

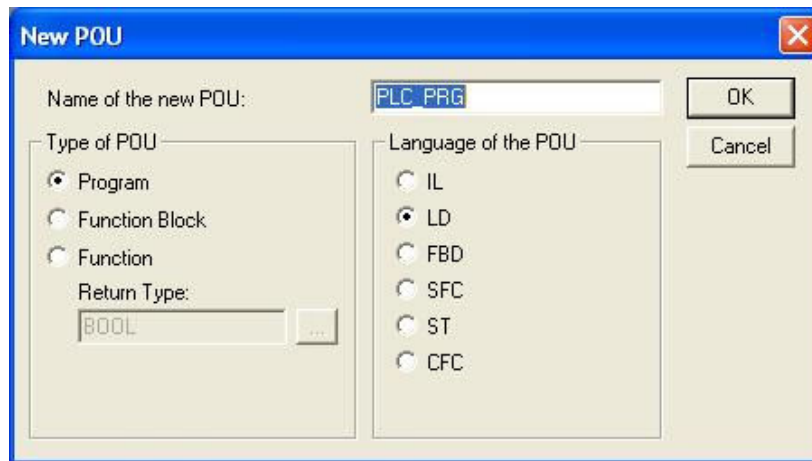
Aprire Codesys

File → New



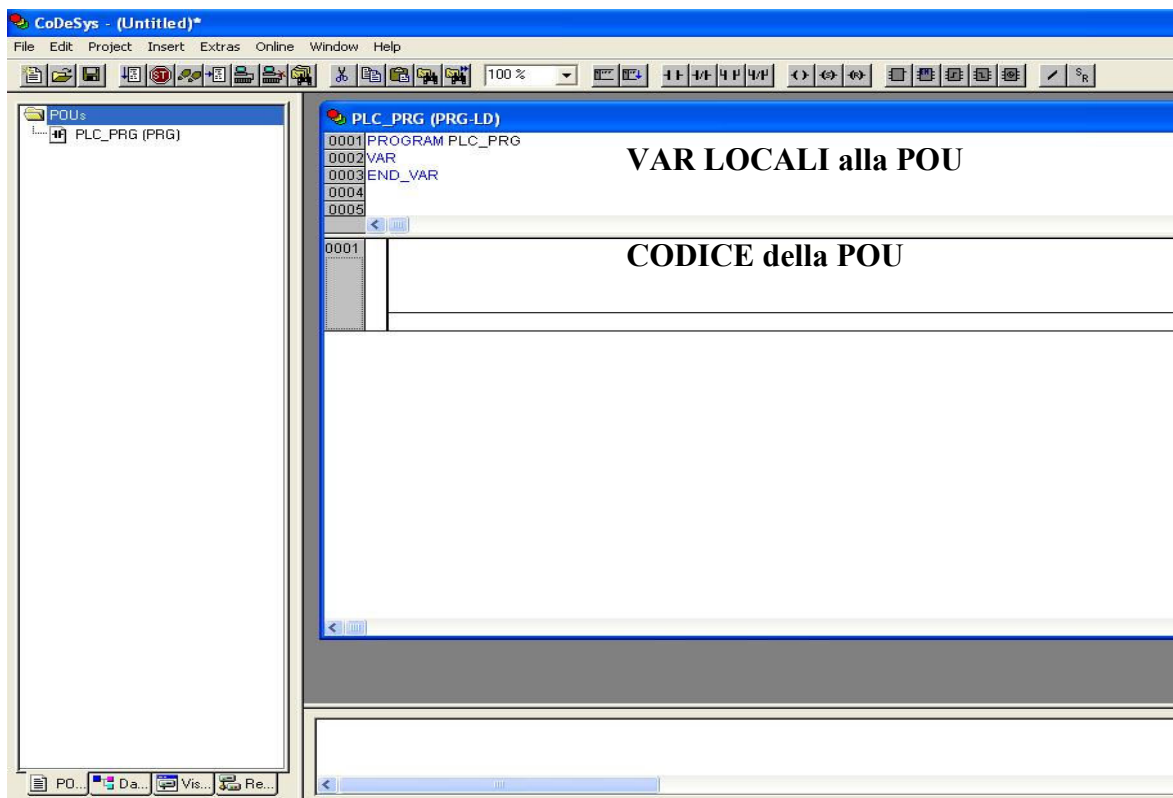
Va lasciato “None” perchè non abbiamo un effettivo PLC target.

Premendo ok →

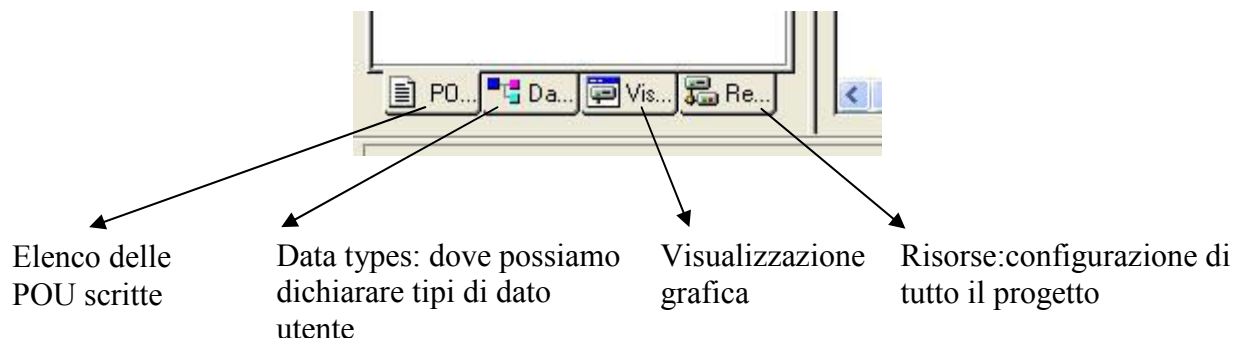


PLC\_PRG è una parola chiave: se non ci sono task configurati Codesys prende PLC\_PRG e lo manda in esecuzione in modalità freewheeling (appena finisce lo rimette in esecuzione). Se ho solo un PLC\_PRG e una visualizzazione (senza il programma ST di simulazione), non ho bisogno di configur task!!! Se invece ho una simulazione scritta in ST devo associare un task sia a PLC\_PRG che alla simulazione in ST!!!

**N.B.** PLC\_PRG deve essere un program! Codesys non può associare task e FB, ma solo a PROGRAM,. La norma dice diversamente.



Il menù in basso a sinistra della schermata sovrastante è:



Andando su Resources:



**Global variables:** dove definisco le variabili globali. Una variabile globale è vista da tutte le POU, anche senza essere dichiarata EXTERN (come vorrebbe la norma).

**Variable Configuration:** vedi help. Una variabile può essere definita in una POU senza specificarla esattamente. La definizione precisa sarà poi nella sezione variable configuration.

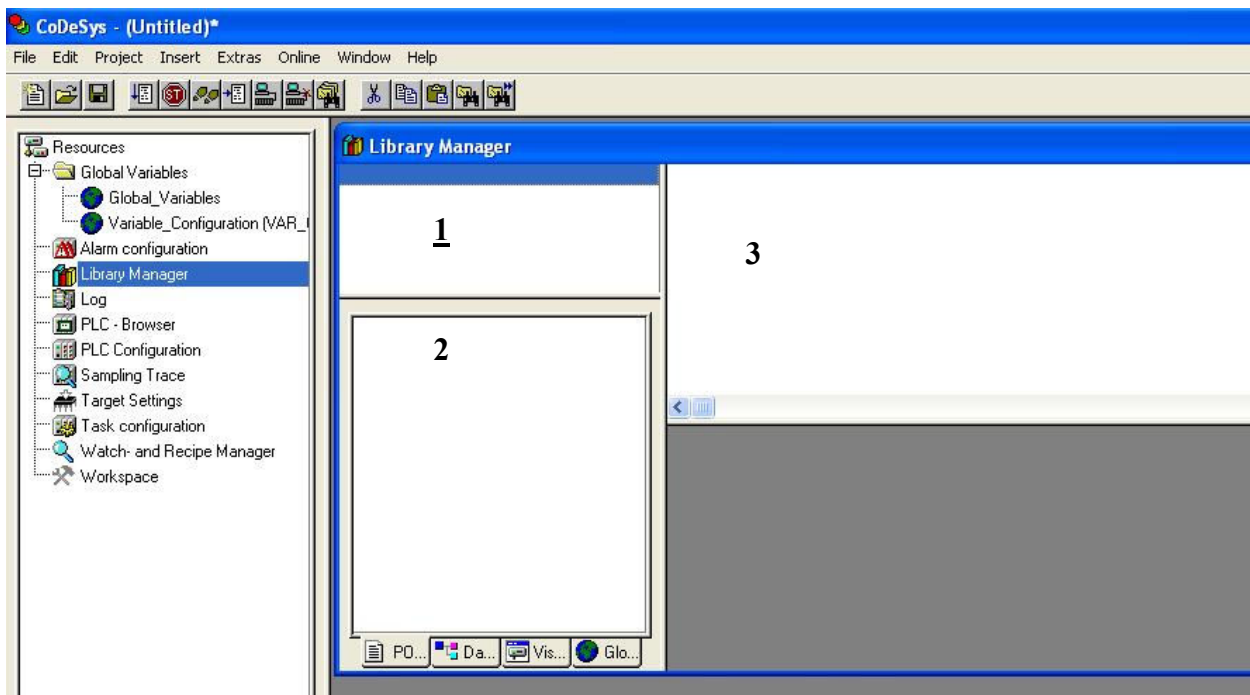
**Es.**

```
FUNCTION_BLOCK locio
VAR
  loci AT %I*: BOOL := TRUE; (*ho definito una var locale al FB che è un ingresso, ma non ho specificato quale*)
  loco AT %Q*: BOOL; (*ho definito una var locale al FB di uscita, ma non ho specificato quale uscita è*)
END_VAR

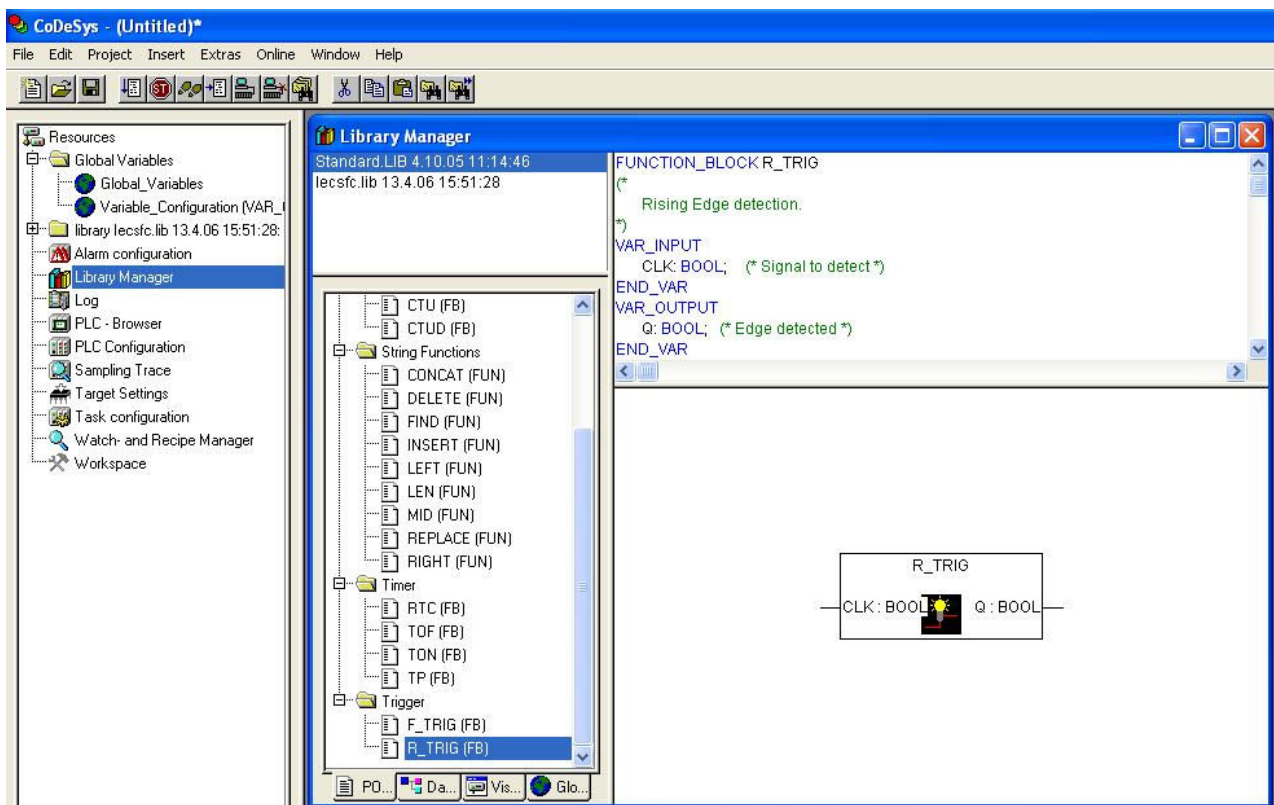
PROGRAM PLC_PRG
VAR
  Hugo: locio; (*istanzio 2 FB di tipo locio*)
  Otto: locio;
END_VAR
VAR_CONFIG
  PLC_PRG.Hugo.loci AT %IX1.0 : BOOL; (* specifico che quella var di quell'istanza di quel FB è I1.0*)
  PLC_PRG.Hugo.loco AT %QX0.0 : BOOL;
  PLC_PRG.Otto.loci AT %IX1.0 : BOOL;
  PLC_PRG.Otto.loco AT %QX0.3 : BOOL;
END_VAR
```

**Alarm Configuration:** non la usiamo

**Library Manager:** mi si apre questa finestra



Di default il progetto non ha la possibilità di inserire chiamate a FB o a functions. Per poterli chiamare ed usare devo agganciare la libreria che li contiene. Mi metto col mouse in 1, click dx → Additional Library → STANDARD.LIB. Se voglio programmare in SFC devo includere anche Iecsc.lib. A questo punto la schermata si deve modificare come segue:



**Log:** data logger per eventi (es. quando è stato avviato il PLC etc etc). **NON LO USIAMO**  
**PLC Browser:** è un'interfaccia comandi per il PLC virtuale. **NON LO USIAMO**

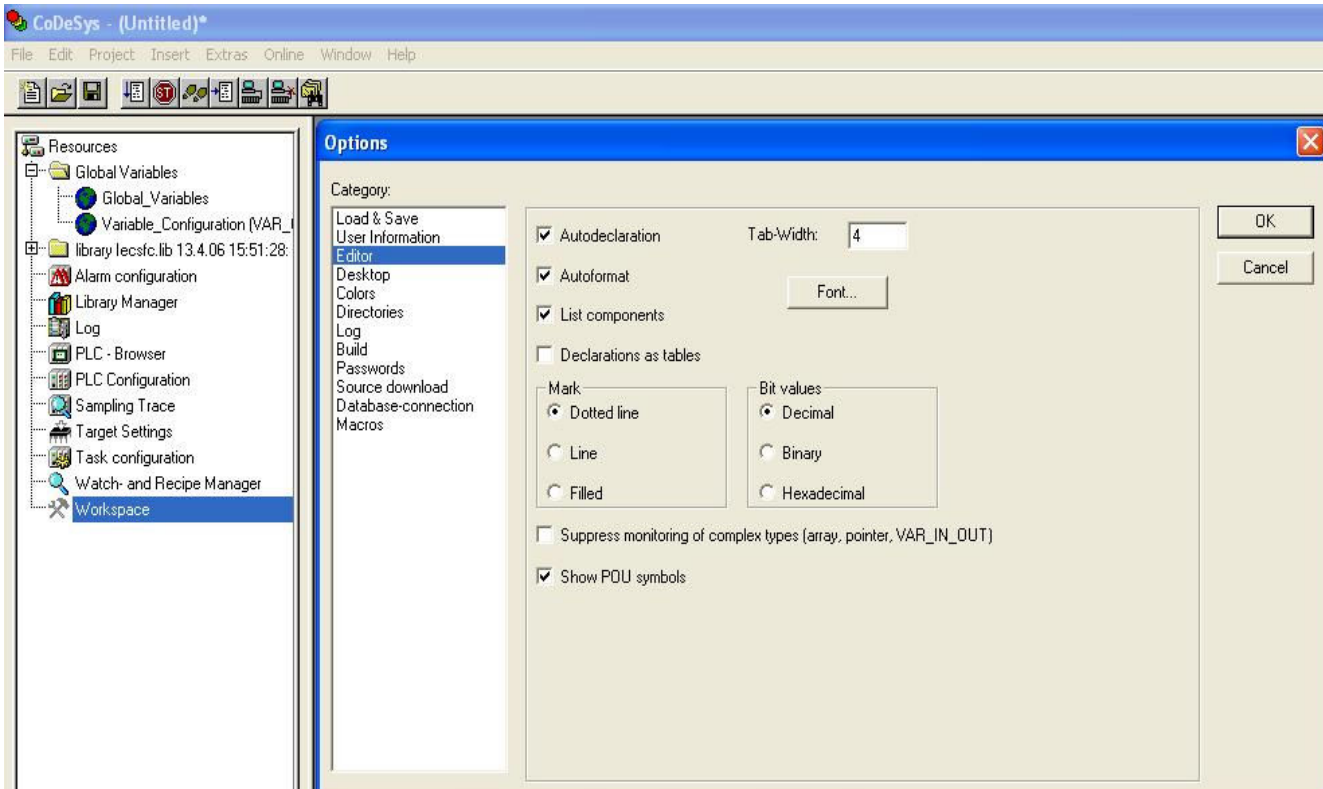
**PLC Configuration:** NON LO USIAMO perché non abbiamo un target

**Sampling Trace:** è un oscilloscopio per visualizzare le tracce dei segnali digitali che usiamo

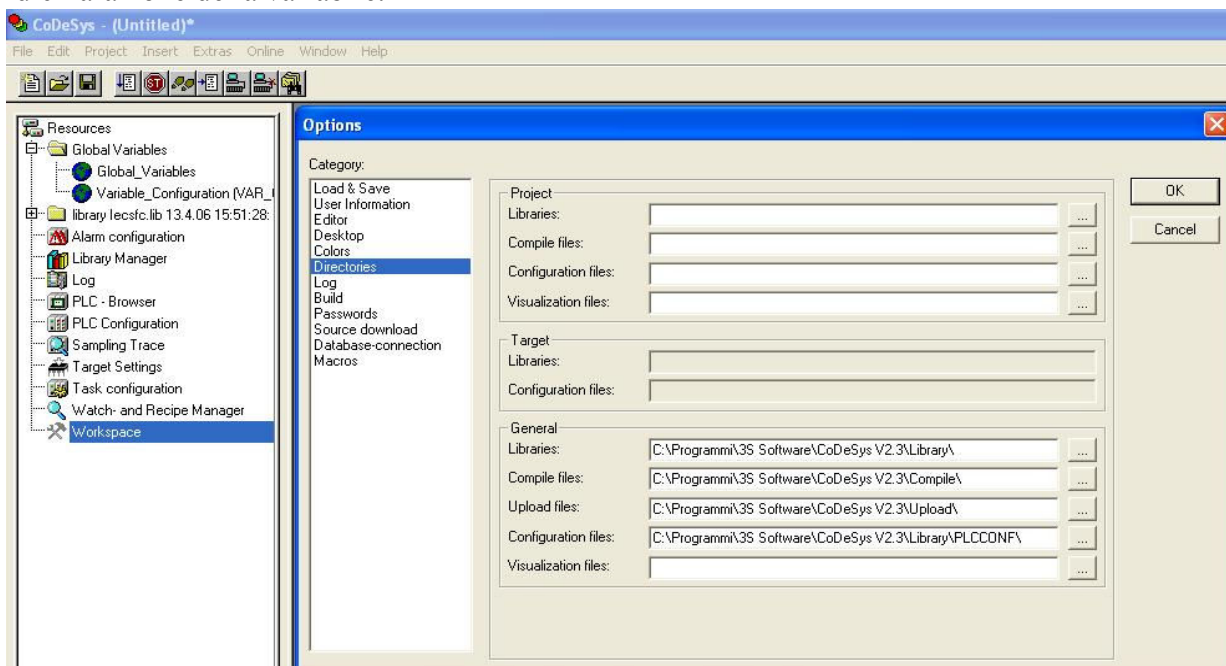
**Target Settings:** è la stessa finestra dell'inizio dove scegliamo il target (None)

**Task Configuration:** molto importante. Serve per agganciare i programmi ai task che li manderanno in esecuzione

**Workspace:** da qui configuro tutto l'ambiente di lavoro, tipo i font, i colori etc etc

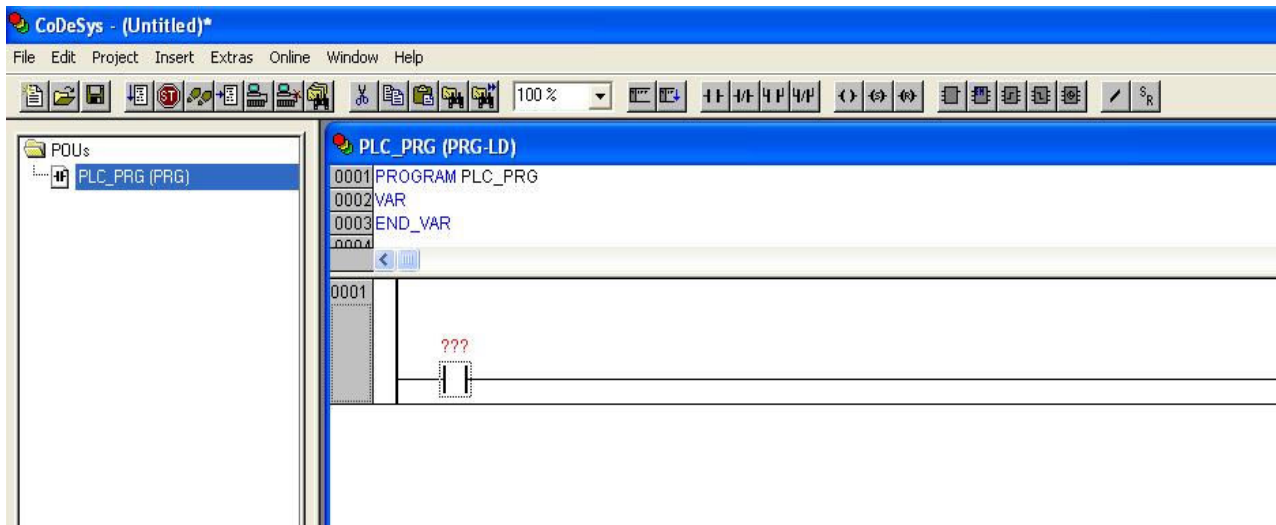


**Workspace → Editor:** Autodeclaration dev'essere selezionata, così quando scriviamo un codice e scriviamo il nome di una variabile non ancora dichiarata, Codesys apre automaticamente la finestra di dichiarazione della variabile.

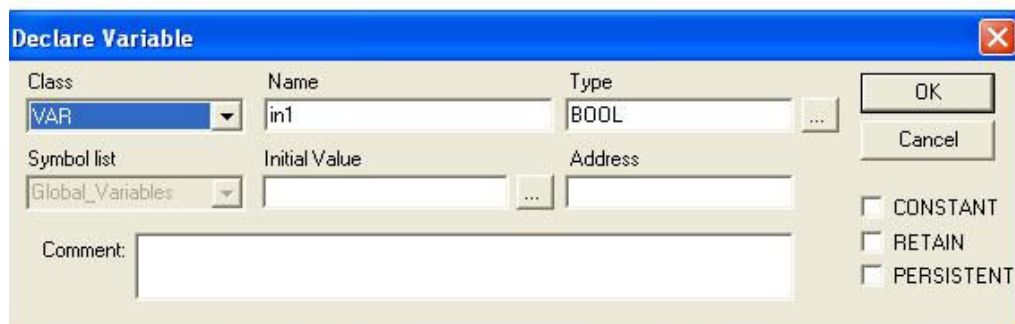


Workspace → Directories: le directories dove Codesys va a cercare tutte le risorse del progetto (librerie etc)

Tornando alla scrittura del codice di una POU:

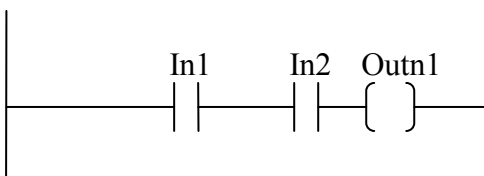


Quando inserisco un contatto o una bobina lui mette ???. Se in Workspace è settato Autodeclaration, scrivendo un nome in ??? che non sia ancora una variabile già dichiarata tra quelle locali alla POU, lui apre la finestra:



Da qui si configura la variabile.

Dopo aver fatto in ladder un rung del tipo:

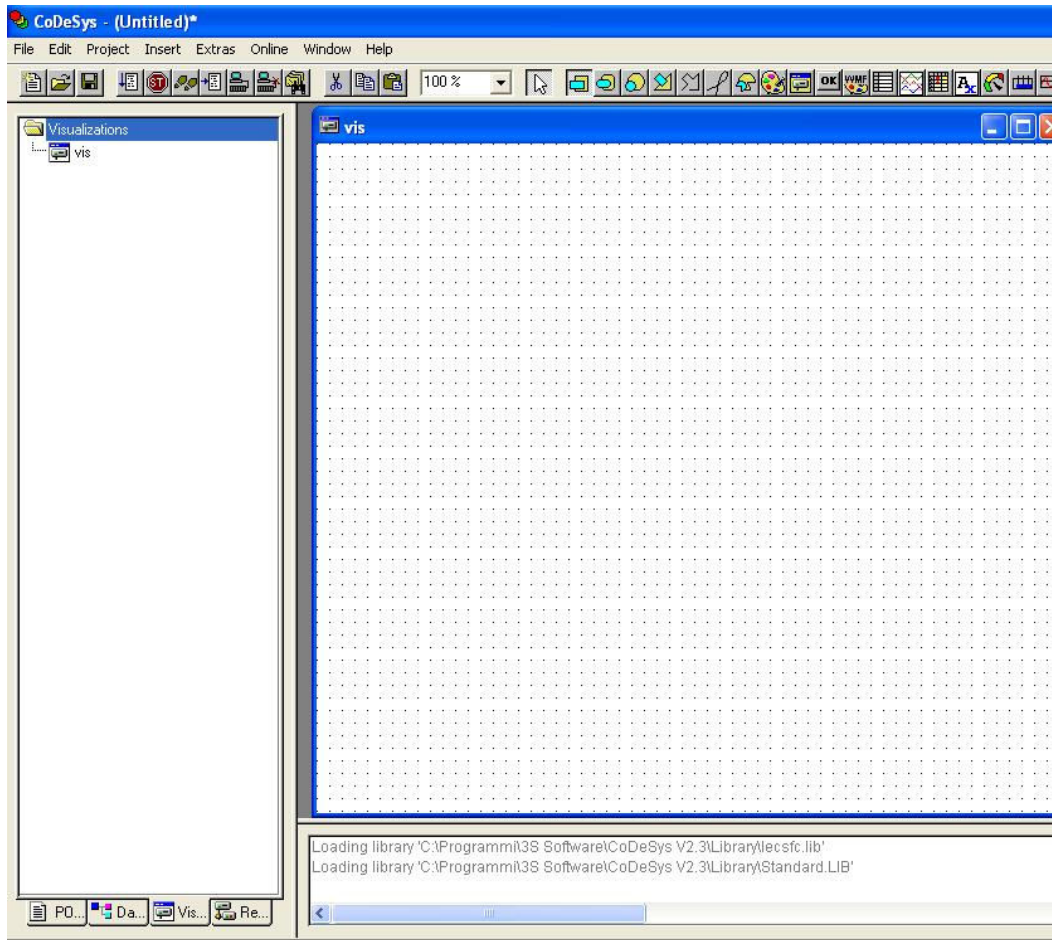


vogliamo visualizzarlo → bisogna configurare un oggetto visualizzazione.

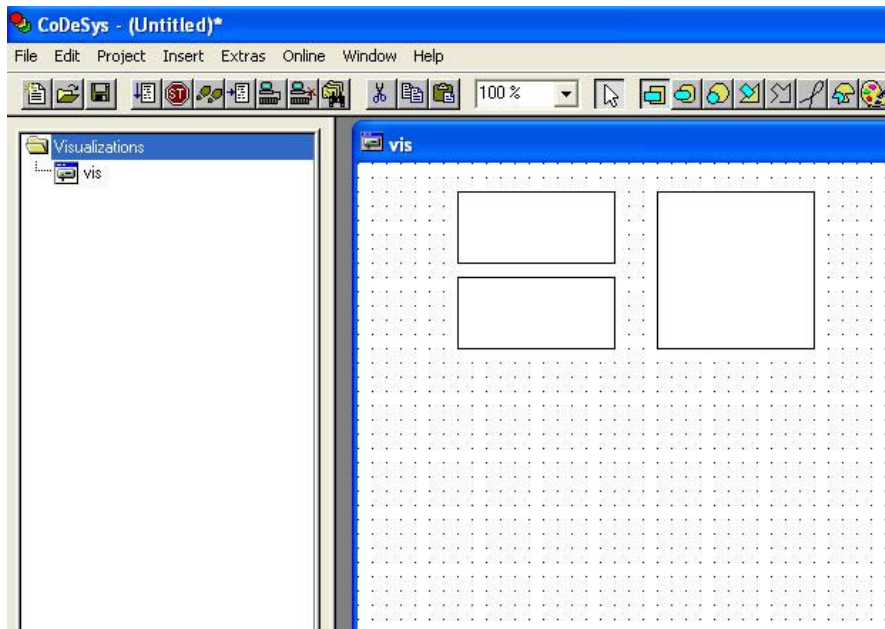
Si va in:



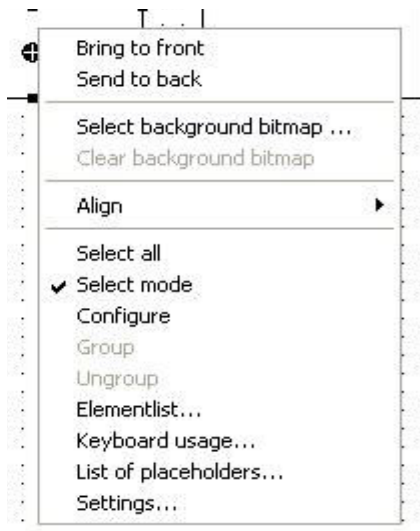
Click dx su Visualization → Add object → e si scrive il nome di una visualizzazione



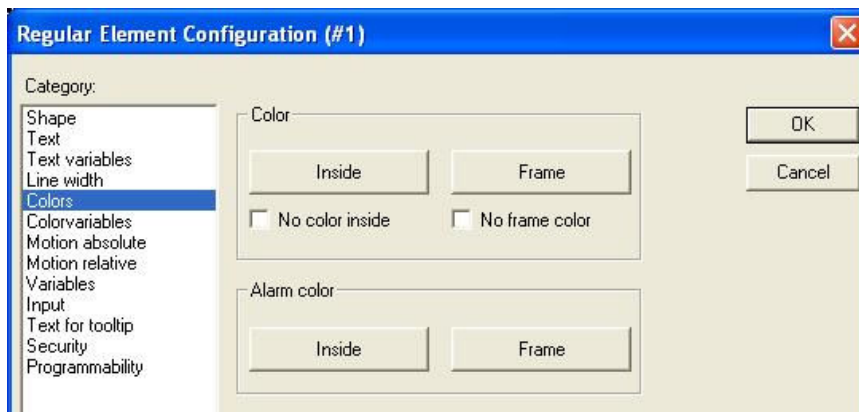
Quando si ha una visualizzazione, la finestra è del tipo sopra. Qui, con drag&drop, si inseriscono gli elementi.. Per il rung di ladder sopra, mettiamo 3 rettangoli.



Cliccando col dx su ogni elemento, si ottiene



Andando su Configure: →

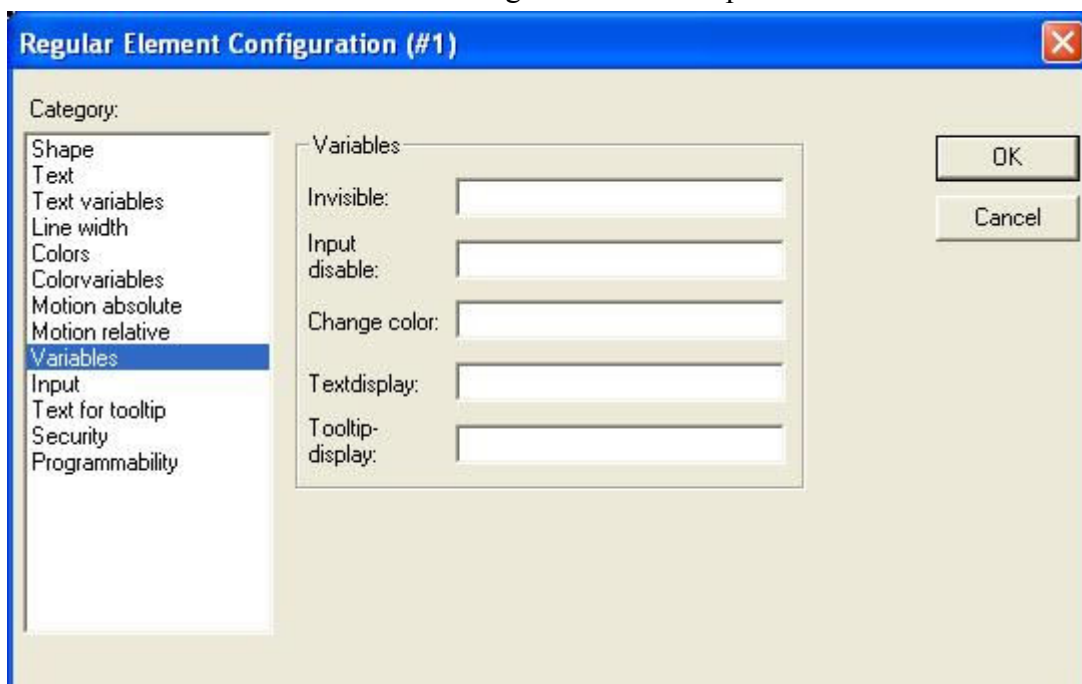


Colors → Inside è il colore dell'interno della figura che si è scelta (rettangolo, cerchio...)

Colors → Frame è il colore del bordo

Alarm Color → è il colore che la figura assume quando la variabile associata diventa vera.

Per associare una variabile al colore bisogna andare sull'opzione Variables :

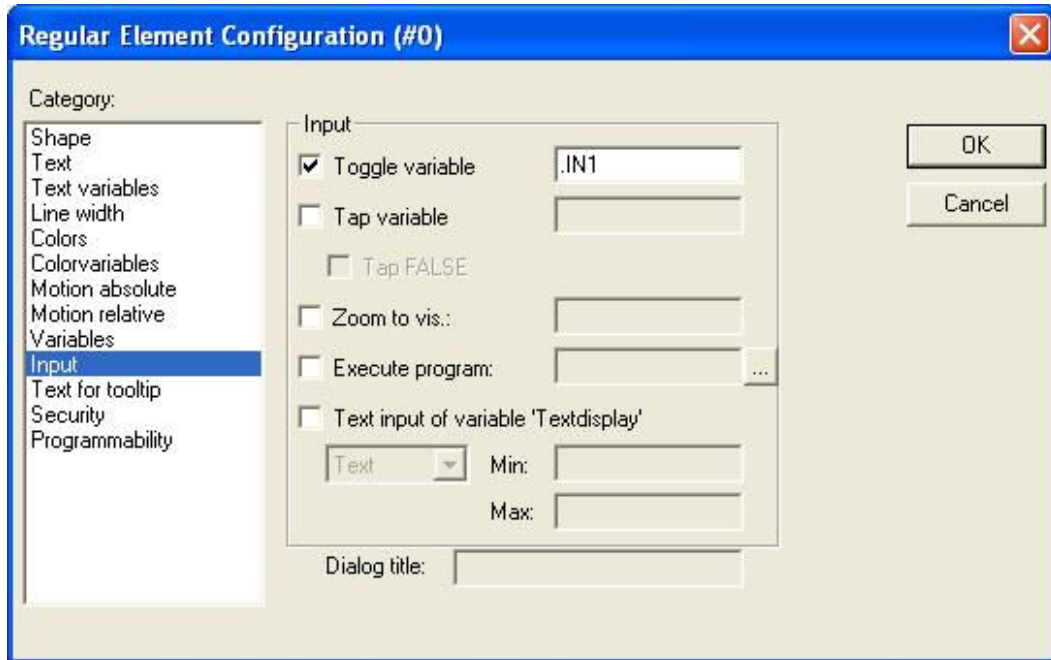


e inserire il nome di una variabile in change color.

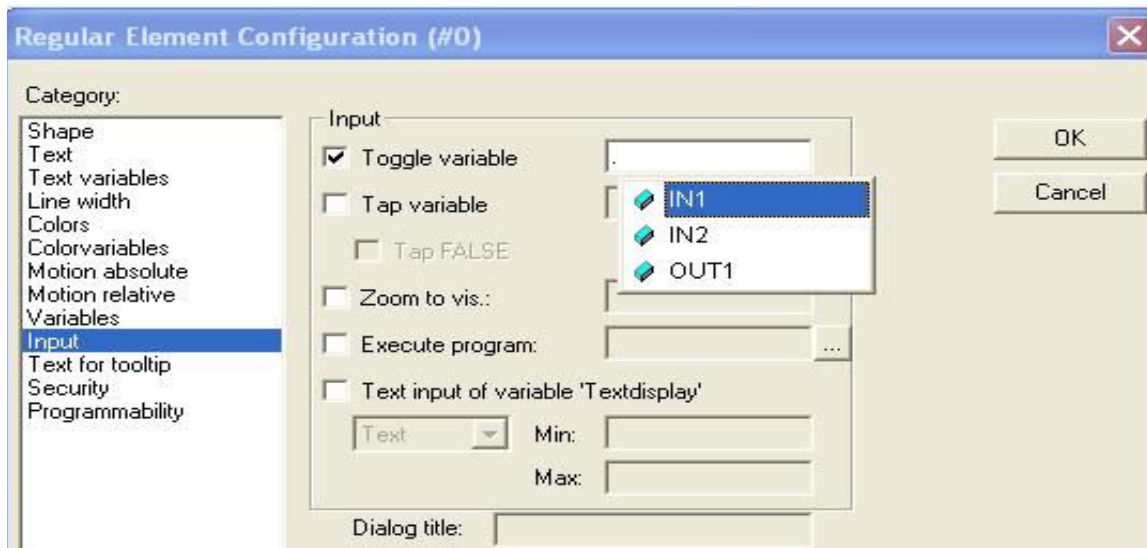
Sempre in questo menù variables:

invisibile: è la variabile che, se falsa mi rende invisibile l'oggetto, se vera lo rende visibile.

Se seleziono Input, invece, si apre:

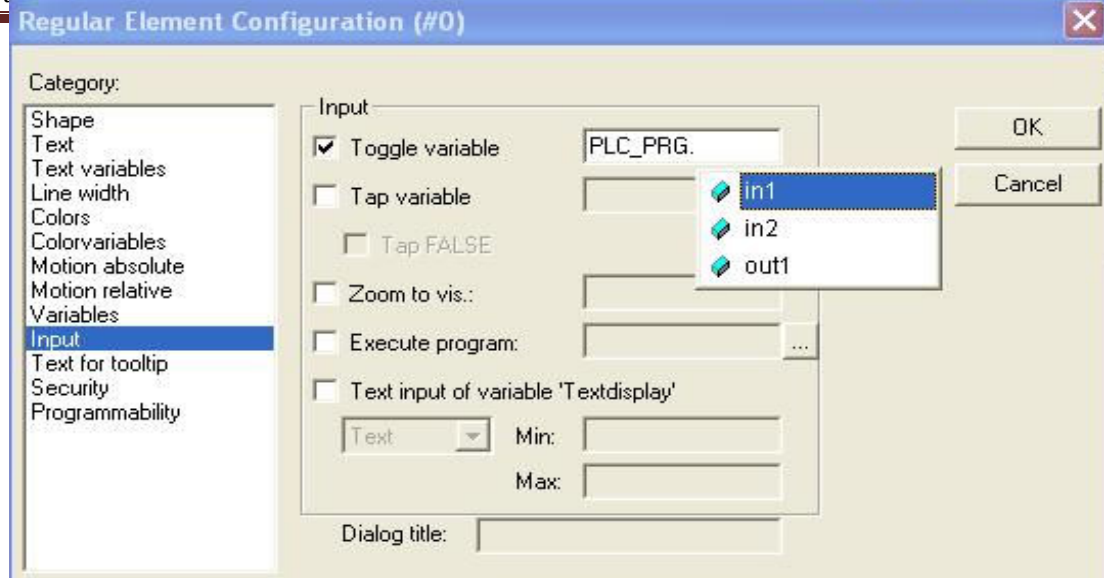


In questo modo sto associando una variabile all'oggetto grafico. Se la variabile che voglio associare è globale, mi basterà digitare nella finestrella il "." e comparirà tutta la lista delle var globali.



Se invece è locale, mi basterà digitare il nome della POU con quella var locale, seguita dal ".", e poi selezionare la variabile desiderata.



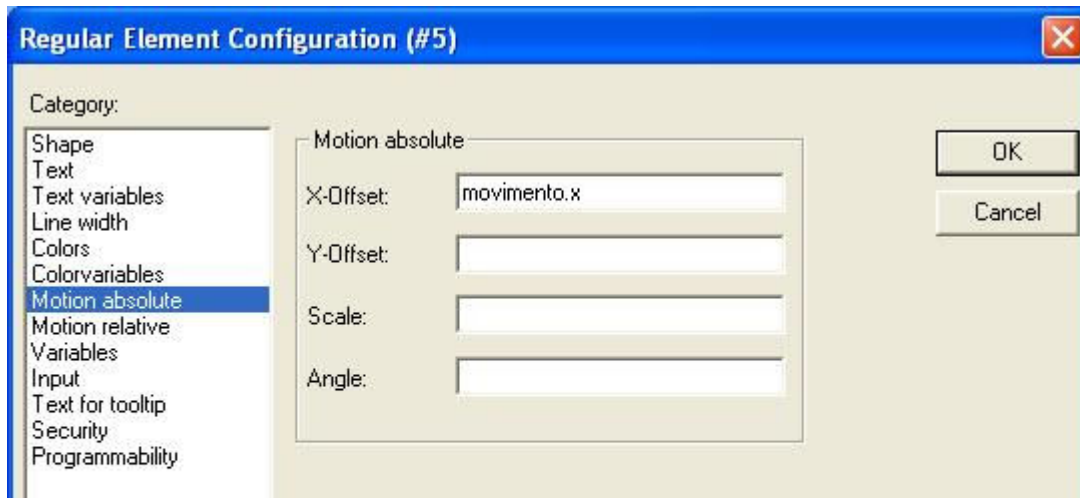


Se la variabile la metto Toggle, è come un interruttore, se la metto Tap è come un pulsante.

Per inserire un'immagine (.bmp o .jpg) basta andare sul menù in alto e selezionare Insert → bitmap. A questo punto bisogna tornare sullo schermo della visualizzazione (accanto alla freccetta del mouse dovrebbe esserci una tavolozza) e disegnare un rettangolo, che lui metterà con una croce diagonale. Rilasciando il mouse si apre la finestra in cui si deve selezionare l'immagine .

Per far muovere un oggetto bisogna associargli in Motion (absolute o relative) una variabile numerica.

**Es.**



Ho associato al movimento lungo l'asse x la variabile x della POU movimento (occhio che avendo aggiunto una POU, in questo caso fatta in ST, devo fare una append program call nel task configuration).

Per far sì che un oggetto grafico visualizzi il contenuto di una stringa e non una stringa fissa, bisogna creare l'oggetto, poi click dx → configure → text : dentro a content scrivere %s. Poi in configure → variables → text display scrivere il nome della variabile stringa (es. PLC\_PRG.stringa)