FB embedded

Con l'ambiente di sviluppo LogicLab vengono fornite funzioni e blocchi funzione (FB) embedded che permettono di accedere alle risorse hardware e software del sistema ***Slim line***. Le funzioni e le FB embedded sono visualizzate da LogicLab nella finestra delle librerie.

Se la finestra non è visualizzata, occorre abilitarne la visualizzazione dalla voce di menù ***View*** → ***Tool windows*** → ***Library***. Attivando il Tab ***Target blocks*** verrà visualizzato un elenco con tutte le funzioni (Indicate con ***F***) ed i blocchi funzione (Indicati con ***B***) embedded.



Agendo con il tasto destro del mouse su ogni singola funzione o blocco funzione è possibile visualizzarne la finestra delle proprietà in cui sono indicate le variabili in ingresso ed il ritorno delle funzioni, mentre per i blocchi funzioni sono indicate le variabili in ingresso ed in uscita, cosi come nella figura a lato.



1.0.2 Librerie

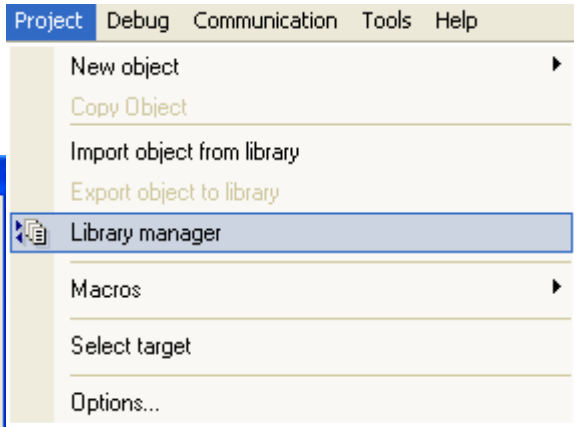
In aggiunta alle funzioni ed FB embedded sono fornite delle librerie che contengono una serie di funzioni e blocchi funzioni che possono essere utilizzati nel proprio programma. Le librerie fornite con LogicLab si trovano nella directory di installazione programma ***Programmi\Elsist\IEC61131Suite\LogicLab2p0\Libraries***, ma è possibile anche utilizzare librerie fornite successivamente o di cui si è eseguito il download dal sito. Esistono due possibilità per utilizzare le librerie:

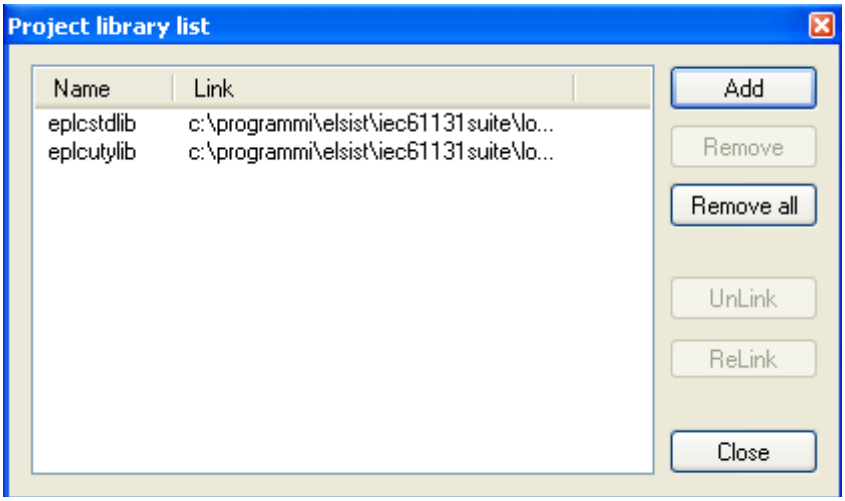
Import libreria: In questo modo vengono importati nel proprio programma tutti gli oggetti presenti nella libreria, gli oggetti possono così essere utilizzati nel programma. Questa è una buona soluzione. La controindicazione è quella di aumentare la dimensione del file di progetto LogicLab (*.ppjs), in quanto deve contenere oltre al proprio programma anche tutti gli oggetti della libreria importata. ***Il programma eseguibile generato conterrà comunque solo gli oggetti utilizzati***.

Import oggetti: In questo modo è possibile importare da una libreria solo gli oggetti (Funzioni, FB, ecc) che interessano, i quali diverranno parte integrante del proprio progetto.

1.0.3 Import libreria

Con questa modalità vengono importati tutti gli oggetti presenti nella libreria. Per importare nel proprio programma l'intera libreria dal menù selezionare la voce ***Project*** → ***Library manager*** si aprirà una finestra come quella sotto riportata.

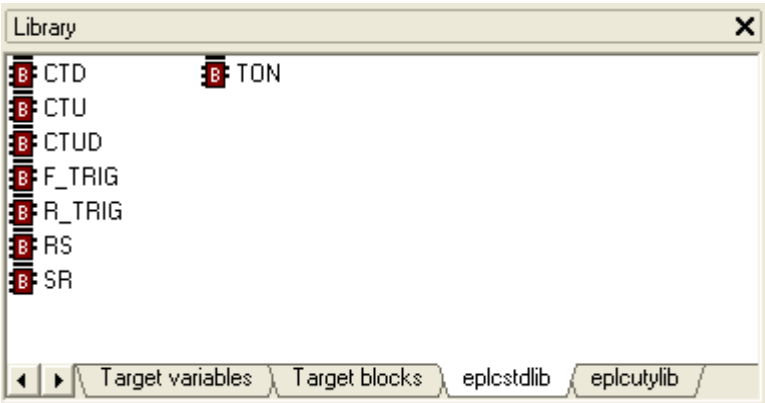




Agendo sul tasto ***Add*** si aprirà una finestra di browser del disco. Scegliere la directory dove si trova la libreria, e selezionare i files di libreria da importare.

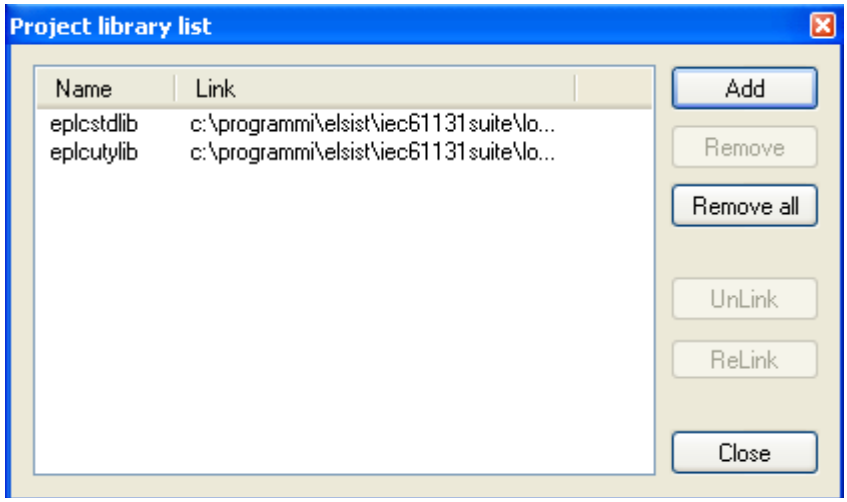
Agendo sul tasto ***Close***, nella finestra ***Library*** di LogicLab (***Ctrl-L***) verranno visualizzati dei tabs aggiuntivi, uno per ogni libreria importata.

Basta trascinare l'oggetto desiderato nel proprio progetto per poterlo utilizzare.



1.0.4 Link a libreria

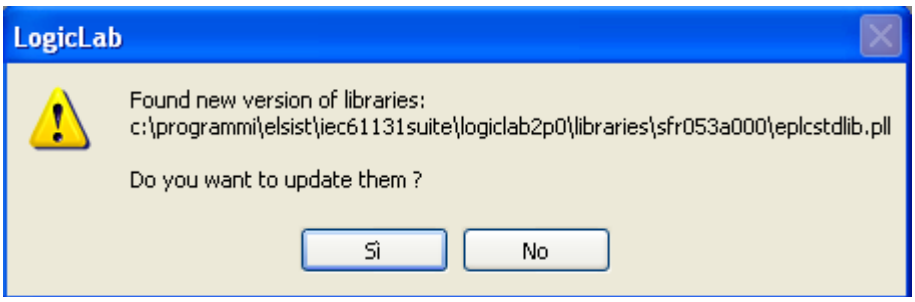
Eseguendo l'import delle librerie nel proprio progetto come indicato al capitolo precedente tutti gli oggetti della libreria importata vengono trasferiti nel proprio file di progetto (*.ppjs), ma viene comunque mantenuto un link alla libreria di origine come si vede dalla finestra a lato.



Questo permette nel caso la libreria sorgente venga modificata con una versione più recente di effettuare l'aggiornamento automatico della nuova libreria nel proprio progetto.

Se la libreria sorgente non è più presente oppure è stata spostata dalla posizione da dove è stata importata, LogicLab non eseguirà più il controllo senza segnalare errori.

Tramite il menù ***Project → Library manager*** che apre la finestra a lato, come si vede, è possibile selezionare le varie librerie e con il tasto ***UnLink*** rimuovere il link oppure con il tasto ***ReLink*** eseguire un link alla nuova posizione dove si trova la libreria.

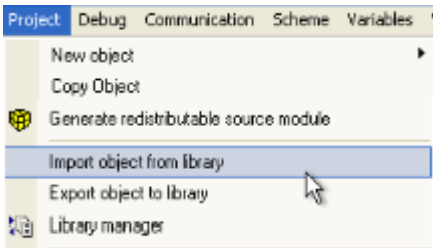


Aprendo il progetto, LogicLab controlla tutte le librerie importate e nel caso una o più librerie sorgenti siano più recenti delle versioni importate viene visualizzato un messaggio di avvertimento che chiede conferma se eseguire oppure no l'aggiornamento delle librerie.

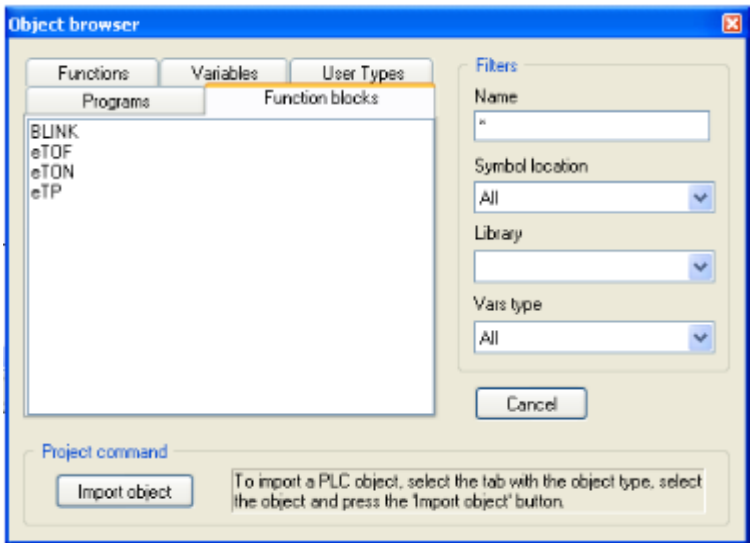
Eseguendo l'aggiornamento tutti gli oggetti della libreria importata presenti nel proprio progetto vengono sovrascritti con gli oggetti presenti nella libreria sorgente ed eventuali nuovi oggetti sono automaticamente importati.

1.0.5 Importazione oggetti

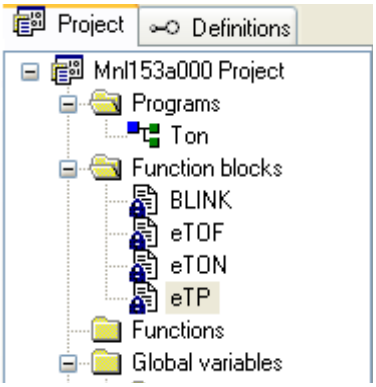
Per importare nel proprio programma oggetti dalle librerie occorre nel menù selezionare la voce ***Project*** → ***Import object from library***. Si aprirà una finestra di browser del disco. Scegliere la directory dove si trova la libreria, e selezionare il file di libreria da cui si desidera importare gli oggetti.



Si aprirà la finestra di ***Object browser*** che permetterà di visualizzare tutti gli oggetti presenti nella libreria. Selezionando i vari tabulatori presenti è possibile visualizzare tutti gli oggetti della libreria ordinati per nome.



Con un click del mouse si evidenzia l'oggetto o gli oggetti desiderati. Con il tasto ***Import object*** gli oggetti selezionati verranno inclusi nel programma.



Come si nota dalla foto a destra alcuni oggetti appaiono visualizzati con un simbolo di lucchetto, questo sta ad indicare che sono oggetti protetti, cioè non possono essere modificati. Una volta importati nel proprio programma gli oggetti resteranno inclusi nel programma stesso e sarà possibile utilizzarli su qualsiasi PC anche se non si dispone della libreria originale.

1.0.6 Considerazioni su link a libreria e su import oggetti

Come visto nei paragrafi precedenti per utilizzare funzioni e/o blocchi funzione di libreria è possibile usare due diversi metodi, importare solo l'oggetto desiderato oppure tutta la libreria nel proprio progetto.

In entrambi i casi l'oggetto verrà incluso nel proprio progetto, in questo modo si è sicuri che anche nel futuro con versioni successive di libreria sarà sempre possibile ricompilare il progetto utilizzando l'oggetto con il quale si era sviluppato e testato.

Nel caso si desideri sostituire l'oggetto con una versione più recente dello stesso si userà un diverso approccio in funzione del fatto che l'oggetto sia presente in una libreria collegata oppure sia stato importato.

Libreria collegata

Come visto precedentemente, le librerie collegate mantengono un riferimento alla libreria di origine, nel percorso di memorizzazione nella distribuzione di LogicLab le librerie sono incluse in directories il cui nome rappresenta la versione. In questo modo potranno essere distribuite versioni successive di libreria, ma il progetto alla sua riapertura farà sempre il controllo con la versione originale senza eseguire l'aggiornamento automatico.

Per effettuare l'aggiornamento di un oggetto di una libreria collegata occorre eseguire un ***ReLink*** alla nuova versione della libreria. ***Attenzione!*** Questa operazione aggiornerà tutti gli oggetti presenti nella libreria.

Oggetto importato

Nel caso di oggetto importato, per effettuare l'aggiornamento, basterà rimuovere l'oggetto attuale dal progetto ed eseguire un import dello stesso oggetto dalla nuova versione della libreria.

Conclusioni

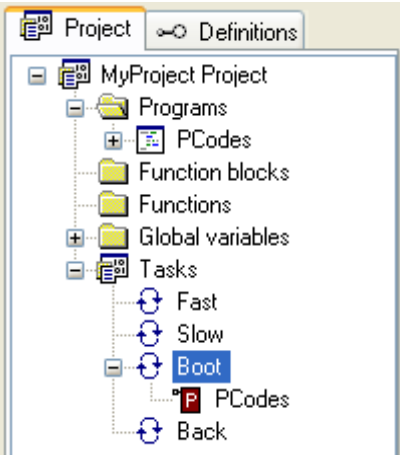
In generale si consiglia di non eseguire il collegamento della libreria ma di includere i singoli oggetti nel proprio progetto, questo permette una più semplice gestione degli aggiornamenti.

Alcune librerie contengono una serie di oggetti (Funzioni e blocchi funzione) che sono di vasto impiego, in questo caso è consigliabile sempre collegare queste librerie. Ecco l'elenco delle librerie che si consiglia di collegare al progetto:

Libreria	Codice	Descrizione
ePLCStdLib	SFR053*000	Libreria standard IEC61131, contiene funzioni e blocchi funzione definiti dalla normativa IEC61131 e non presenti nella libreria embedded del prodotto.
ePLCAuxLib	SFR058*000	Libreria ausiliaria, contiene funzioni e blocchi funzione di varia utilità.

1.0.7 Protezione funzioni e blocchi funzione

Alcune funzioni e/o blocchi funzione di libreria possono essere protetti da un codice che deve essere ordinato separatamente. Per abilitarne l'esecuzione occorre sbloccarle definendone il codice (Stringa alfabetica di 18 caratteri) con la funzione ***SysPCodeAccept***.



La funzione deve essere eseguita una sola volta passando il codice di protezione, se il codice è corretto la funzione ritorna ***TRUE*** e la relativa funzione sarà sprotetta fino al prossimo riavvio del programma. E' possibile eseguire più chiamate alla funzione una per ogni codice di protezione da definire.

Il consiglio è di inserire le varie chiamate alla funzione in un programma che verrà eseguito nella task di boot quindi prima di ogni chiamata ad altri programmi, garantendo lo sblocco delle funzioni desiderate.

A lato si può vedere come in un progetto il programma di definizione codici di protezione ***PCodes*** sia definito nella esecuzione della task di boot. Di seguito riportiamo il codice sorgente del programma ***PCodes*** realizzato in linguaggio ST. Naturalmente i codici riportati sono di fantasia pertanto se eseguito la funzione ***SysPCodeAccept*** ritornerà sempre ***FALSE***.

Definizione variabili

	Name	Type	Address	Array	Init value	Attribute	Description
1	CodesOk	BOOL	Auto	[0..2]	3(0)	..	Protection codes ok

Programma ST

```
(* Check the protection codes. *)

    CodesOk[0]:=SysPCodeAccept('abcdefghijklmnopqr'); (* Protection code ok (Function 1) *)
    CodesOk[1]:=SysPCodeAccept('rqponmlkjihgfedcba'); (* Protection code ok (Function 2) *)
    CodesOk[2]:=SysPCodeAccept('abcdefghiihgfedcba'); (* Protection code ok (Function 3) *)

(* [End of file] *)
```

Normalmente le funzioni ed i blocchi funzioni protetti possono funzionare in modo demo per un certo periodo di tempo dalla loro prima esecuzione dopo l'accensione del sistema. Terminato il tempo di prova termina il funzionamento e viene generato un errore che è rilevabile con la funzione ***SysGetLastError***.