

Programmare con Codesys 3.5: la versione installata

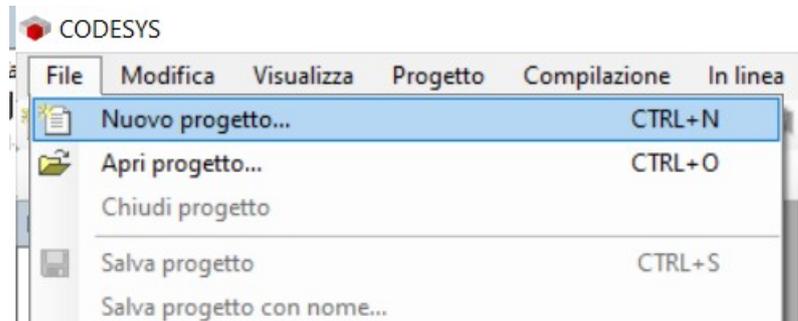
Versione di riferimento: 3.5.17.30 (dal sito della 3S software, in blu la versione che si è scaricata su Windows 10).

3.5.17.50	3.5.17.50 32 bit (.exe)	3.5.17.50 64 bit (.exe)	3.5.17.50 32 bit (.zip)	3.5.17.50 64 bit (.zip)	-	28.10.2022
3.5.17.40	3.5.17.40 32 bit (.exe)	3.5.17.40 64 bit (.exe)	3.5.17.40 32 bit (.zip)	3.5.17.40 64 bit (.zip)	-	10.06.2022
3.5.17.30	3.5.17.30 32 bit (.exe)	3.5.17.30 64 bit (.exe)	3.5.17.30 32 bit (.zip)	3.5.17.30 64 bit (.zip)	-	25.01.2022

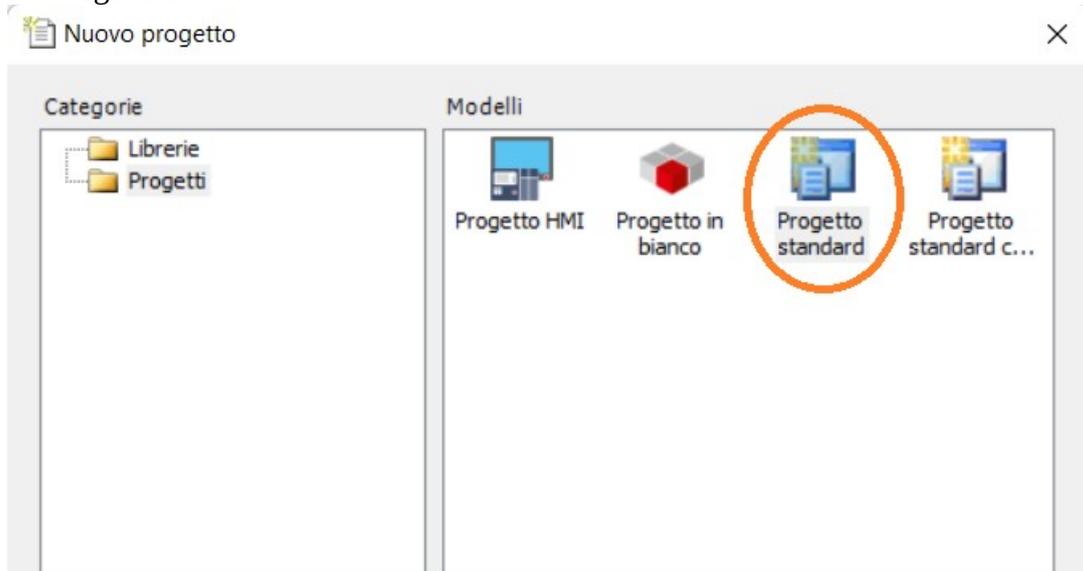
Primo progetto

Dopo l'installazione, aprire Codesys 3.5

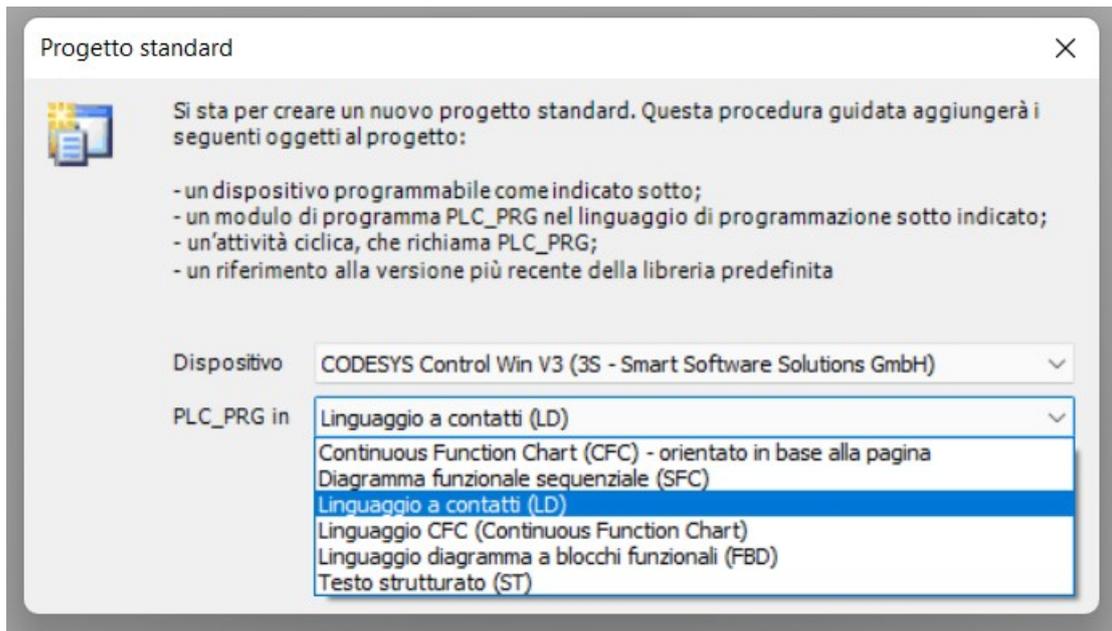
Dal menù File → Nuovo progetto



scegliere "Progetto standard"

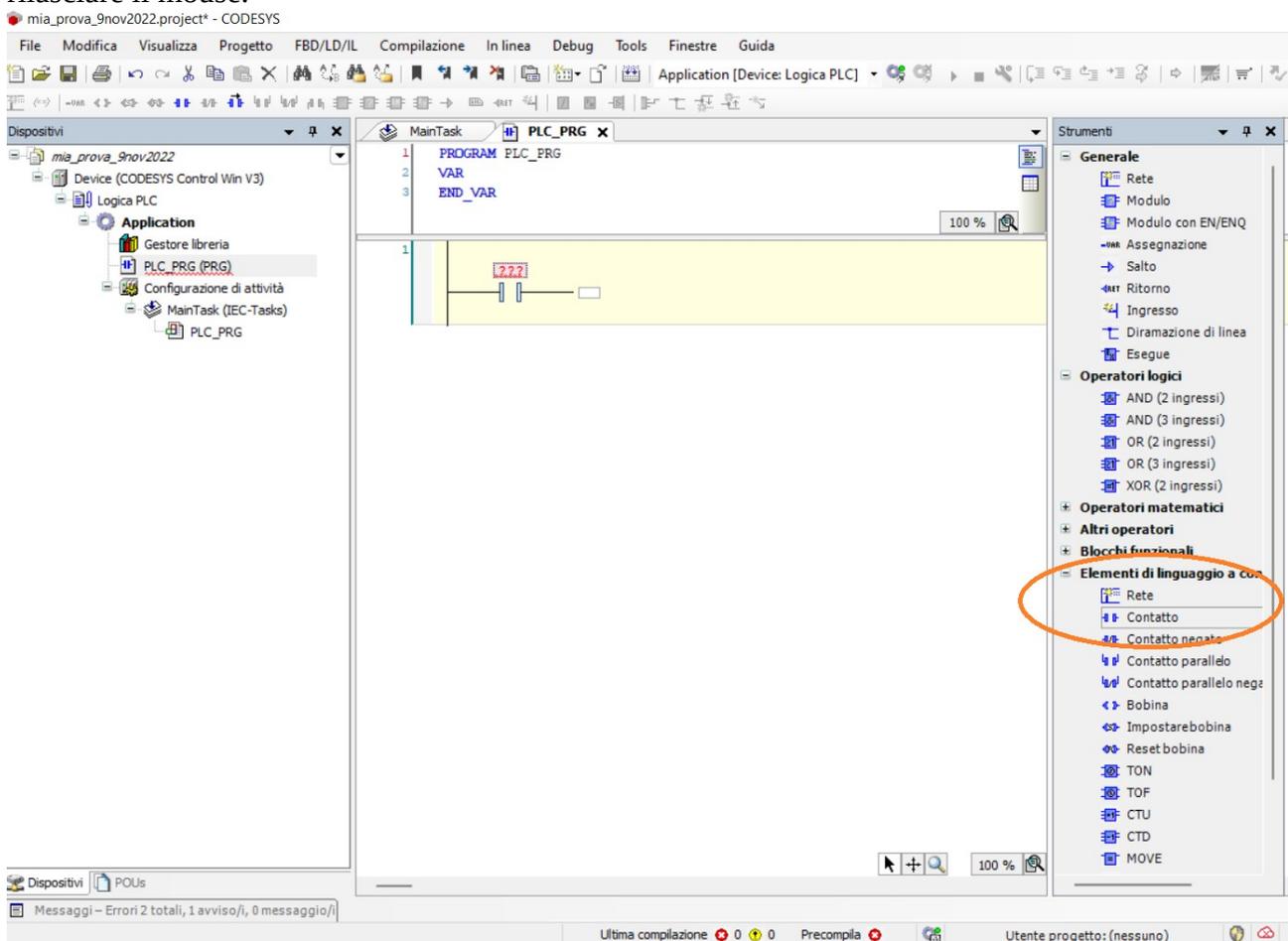


Nel campo "Dispositivo" selezionare come nell'immagine sotto (CODESYS Control Win V3...), e scegliere "Linguaggio a Contatti".

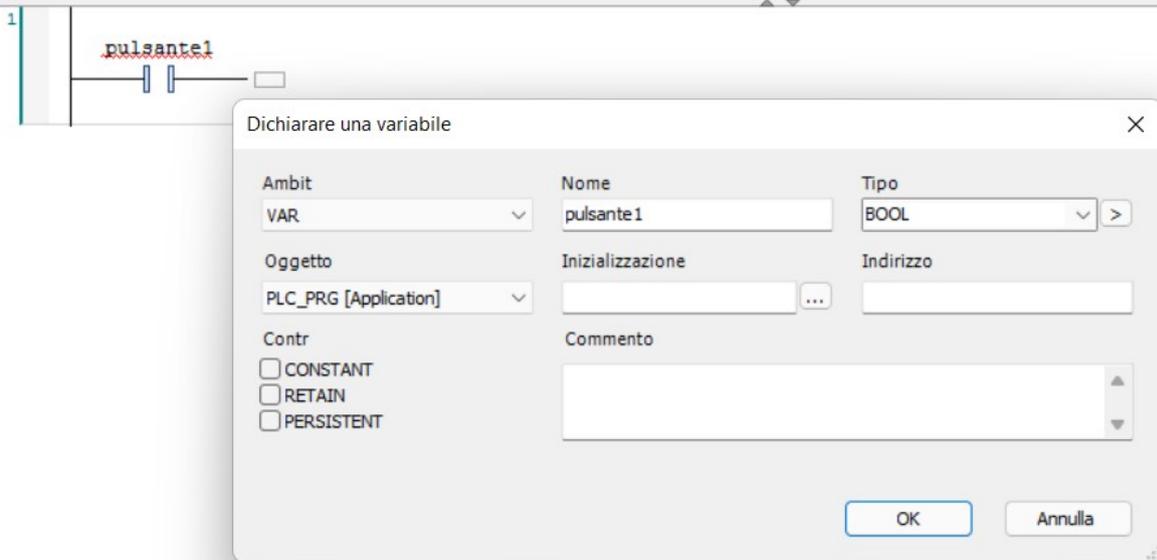


Cliccare due volte su “PLC_PRG”, l’ultima voce in fondo a tutte quelle presenti nella finestra “Dispositivi”, a sinistra. Questo è il nome di default associato al primo programma creato. Popolare la riga di ladder con contatti/bobine/FB presenti nella finestra di destra, chiamata “Strumenti”.

Ad esempio, per inserire un contatto, selezionare il contatto dalla finestra di destra e, con DRAG&DROP (cioè cliccandoci su e tenendo cliccato) trascinare il contatto sulla riga. Poi rilasciare il mouse.

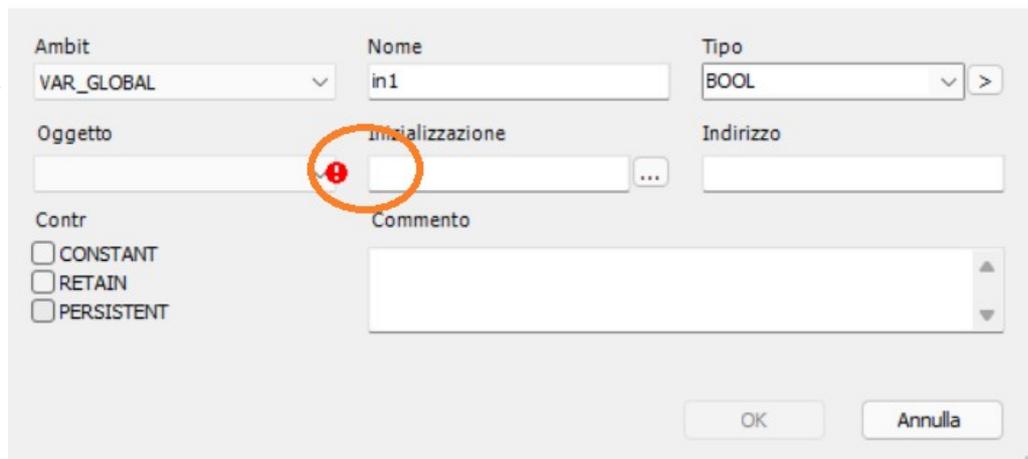


Chiamando il contatto “pulsante1” appare la finestra di autodichiarazione. Supponiamo di volere questa variabile GLOBALE.

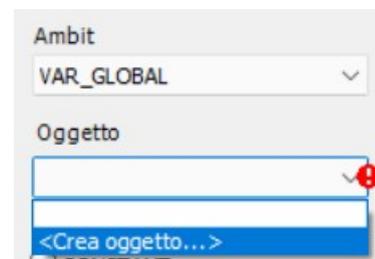


Nel campo “Ambit” (che è l’equivalente di “Class” per Codesys 2.3) scegliere VAR GLOBAL, poi mettere l’indirizzo della variabile hardware, %IX0.0.

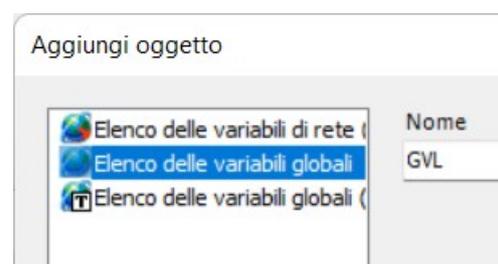
Apparirà un punto interrogativo rosso.



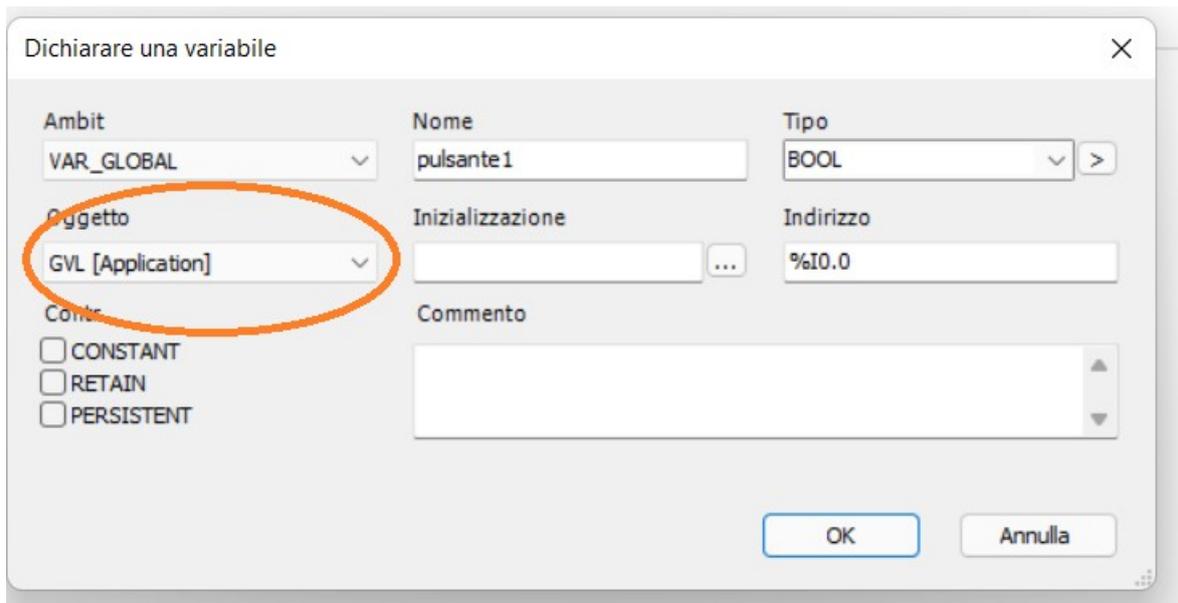
Nel campo “Oggetto” scegli “Crea oggetto”



Si apre una nuova finestra. Lascia tutto come appare e clicca su Aggiungi. In sostanza questa versione di Codesys, a differenza della 2.3, ha bisogno di dare un nome al contenitore delle variabili globali. Il suo nome di default è GVL (Global Variable List).

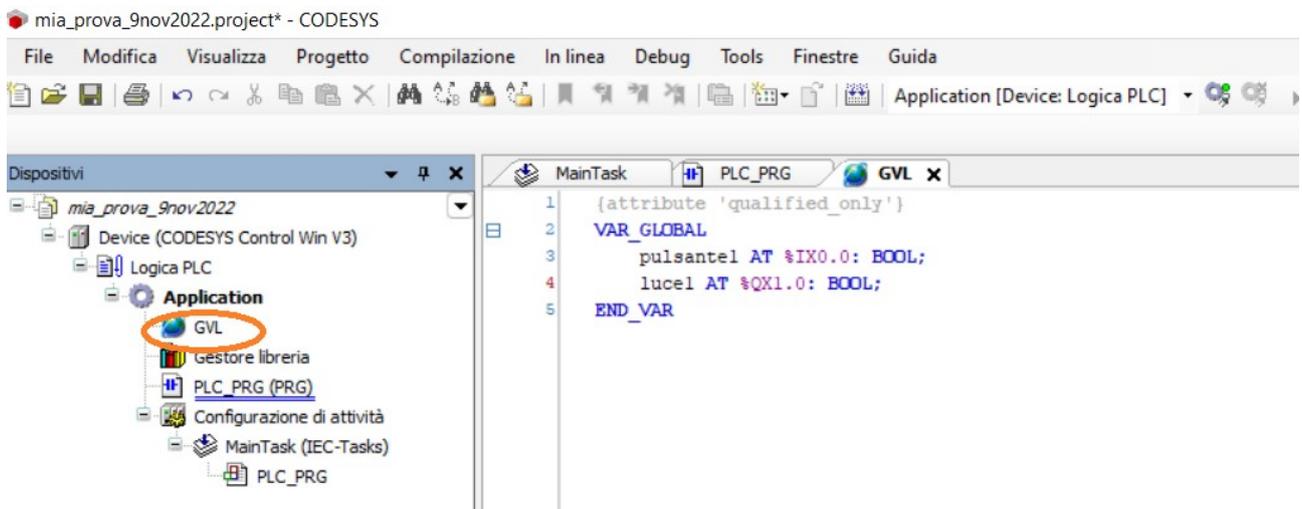


Quando poi assegneremo ai nostri contatti o alle nostre bobine le variabili globali, le dovremo chiamare GVL.nomevariabile, per dire che è una globale.



A questo punto nella struttura del programma è comparsa una nuova voce, sotto Application, che è appunto GVL, la lista delle variabili globali.

Puoi anche aggiungere a mano (cioè scrivendo la riga carattere per carattere), dentro a GVL, la variabile luce1 AT %QX1.0:BOOL;



Oltre al contatto, trascinare sulla riga di ladder anche una bobina standard, e associarla alla variabile luce1 globale.

A questo punto la riga di ladder diventa:



NOTA BENE che le variabili globali, per essere riconosciute dal software, devono appunto avere il nome del “contenitore” delle globali, che da noi si chiama GVL, più il “.” più il nome della variabile. Quindi alla fine, ciò che originariamente si chiamava “pulsante1” ora si chiama “GVL.pulsante1”.

Idem per l'uscita, che si chiama GVL.luce1.

NOTA BENE: Codesys 3.5, a differenza di Codesys 2.3, vuole che si specifichi sempre, anche per i booleani, la dimensione del dato, in caso di variabili AT. Quindi nelle variabili globali abbiamo dovuto scrivere anche la X, sia nell'ingresso che nell'uscita, altrimenti ci avrebbe dato un errore. Invece che %I0.0 abbiamo scritto %IX0.0

Per mandare in esecuzione il codice andiamo sul menù "Compilazione" → "Crea codice"

Qui vediamo se abbiamo fatto degli errori o meno.

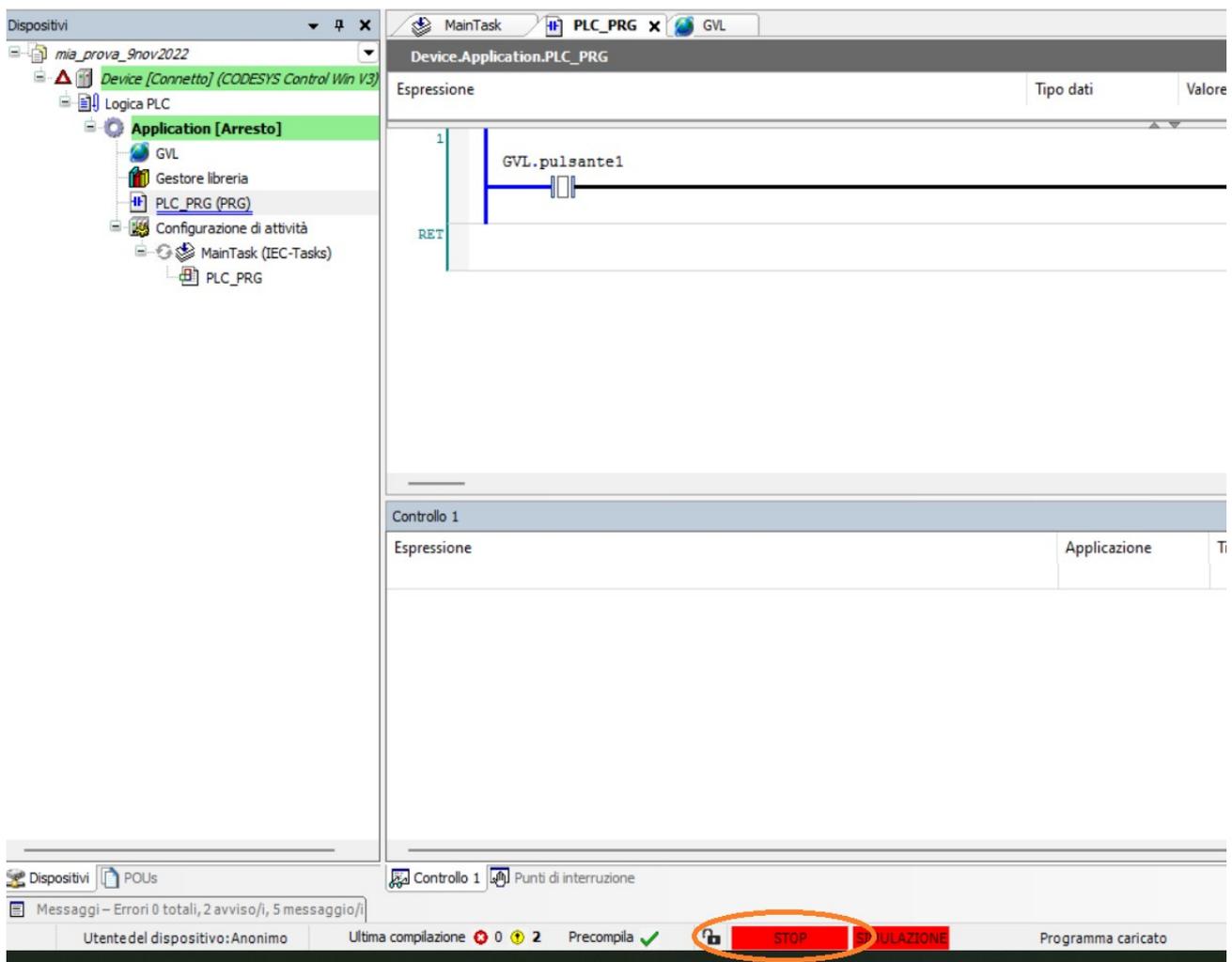
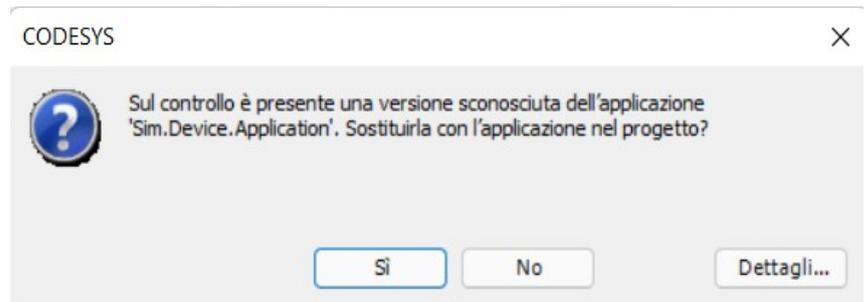
Se è tutto ok procediamo e andiamo nel menù "In linea" → "Simulazione"

Poi "In Linea" → "Login"

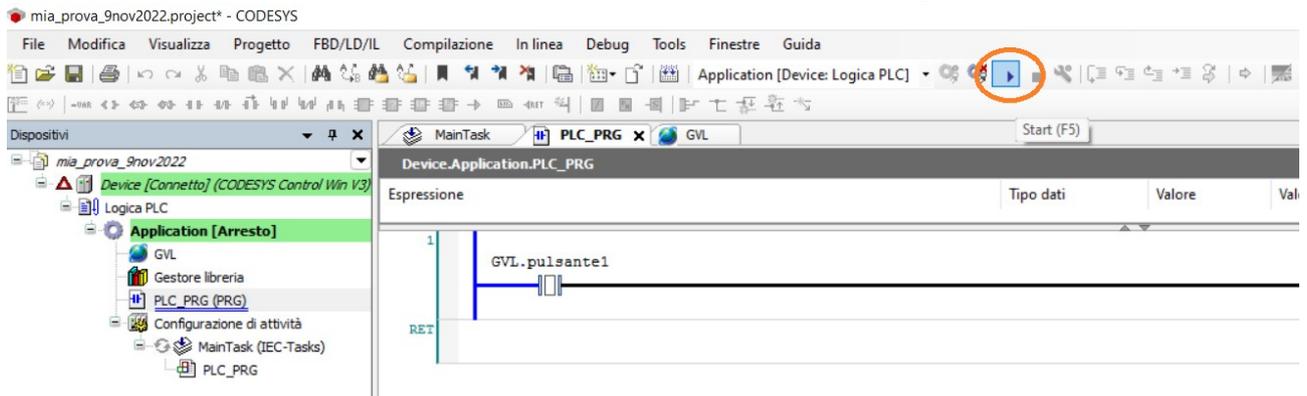
Salta fuori questa finestra

Dare l'ok (Si)

Di norma si parte in stato di stop,
visibile dalla scritta rossa STOP in
basso



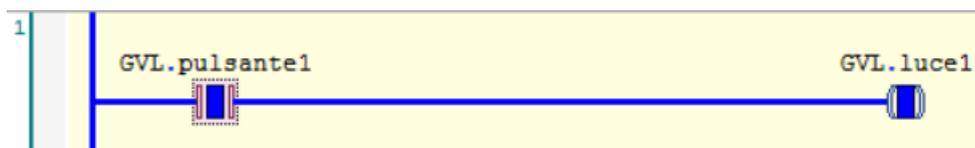
Per andare in START (cioè in RUN del PLC) cliccare sull'icona a triangolo in alto



Adesso la parte sotto è diventata verde e dice "IN ESECUZIONE"

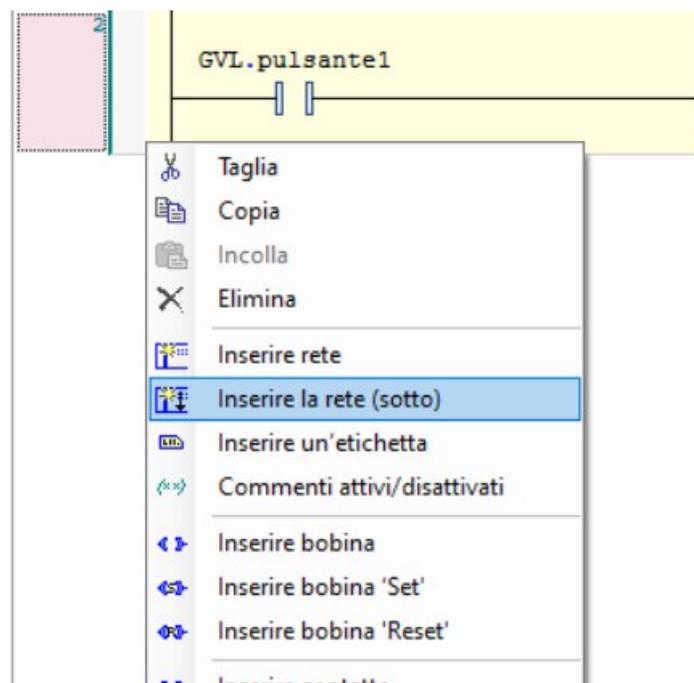


Qui si può operare come Codesys 2.3, ovvero cliccando due volte sul contatto finché diventa blu e poi premendo CTRL + F7, in modo da forzare il cambio di stato del contatto, come se stessi fisicamente agendo sull'oggetto hardware collegato al morsetto %IX0.0. Vedremmo così che la bobina diventa true.



Ora usciamo dallo stato di esecuzione (RUN) e torniamo a modificare il codice. Per fare questo vado nel menù "In Linea" → "Logout" e poi, sempre nel menù In Linea, tolgo la spunta da "Simulazione".

Ora aggiungo una nuova riga di ladder. Clicco col tasto dx del mouse sotto alla riga1 e seleziono "Inserire la rete (sotto)"



Dai menù a destra inserisco sul nuovo rung un FB Ton (sempre con DRAG&DROP).

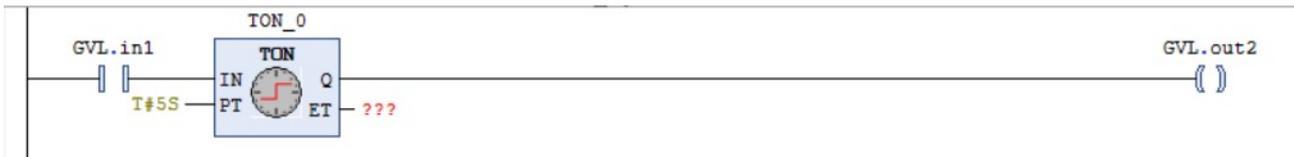
A differenza di Codesys 2.3, in Codesys 3.5 i timer sono già disponibili, senza dover richiamare alcuna libreria aggiuntiva.

Codesys crea così un'istanza del FB Ton e automaticamente mi mette anche il simbolo di contatto in ingresso al blocco. L'istanza la chiamo TON_0 e la dichiaro come locale (quindi, quando si apre l'autodichiarazione del blocco, in Ambient lascio VAR).

Sul contatto ci scrivo la variabile "in1", per la quale farò la stessa cosa di prima: in Ambient metto GLOBAL, nel suo Address devo ricordarmi di mettere la X (quindi sarà %IX0.1) poi dovrò riscriverla come "GVL.in1".

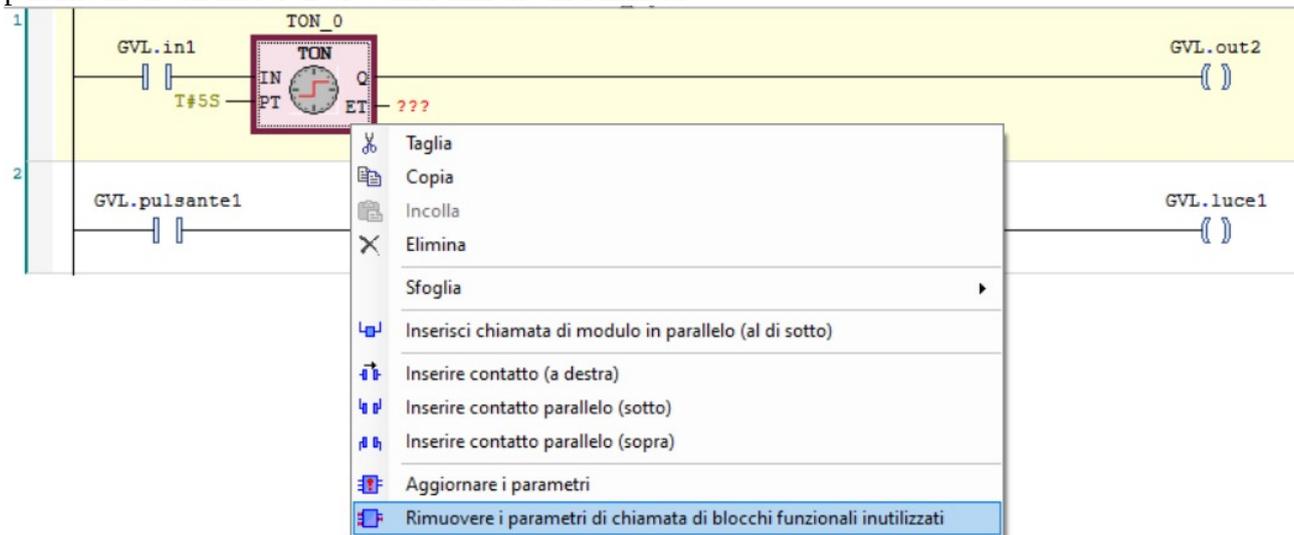
Infine, dopo il FB, su Q aggiungo una bobina, che assocerò alla variabile globale GVL.out2, associata a %QX1.1.

Quindi la nuova riga è così:



A differenza di Codesys 2.3, Codesys 3.5 lascia segnati in rosso i parametri del blocco che non sono stati usati.

Per evitare errori clicchiamo col pulsante dx del mouse sul Ton e selezioniamo "Rimuovere i parametri di chiamata di blocchi funzionali inutilizzati"



Così ci toglie i punti interrogativi rossi sul parametro ET, e il tutto diventa:



Adesso compilo e mando in esecuzione.

Quindi:

- menù "Compilazione" → "Crea codice"

Se tutto è ok:

- menù "In linea" → "Simulazione"

- poi menù "In Linea" → "Login"

- poi premere lo start (icona a triangolo)

Anche in questo caso possiamo agire sul contatto GVL.in1 con il doppio click + CTRL+F7 e far partire il timer.

Come in Codesys 2.3, se mentre eseguiamo teniamo in vista la finestra delle variabili, in questo caso locali, posso vedere il tempo del timer che trascorre, istante per istante.

Espressione	Tipo dati	Valore	Valore preparato	Indirizzo	Commento
TON_0	TON				
IN	BOOL	TRUE			Rising edge: starts delay counter
PT	TIME	T#5s			Time for the delay counter [ms]
Q	BOOL	FALSE			! "FALSE" if "IN" is "FALSE"
ET	TIME	T#3s720ms			Elapsed time since rising edge at "IN"

Si noterà che mentre TON e TOFF sono subito visibili dalla lista dei possibili function block (lista presente nella finestra di Codesys, a destra), il TP non lo è.

In realtà anche lui è già inglobato di default nel progetto software.

Creiamo un'altra riga di ladder andando sulla riga 2, cliccando col tasto dx del mouse e selezionando "Inserire la rete (sotto)"

Poi clicchiamo col tasto dx del mouse sulla riga vuota e selezioniamo la voce "Inserire chiamata di modulo" (che sarebbe l'equivalente di "Inserire un FB").

Ritrovo così la classica lista di function block standard presente in Codesys 2.3

Creo quindi la terza riga così come sotto, inserendo le opportune variabili globali e mantenendo locale l'istanza del TP che ho chiamato TP_0

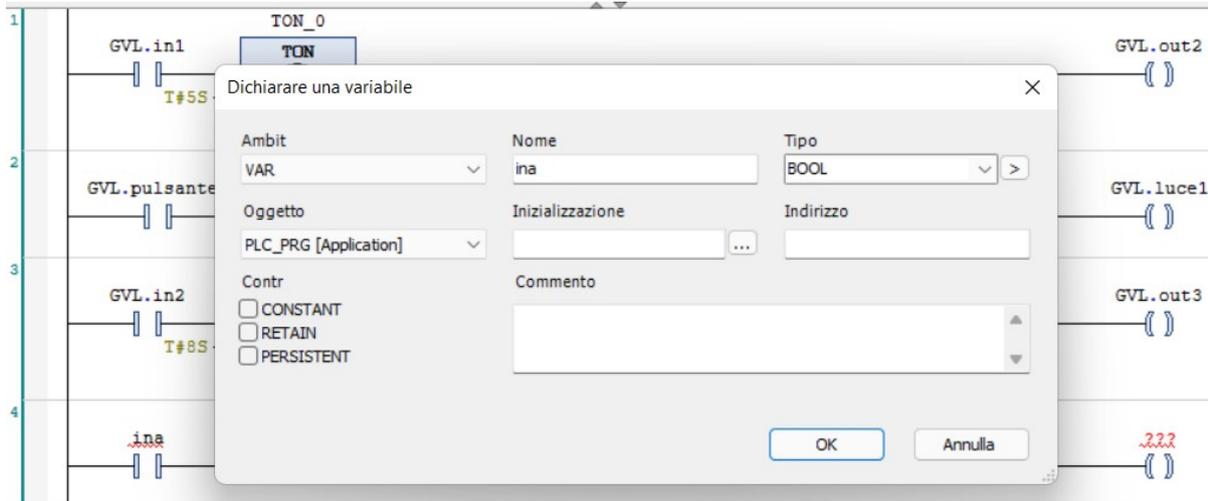
Ricerca testo	Categorie	Nome	Tipo	Origine
	Blocchi funzionali			
	Chiamate di moduli			
	Parole chiave			
	Conversioni			
		Standard	Libreria	Standard, 3.5.17.0 (...)
		Bistabile Fu...		
		Counter		
		Miscellane...		
		Timer		
		TOF	FUNCTION_BLOCK	Standard, 3.5.17.0 (...)
		TON	FUNCTION_BLOCK	Standard, 3.5.17.0 (...)
		TP	FUNCTION_BLOCK	Standard, 3.5.17.0 (...)
		Trigger		



I commenti in Codesys 3.5

Proviamo ad inserire una quarta riga, fatta solo di un contatto e di una bobina che lasciamo locali, e che chiamiamo “ina” e “outa”.

Quando ci apparirà l’autodichiarazione, se li vogliamo locali ricordiamo di lasciare selezionato, nel campo “Ambit”, VAR (cioè VAR LOCALI).



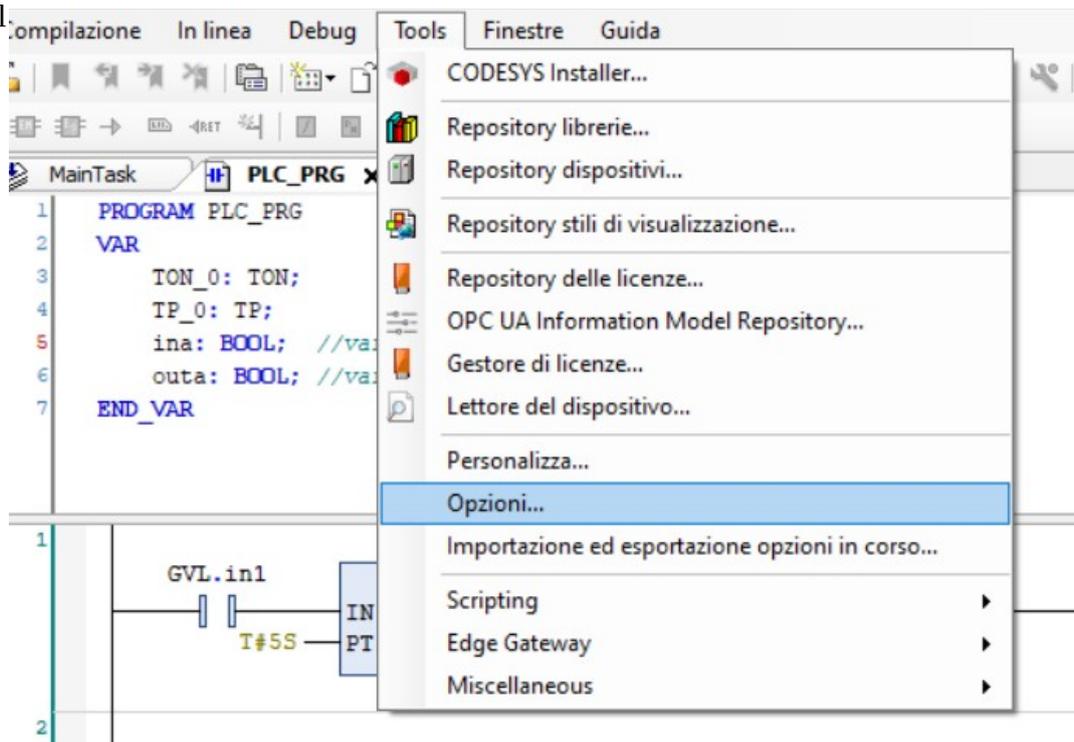
Le variabili vengono correttamente inserite nella parte delle variabili locali di PRG_PRG

```
PROGRAM PLC_PRG
VAR
    TON_0: TON;
    TP_0: TP;
    ina: BOOL;
    outa: BOOL;
END_VAR
```

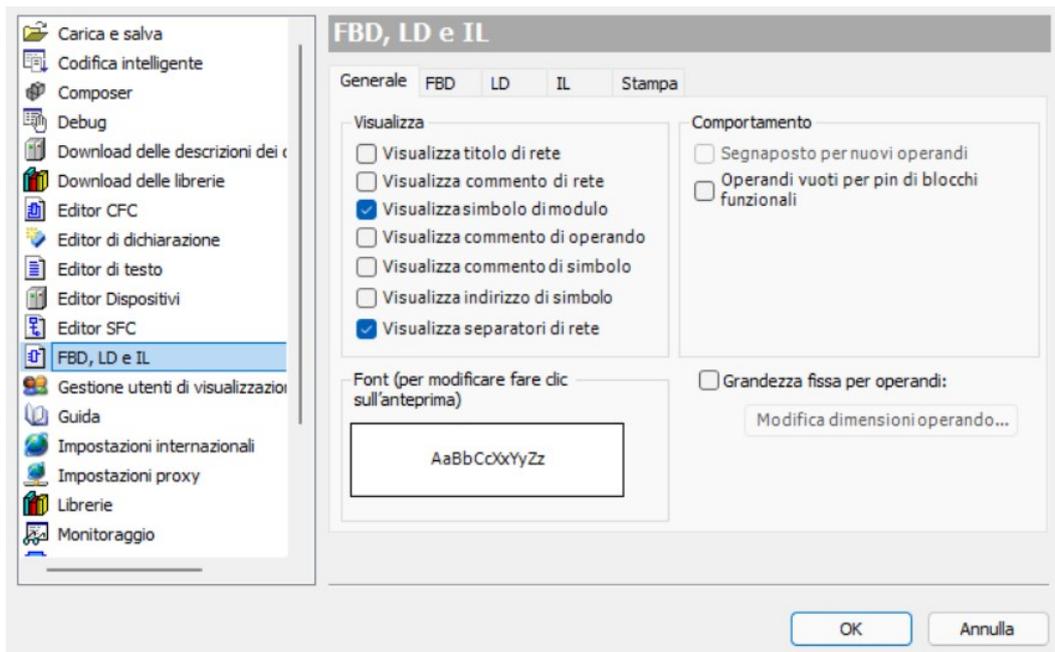
Aggiungiamo a mano i commenti, che in questo caso sono identificati non più da (*.*) come in Codesys 2.3 ma, come nel linguaggio C, dagli slash //

```
TON_0: TON;
TP_0: TP;
ina: BOOL; //variabile di ingresso
outa: BOOL; //variabile di uscita
END_VAR
```

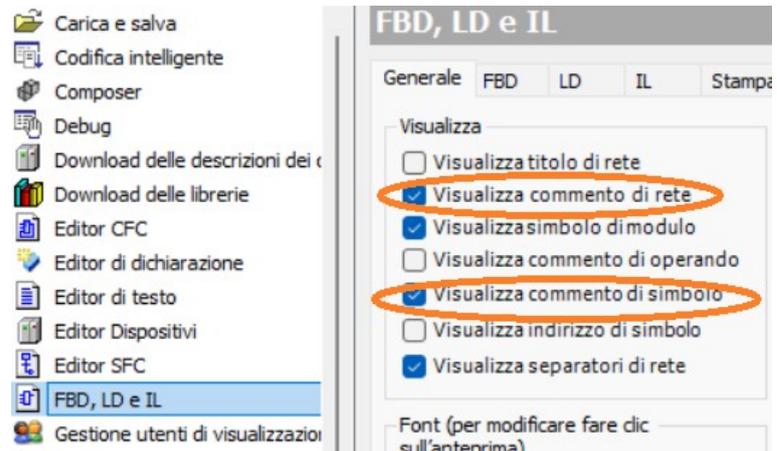
Ora andiamo nel menù "Tools" → "Opzioni"



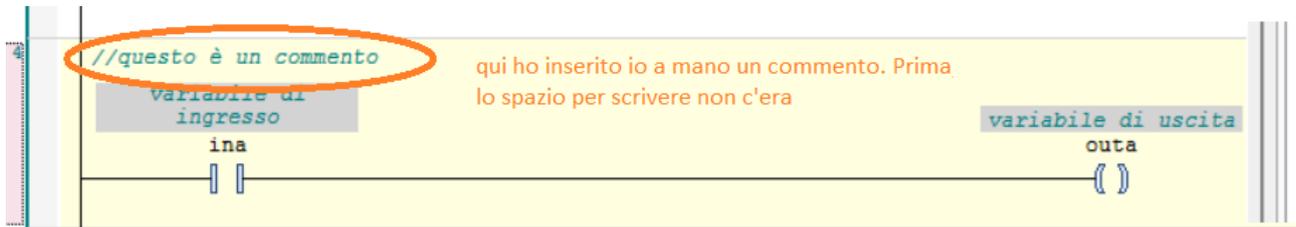
Qui si apre una finestra, da cui selezioniamo, a sinistra, la voce "FBD, LD, IL"



Mettiamo la spunta sulle voci "Visualizza commento di simbolo" e "Visualizza commento di rete"



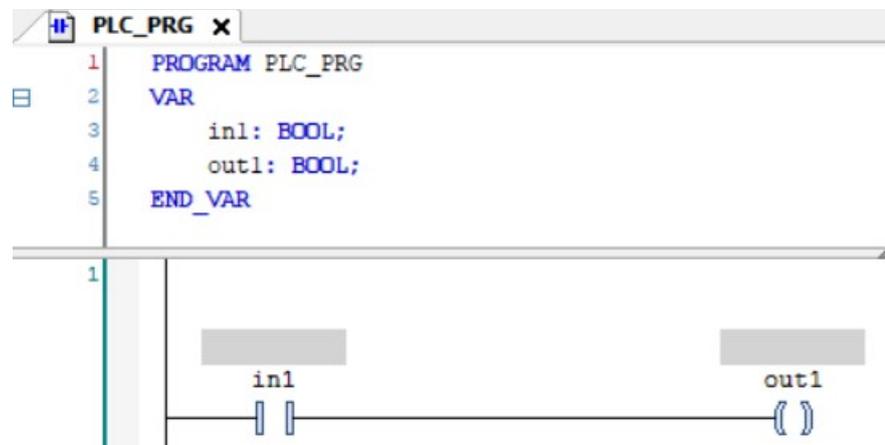
In questo caso vedo sia apparire sopra al contatto e alla bobina i loro commenti, che apparire lo spazio per poter scrivere io un commento sulla riga di ladder. Come si vede dal menù sopra, posso anche decidere di far apparire l'indirizzo fisico, sopra al contatto o alla bobina, se sono associati ad un qualche %I... o %Q...



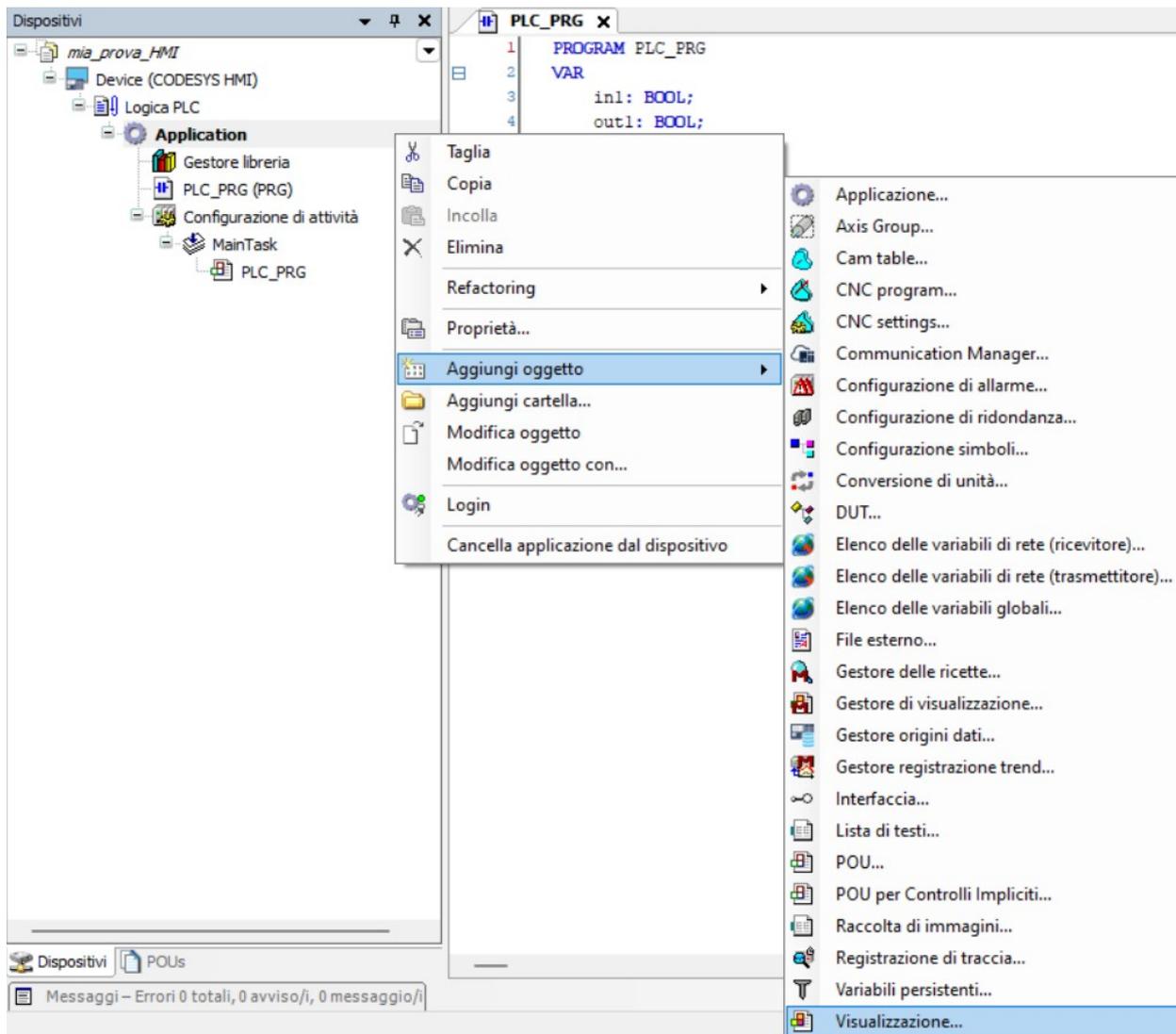
Gestire una visualizzazione grafica

Facciamo un nuovo progetto col classico PLC_PRG in ladder.

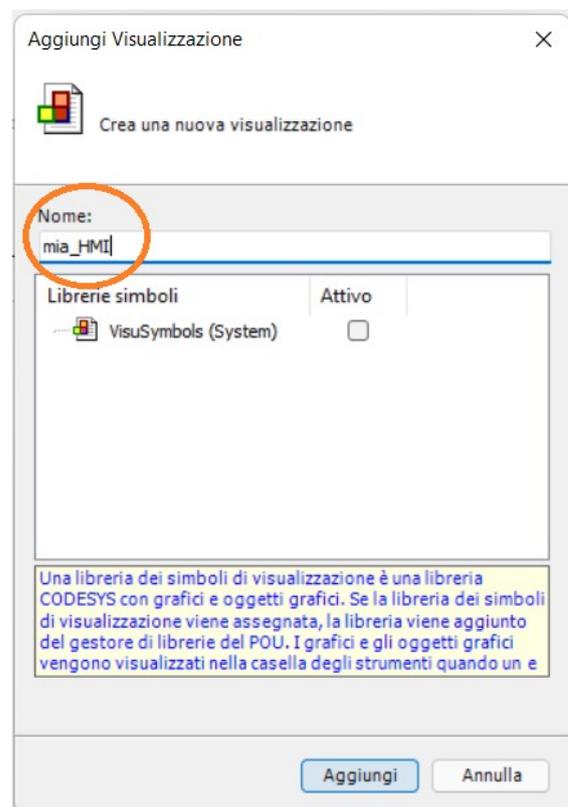
Inserisco una riga con un contatto e una bobina, che posso lasciare locali e che chiamerò rispettivamente "in1" e "out1". Quindi avrò questo:



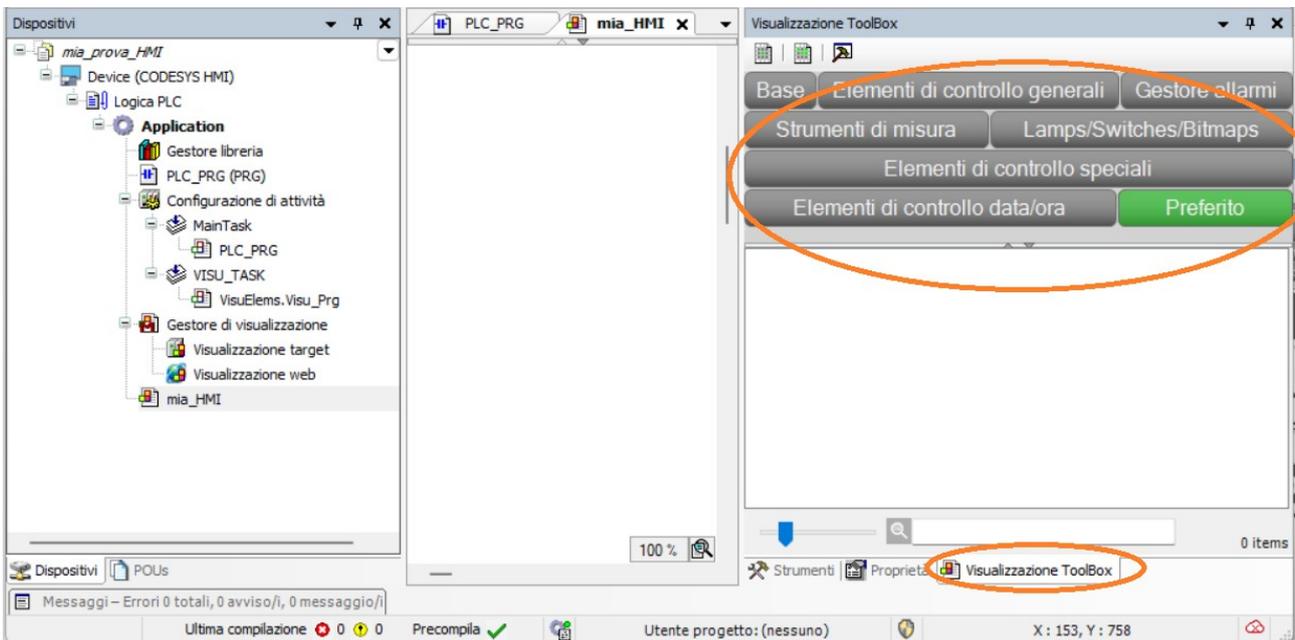
A questo punto, nella finestra verticale, a sinistra, chiamata "Dispositivi", individuo "Application", ci clicco sopra col destro del mouse, e seleziono "Aggiungi oggetto" → "Visualizzazione". Da notare che Visualizzazione è l'ultima voce di un menù molto lungo che si è aperto.



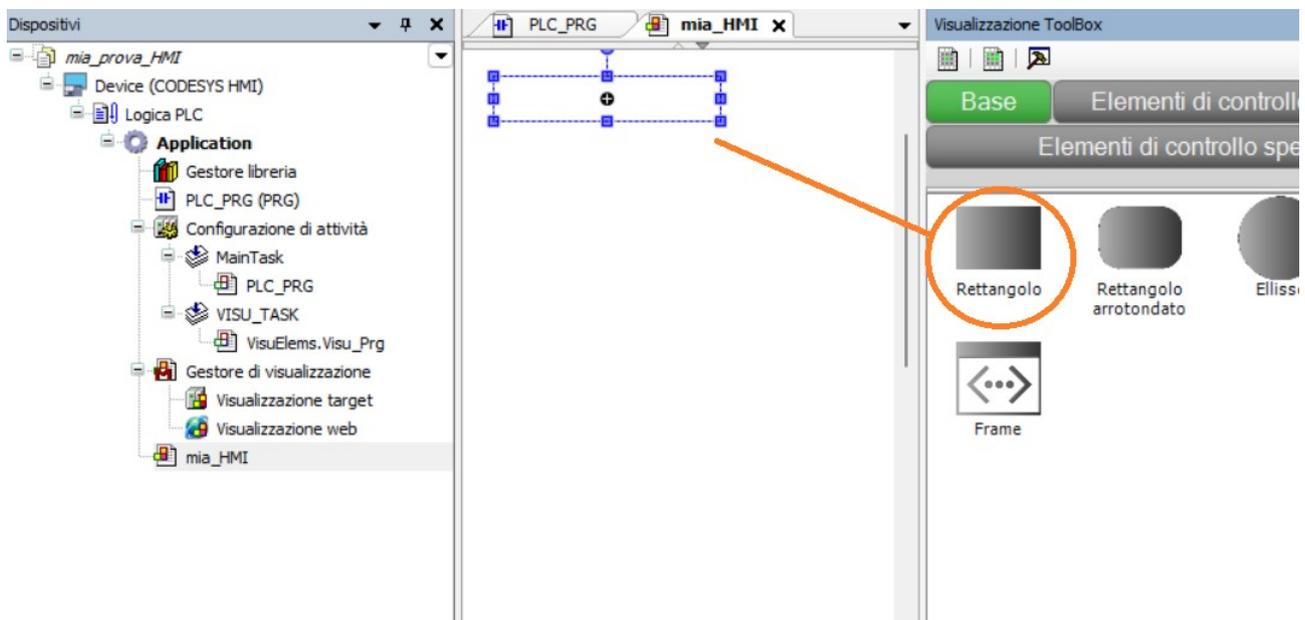
Diamo un nome alla visualizzazione, stando attenti, come al solito, a non inserire dentro al nome né spazi né caratteri speciali che non siano l'underscore “_”, poi premiamo “Aggiungi”.



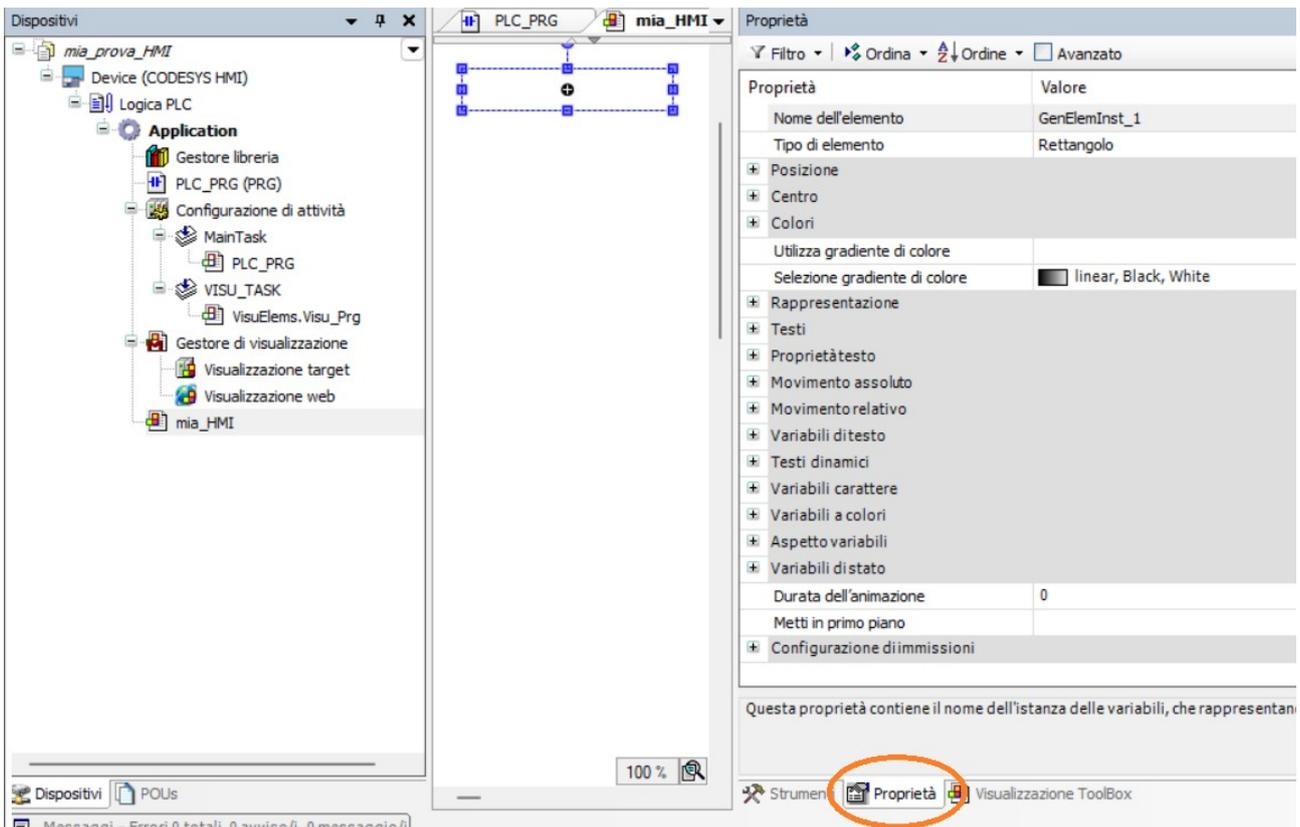
A questo punto, con l'HMI aperta, se nella finestra di destra si clicca su "Visualizzazione toolbox", compaiono una serie di voci relative ad oggetti grafici che possiamo inserire. Questi sono BASE; ELEMENTI DI CONTROLLO GENERALI etc



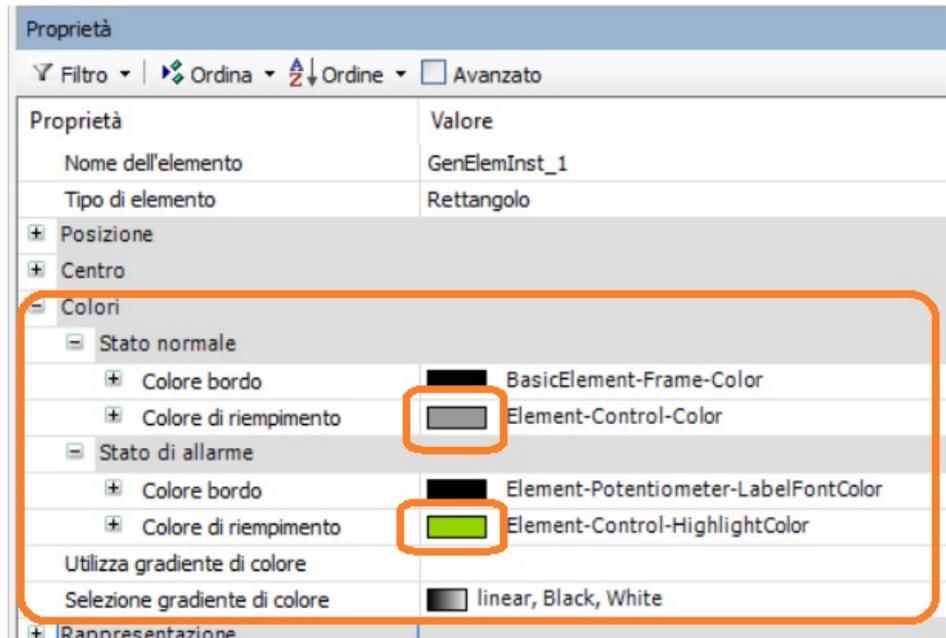
Tra gli elementi BASE, prendiamo un rettangolo e posizioniamolo sull'HMI



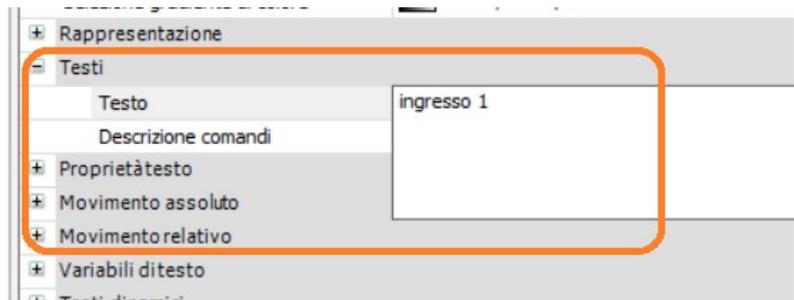
A destra compariranno tutte le sue proprietà, che ci consentiranno di collegarlo alle nostre variabili del PLC e a definirne il comportamento



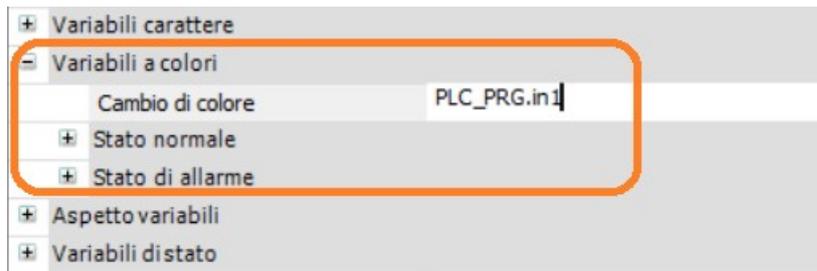
Ad esempio andiamo nella sezione Colori e scegliamo un colore grigio per il riempimento dello stato normale e un colore verde per il riempimento dello stato di allarme



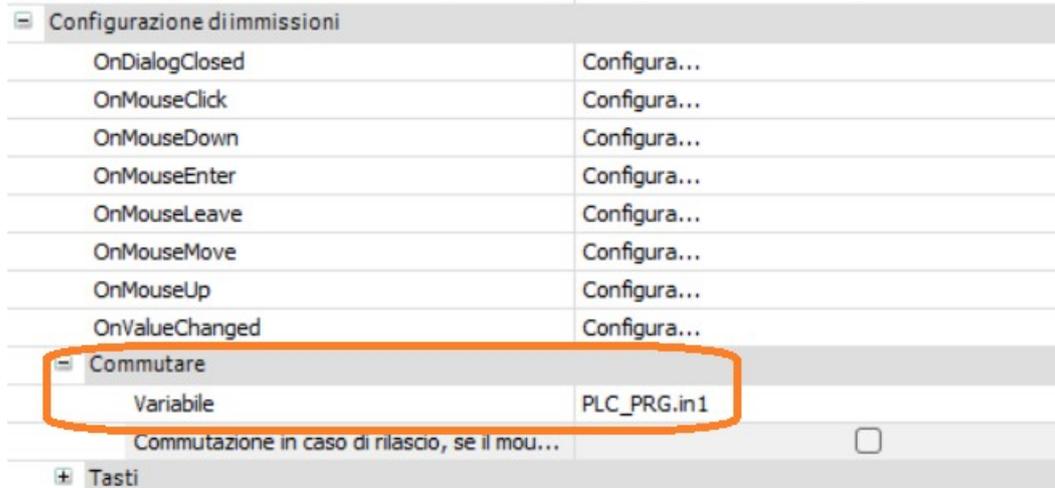
Poi andiamo nella sezione Testi e scriviamo "ingresso 1" (si ricorda che questo è un commento, non è il nome della variabile agganciata all'oggetto grafico)



Nella sezione “Variabili a colori”, dentro al campo “Cambio colore” inserire il nome della variabile locale in1, che è dichiarata dentro a PLC_PRG (quindi PLC_PRG.in1)



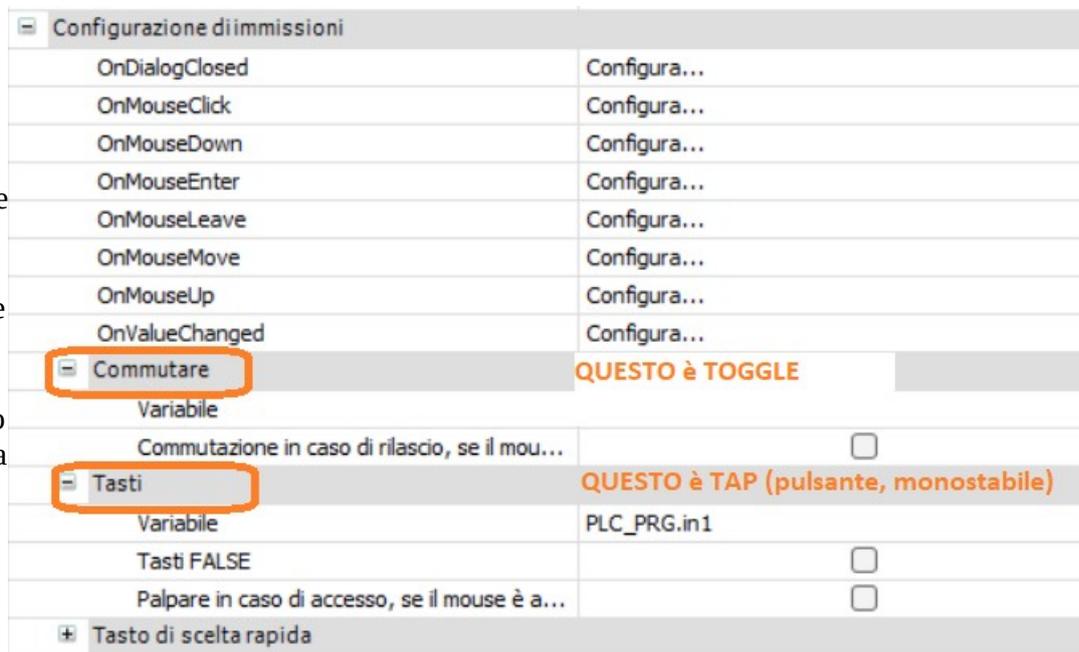
Nella sezione “Configurazione di immissioni”, dentro al sottocampo “Commutare”, scriviamo ancora il valore della variabile associata a questo oggetto grafico, cioè sempre PLC_PRG.in1



Per il momento la configurazione del rettangolo è conclusa.

NOTA BENE

La distinzione tra oggetto grafico di ingresso che si comporterà come un interruttore (toggle, bistabile) o come un pulsante (tap, monostabile) è relativa al campo in cui io scrivo la variabile nella “Configurazione di immissioni”. La traduzione dall’inglese all’italiano, in questo caso, è molto fuorviante.



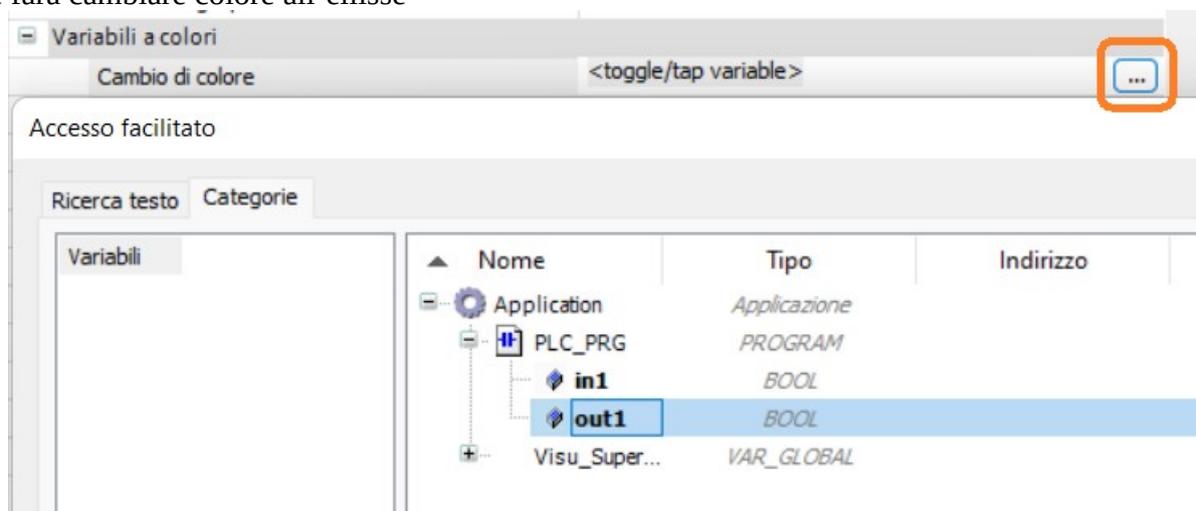
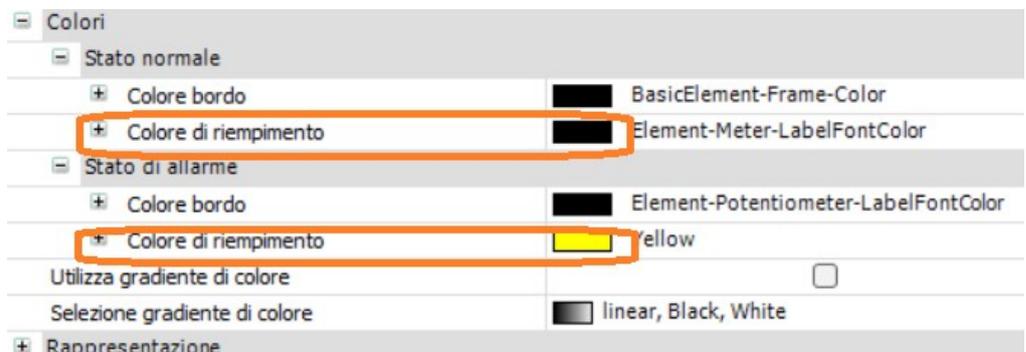
Questo “Configurazione di immissioni” sarebbe l’equivalente della sezione “Input” di Codesys 2.3. Se la variabile la metto nel campo *Commutare*, allora l’oggetto grafico sarà usato come toggle, se la metto nel campo *Tasti*, allora sarà usato come tap.

Tornando a noi, nell'HMI inseriamo anche un'ellisse (che è un altro degli elementi di BASE del Toolbox della visualizzazione)



La posizioniamo nella visualizzazione e poi la configuriamo. Scegliamo il colore nero come colore base dello sfondo e il giallo come colore di allarme del riempimento.

Poi mi sposto nel campo "Variabile a colori", nel sottocampo "Cambio di colore", vado sul pulsante con i puntini ..., clicco e mi si apre una finestra in cui scelgo la variabile out1, che sarà appunto quella che farà cambiare colore all'ellisse



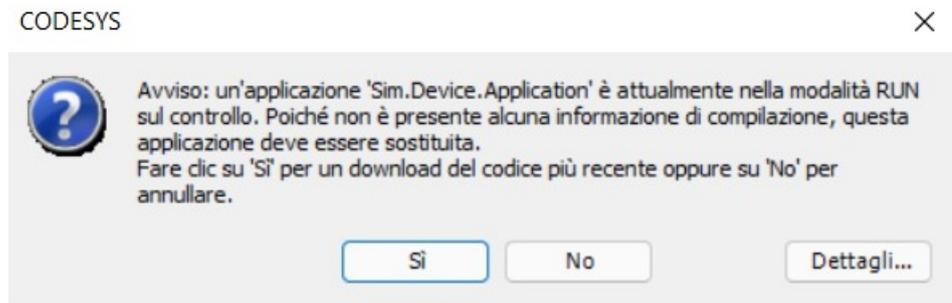
Così ottengo



A questo punto ho finito. Infatti l'ellisse rappresenta un'uscita, quindi il campo di configurazione dell'input (il famoso "Configurazione di immissioni") non va settato.

Devo quindi compilare il progetto, andare in simulazione, dare il login e mettere in RUN la cpu. Vado sul menù Compilazione → Crea Codice
Se non ci sono errori:

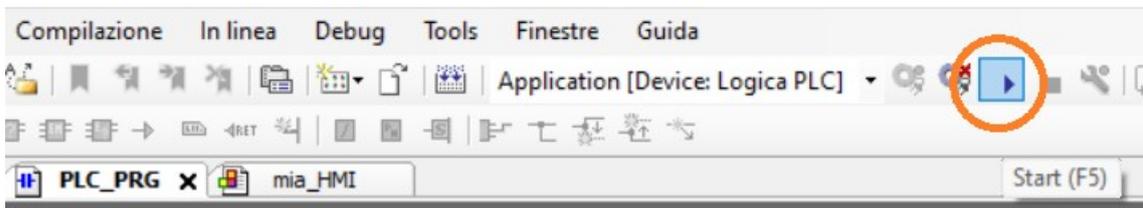
vado sul menù In linea → Simulazione
Poi sul menù In linea → Login
Se mi si apre questa
finestra clicco su Sì



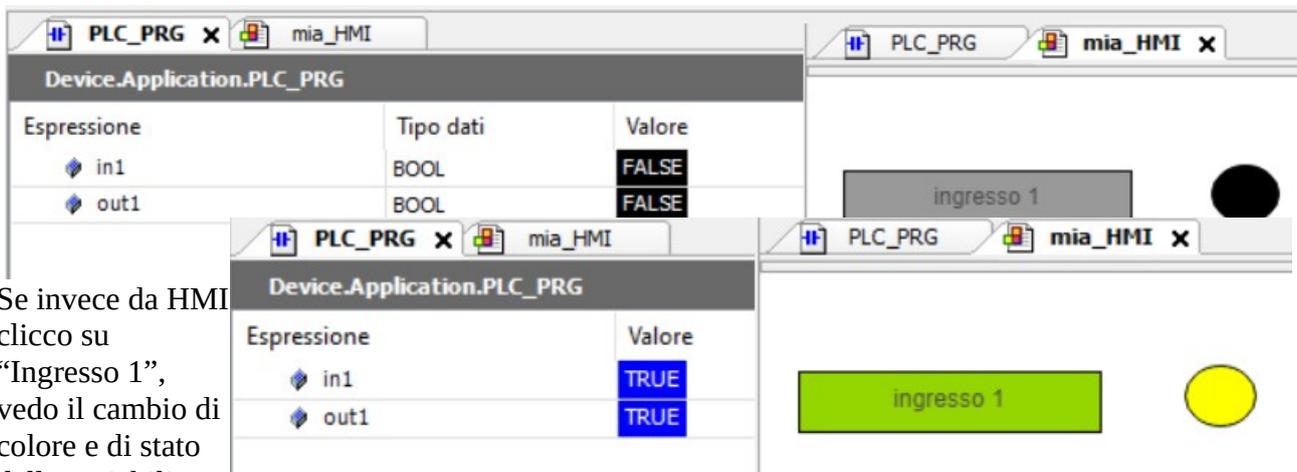
Può darsi che mi si visualizzi un messaggio del genere

La visualizzazione in linea è in attesa di una connessione. Reavvi l'applicazione.

Noi semplicemente diamo il RUN, al solito premendo sul tastino a forma di triangolo



A questo punto, se non tocco nulla, tutto è spento, e io vedo questo

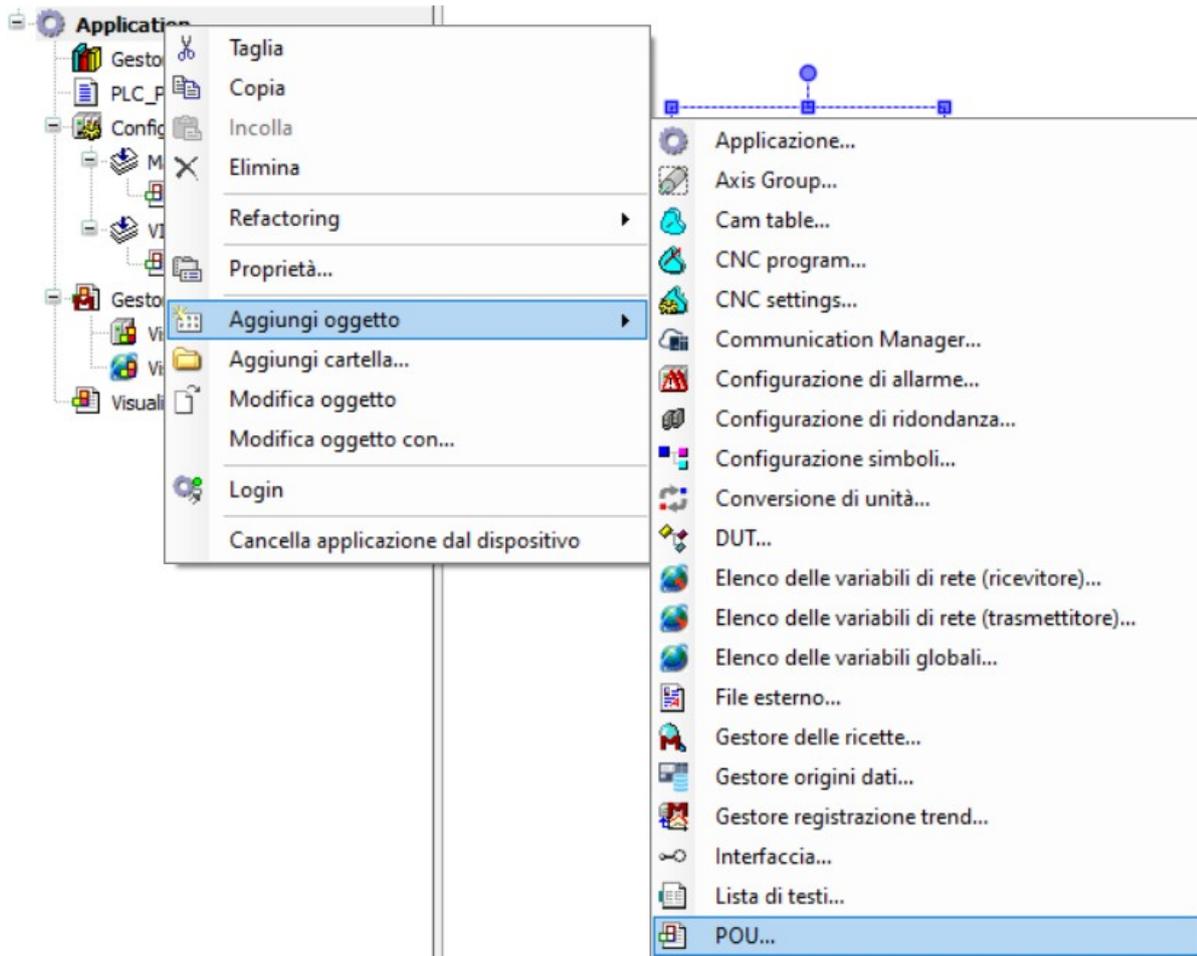


Se invece da HMI
clicco su
“Ingresso 1”,
vedo il cambio di
colore e di stato
delle variabili

booleane, con comportamento bistabile o monostabile a seconda di dove ho associato la variabile in1, se nel campo *Commutare* o nel campo *Tasti*

Far ruotare gli oggetti grafici, task, linguaggio ST

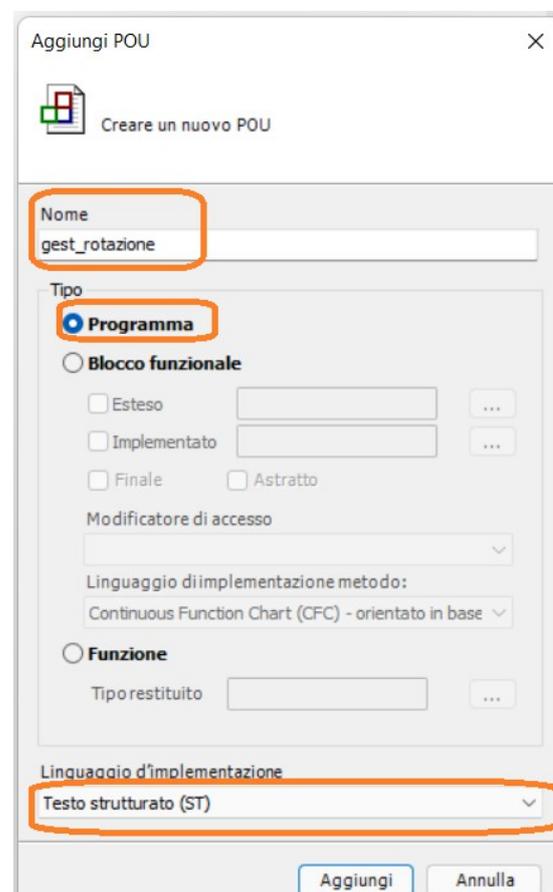
Supponiamo di scrivere una POU in testo strutturato.
Andiamo nel menù Application → Aggiungi oggetto → POU



Appare una finestra in cui scrivo il nome del programma “gest_rotazione” e in cui scelgo il linguaggio di programmazione (ST)

Poi clicco su “Aggiungi”

A questo punto si apre la finestra vuota di un programma che posso scrivere in ST.



Scrivo le istruzioni che seguono, con relativa dichiarazione di variabili locali alla POU appena creata.

```

1  PROGRAM gest_rotazione
2  VAR
3      motore : BOOL;
4      angolo : INT;
5  END_VAR
6
7  IF motore THEN
8      angolo := angolo +1;
9  END_IF

```

“angolo” è la variabile che userò per impostare il valore numerico dell’angolo di rotazione di un oggetto grafico da inserire nell’HMI.
 Ora vado nella Visualizzazione e disegno un rettangolo.

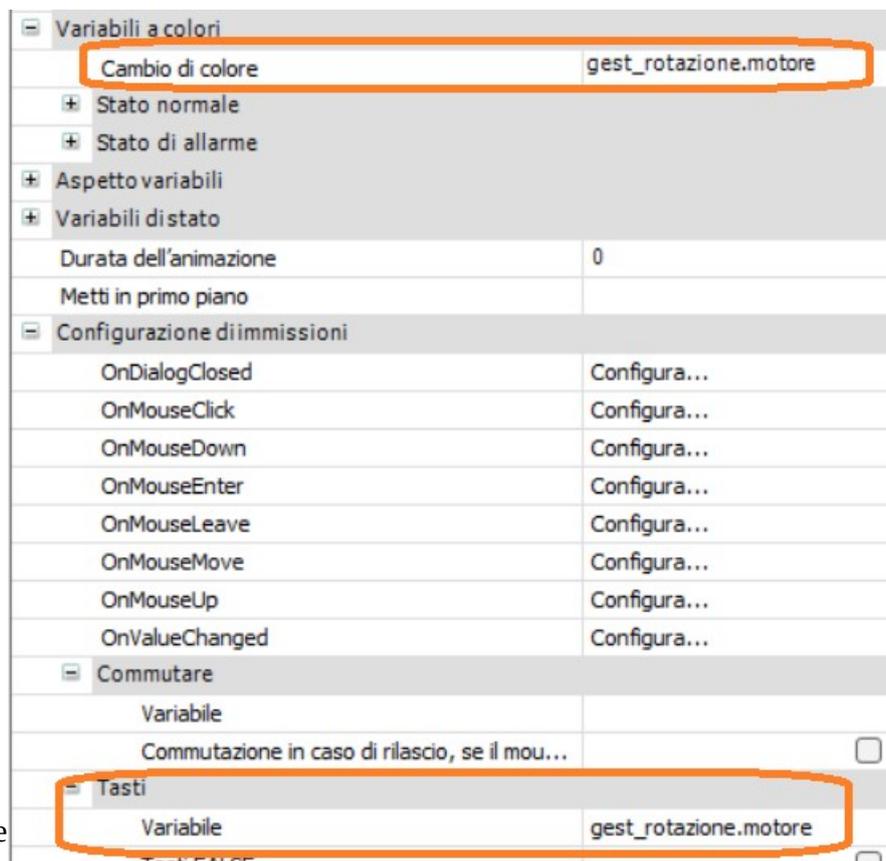
Nelle sue proprietà grafiche, visibili a destra dello schermo, scelgo il colore di allarme come rosso e nel campo “Movimento Assoluto” → “Rotazione Interna” inserisco la variabile “angolo” che è locale alla POU chiamata “gest_rotazione” (quindi devo scrivere gest_rotazione.angolo)

Proprietà

Filtro | Ordina | Ordine | Avanzato

Proprietà	Valore
Nome dell'elemento	GenElemInst_1
Tipo di elemento	Rettangolo
Posizione	
Centro	
Colori	
Stato normale	
Colore bordo	BasicElement-Frame-Color
Colore di riempimento	BasicElement-Fill-Color
Stato di allarme	
Colore bordo	BasicElement-Alarm-Frame-Color
Colore di riempimento	Element-Alarm-Frame-Color
Utilizza gradiente di colore	<input type="checkbox"/>
Selezione gradiente di colore	linear, Black, White
Rappresentazione	
Testi	
Testo	
Descrizione comandi	
Proprietà testo	
Movimento assoluto	
Movimento	
Rotazione	
Graduazione	
Rotazione interna	gest_rotazione.angolo
Utilizzo valori REAL	<input type="checkbox"/>

Rimanendo sempre nella finestra delle Proprietà, andando più sotto, imposto anche nel campo “Variabili a colori” la variabile “motore” della POU “gest_grafica” e nel campo “Tasti” sempre la stessa variabile.



Questo vuol dire che, se metto in esecuzione e clicco sul rettangolo, questo diventerà rosso e farà cambiare lo stato della variabile “motore”. Ma se motore diventa true, e fintanto che rimane TRUE (cioè finché tengo cliccato col mouse sul rettangolo) allora il programma gest_grafica fa incrementare la variabile angolo a passi di 1. Ma se la variabile angolo si incrementa, allora il rettangolo ruoterà su se stesso.

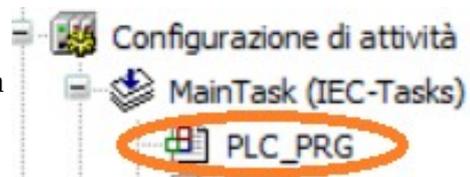
Togliendo il mouse dal rettangolo, invece, esso cesserà di ruotare.

A questo punto siamo quasi pronti.

Manca un passaggio.

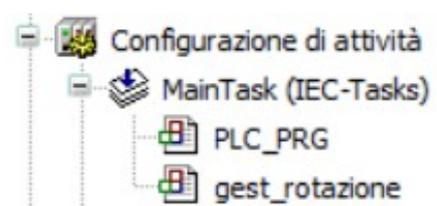
Abbiamo creato una POU nuova ma non abbiamo detto a Codesys come mandarla in esecuzione.

Si noterà che la POU col nome di default, ovvero “PLC_PRG” è già automaticamente agganciata ad un task, senza che io abbia dovuto fare altro.



Per collegare a questo task anche la nuova POU creata, mi basta andare sul suo nome, tenere cliccato e “trascinarla” dentro alla voce “MainTask (IEC-Tasks)”

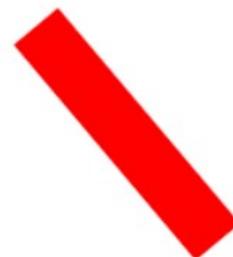
A questo punto ottengo:



Adesso ho collegato tutto.

Posso compilare, andare in simulazione, andare online e mettere in run.

Cliccando sul rettangolo, questo diventa rosso e inizia a ruotare.



Creare un array di struct, usando i tipi di dato definiti dall'utente (DUT = Dati Utente)

Creiamo un nuovo progetto. Vogliamo creare un array di variabili, ognuna delle quali faccia riferimento ad un oggetto grafico con diversa posizione orizzontale e diversa visibilità (alcuni oggetti visibili e alcuni invisibili).

Ognuna di queste variabili sarà quindi strutturata in 2 campi: il campo x (la quota orizzontale) e il campo "visibile" (la sua visibilità).

La variabile è quindi di tipo "struct", che è un tipo definito dall'utente.

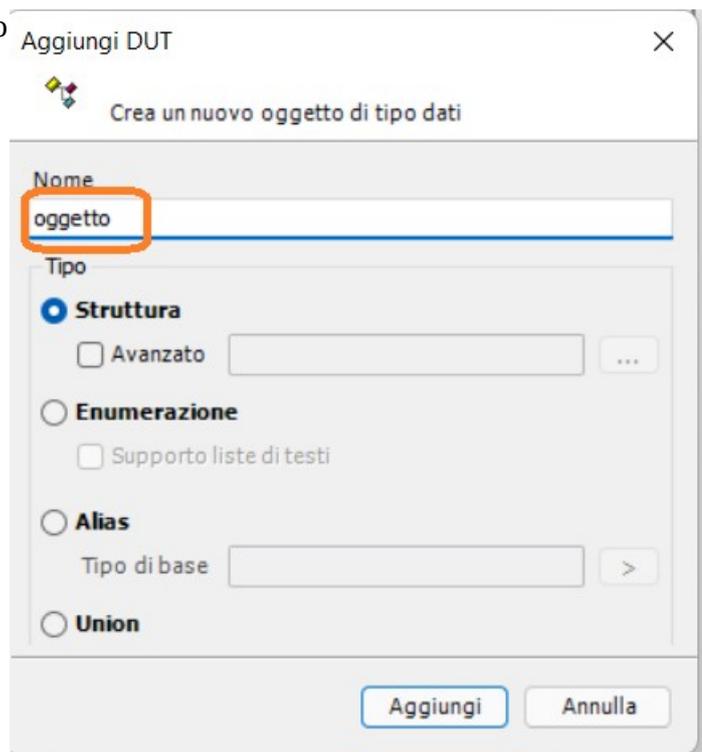
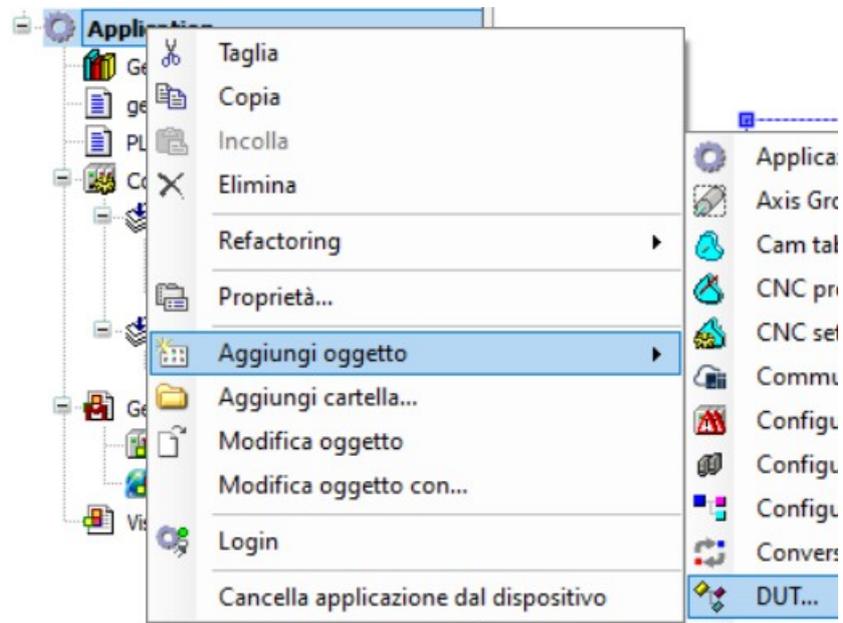
Per definire uno struct devo andare in "Application" → "Aggiungi oggetto" → "DUT"

Chiamare la variabile "oggetto", scegliendo Struttura, e cliccare su "Aggiungi"

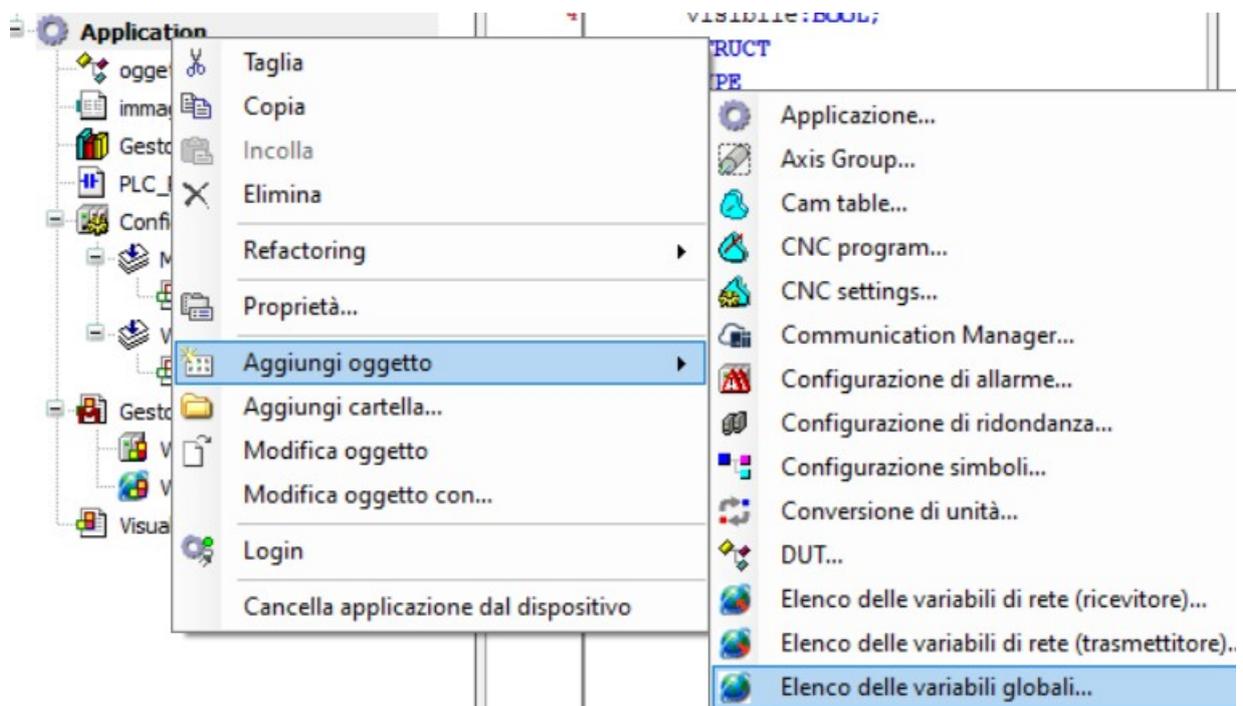
Si apre una finestra precompilata con varie scritte (TYPE...STRUCT...) in cui posso definire i campi della mia variabile "oggetto".

Scrivo come segue qui sotto:

```
1  TYPE oggetto :
2  STRUCT
3      x:INT;
4      visibile:BOOL;
5  END_STRUCT
6  END_TYPE
7
```



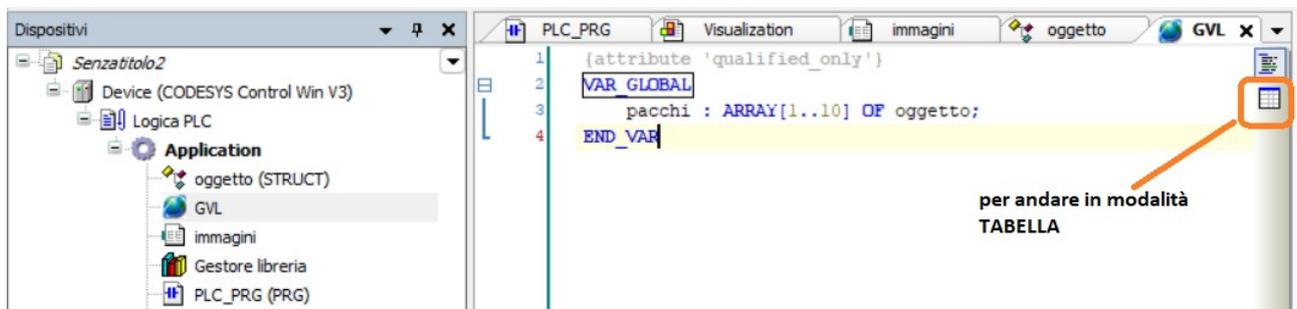
Adesso facciamo di nuovo click dx su Application → Aggiungi oggetto → Elenco delle variabili globali



Nella finestra che si apre, lasciamo il nome esistente di default, cioè GVL, e clicchiamo su “Aggiungi”.

Ora, nella finestra di sinistra, dentro all’applicazione si è creata anche la voce “GVL”.

Se la selezioniamo, si apre la finestra delle variabili globali, dove, a mano, possiamo inserire la seguente scrittura che ci crea un array chiamato “pacchi”, di 10 elementi, ognuno contenente uno struct di tipo “oggetto”, come da noi creato.

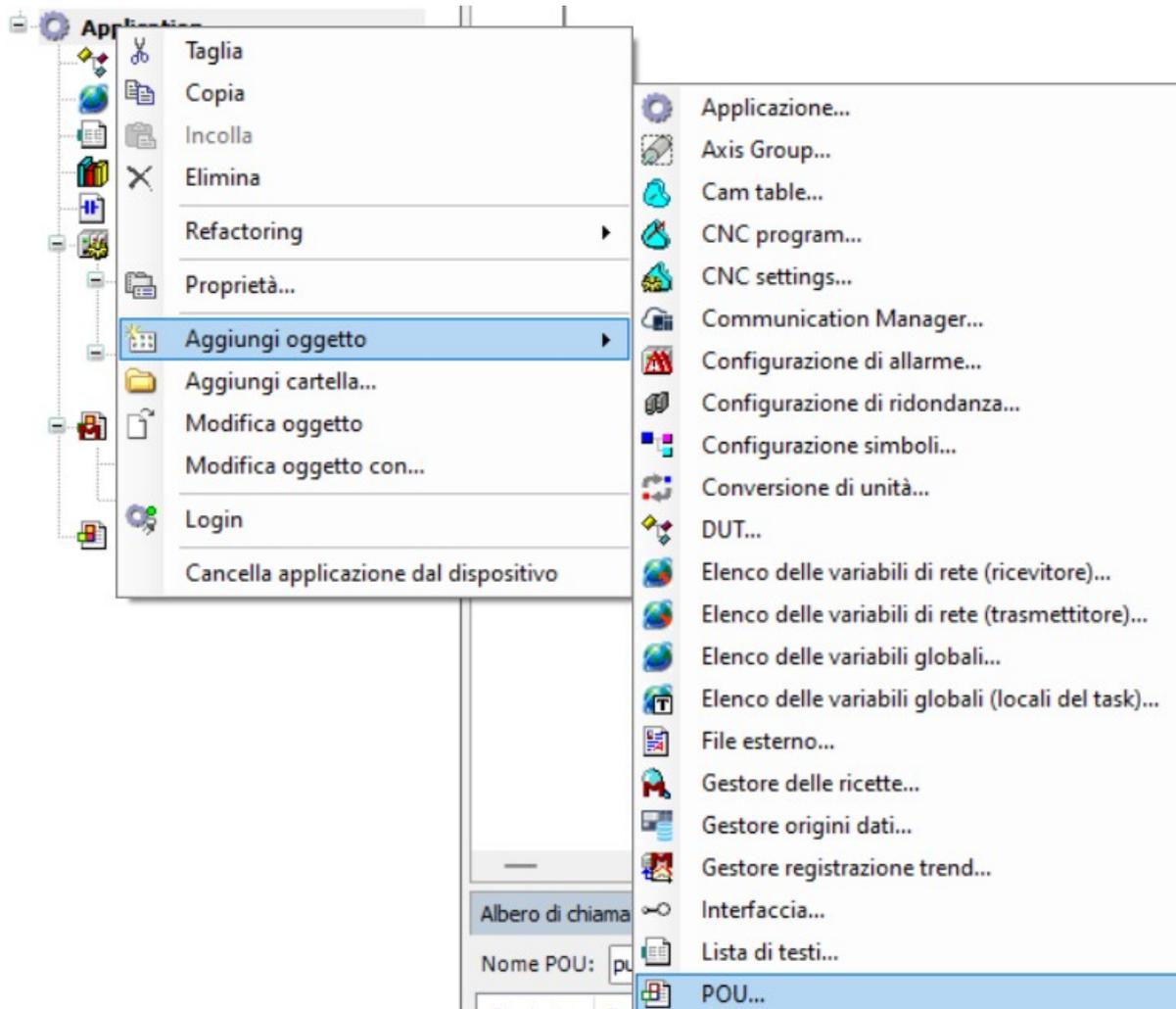


In versione tabellare è così

	Ambito di validità	Nome	Indirizzo	Tipo dati	Inizializzazione	Commento	Attril
1	VAR_GLOBAL	pacchi		ARRAY[1..10] OF oggetto			

Ora creiamo una seconda POU, di nome “grafica”, in ST

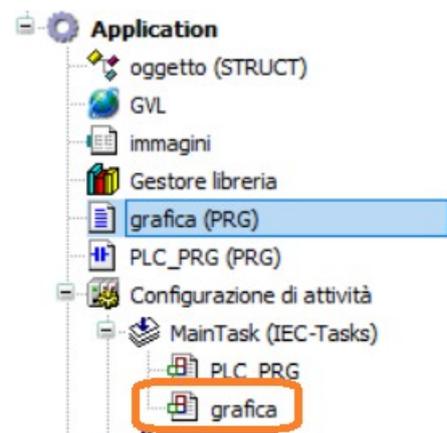
Quindi clicchiamo dx col mouse su Application poi selezioniamo “Aggiungi oggetto” → “POU...”



Creiamo la nuova POU in ST e poi clicchiamo su Aggiungi.

The image shows the 'Aggiungi POU' dialog box. It has a title bar with 'Aggiungi POU' and a close button. Below the title bar is a button labeled 'Creare un nuovo POU'. The main area contains a text field for 'Nome' with the value 'grafica'. Below this is a 'Tipo' section with two radio buttons: 'Programma' (selected) and 'Blocco funzionale'. Under 'Blocco funzionale', there are three checkboxes: 'Esteso', 'Implementato', and 'Finale', each with a text field and a button. There is also an 'Astratto' checkbox. Below these is a 'Modificatore di accesso' dropdown menu. The 'Linguaggio di implementazione metodo:' dropdown menu is set to 'Testo strutturato (ST)'. Under the 'Funzione' radio button, there is a 'Tipo restituito' text field and a button. At the bottom, there is a 'Linguaggio d'implementazione' dropdown menu set to 'Testo strutturato (ST)'. At the very bottom are two buttons: 'Aggiungi' and 'Annulla'.

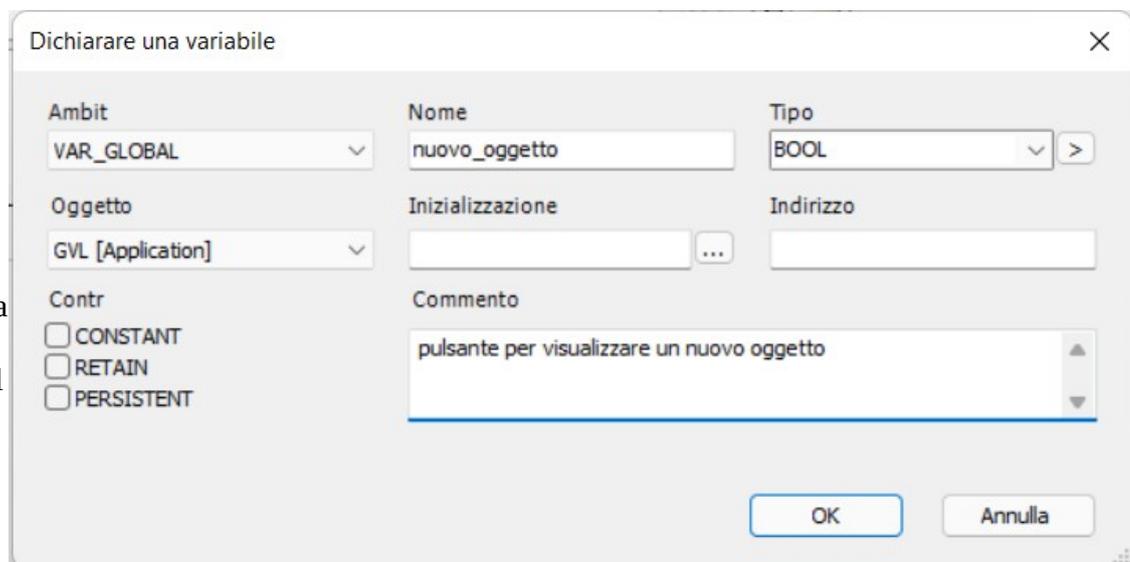
Trasciniamo la nuova POU dentro al Main Task, così che entrambe le POU siano richiamate dal task.



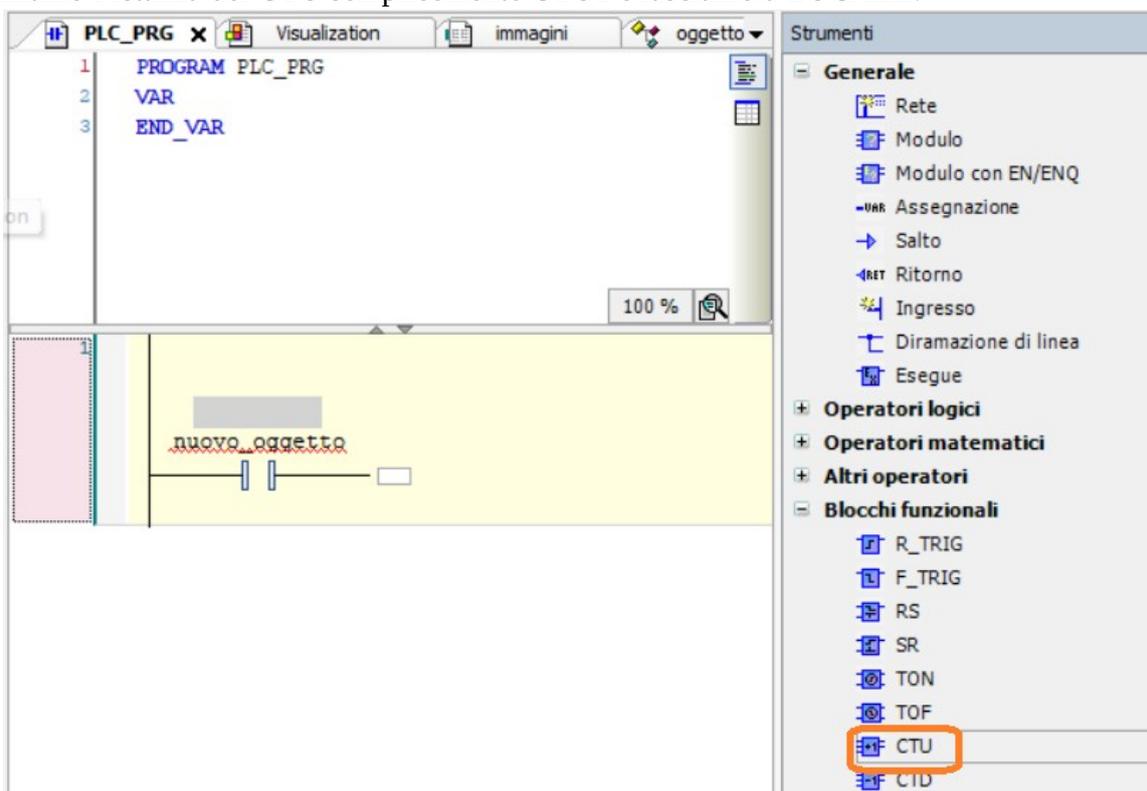
Torniamo in PLC_PRG e creiamo una riga di ladder in cui il contatto denominato “nuovo_oggetto”, associato ad una variabile globale, entra nell’ input di un function block di tipo CTU.

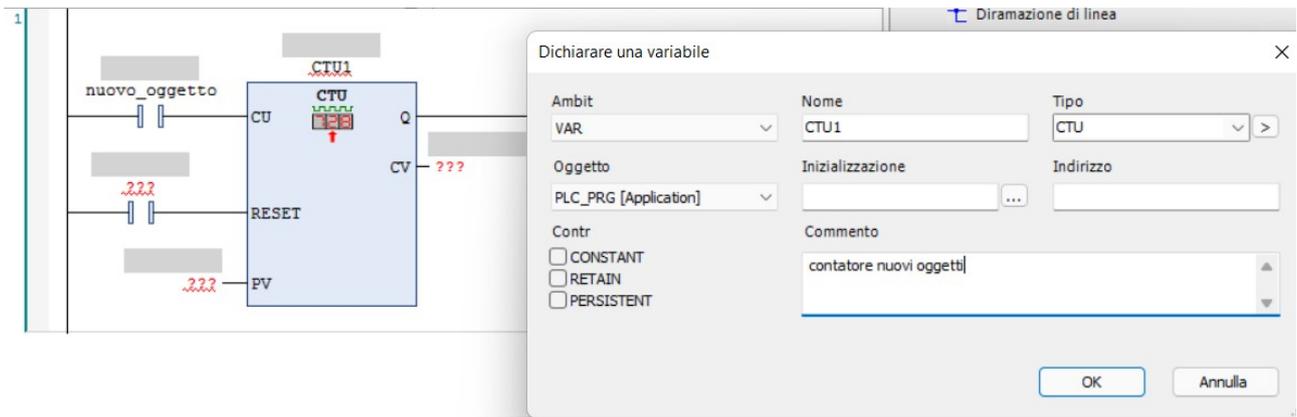
Metto quindi un contatto e lo dichiaro in questo modo

Poi, tra gli strumenti, a destra, seleziono il CTU e lo trascino nella riga di ladder.

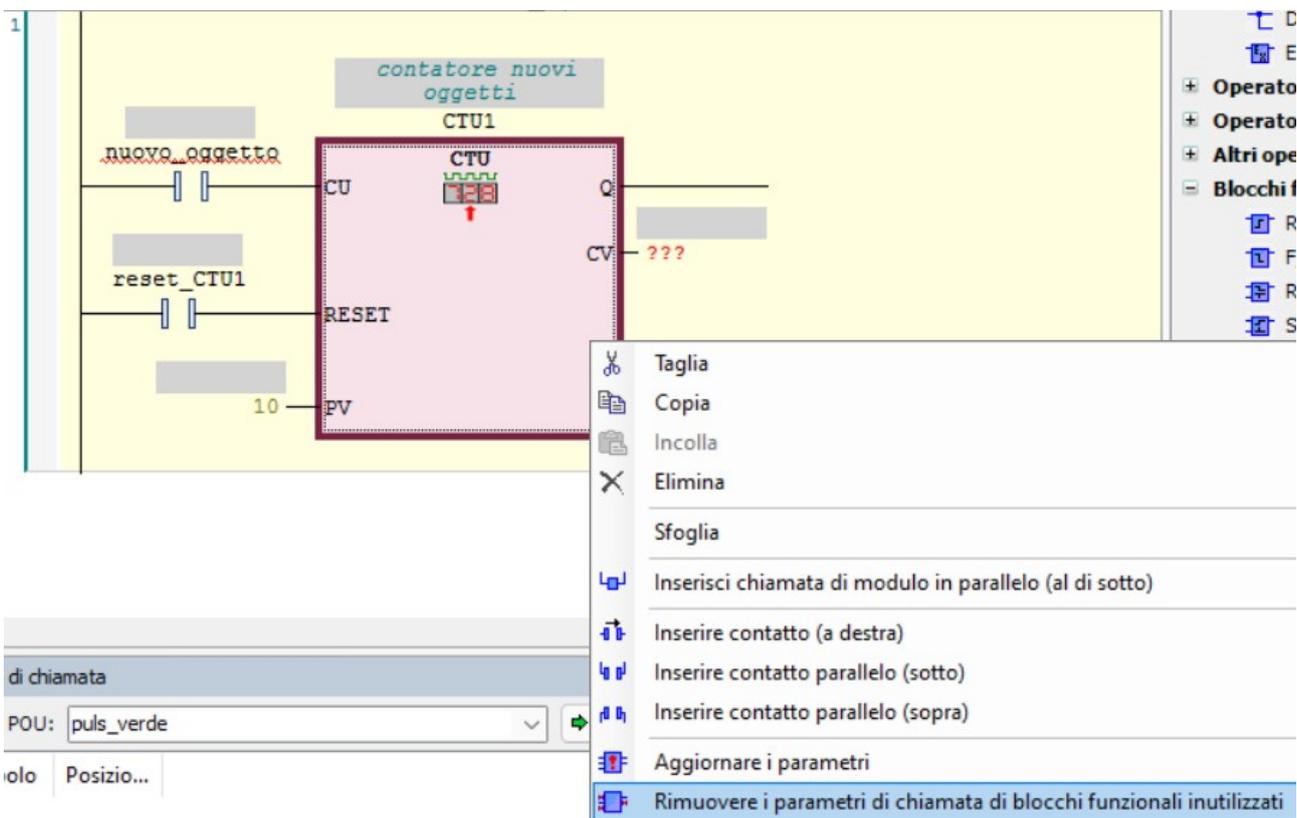


Chiamiamo l’istanza del CTU semplicemente CTU1 e lasciamola LOCALE.

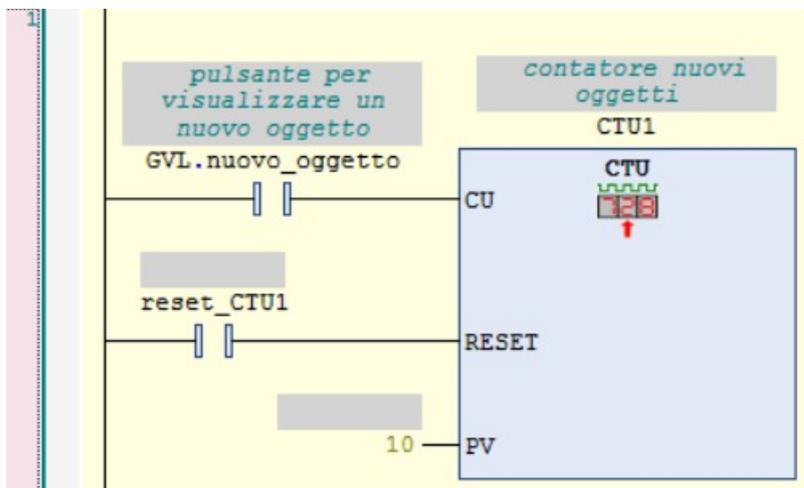




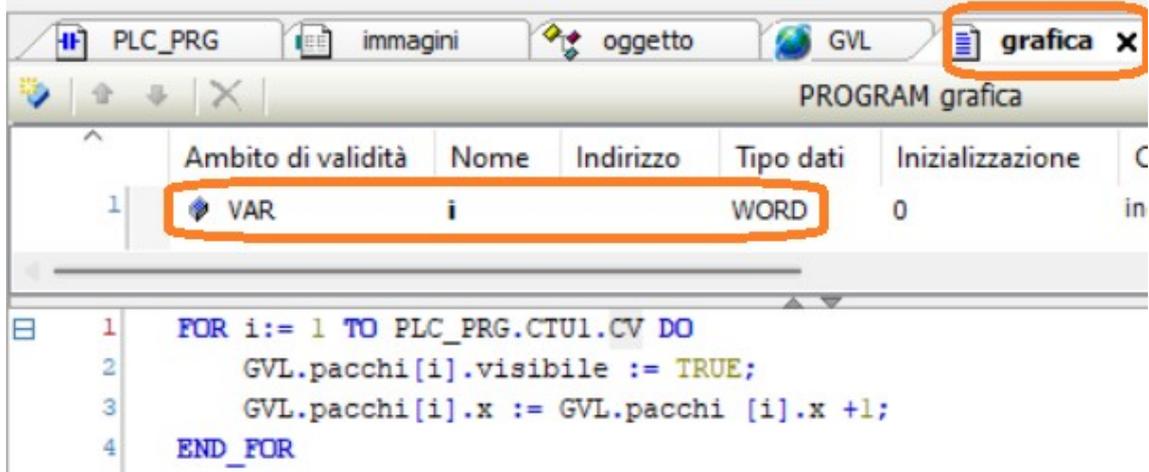
Mettiamo come PV (Preset value) il valore 10, aggiungiamo il contatto “reset_CTU1” sull’ingresso di RESET (dichiarando reset_CTU1 come un booleano locale) e poi clicchiamo col dx sul blocco e selezioniamo la voce “Rimuovere i parametri...” per non avere errori sull’uscita CV (Counted value), dato che non la useremo.



Infine il contatto su cui abbiamo scritto “nuovo_oggetto” ovviamente va rinominato “GVL.nuovo_oggetto”, essendo la variabile globale.

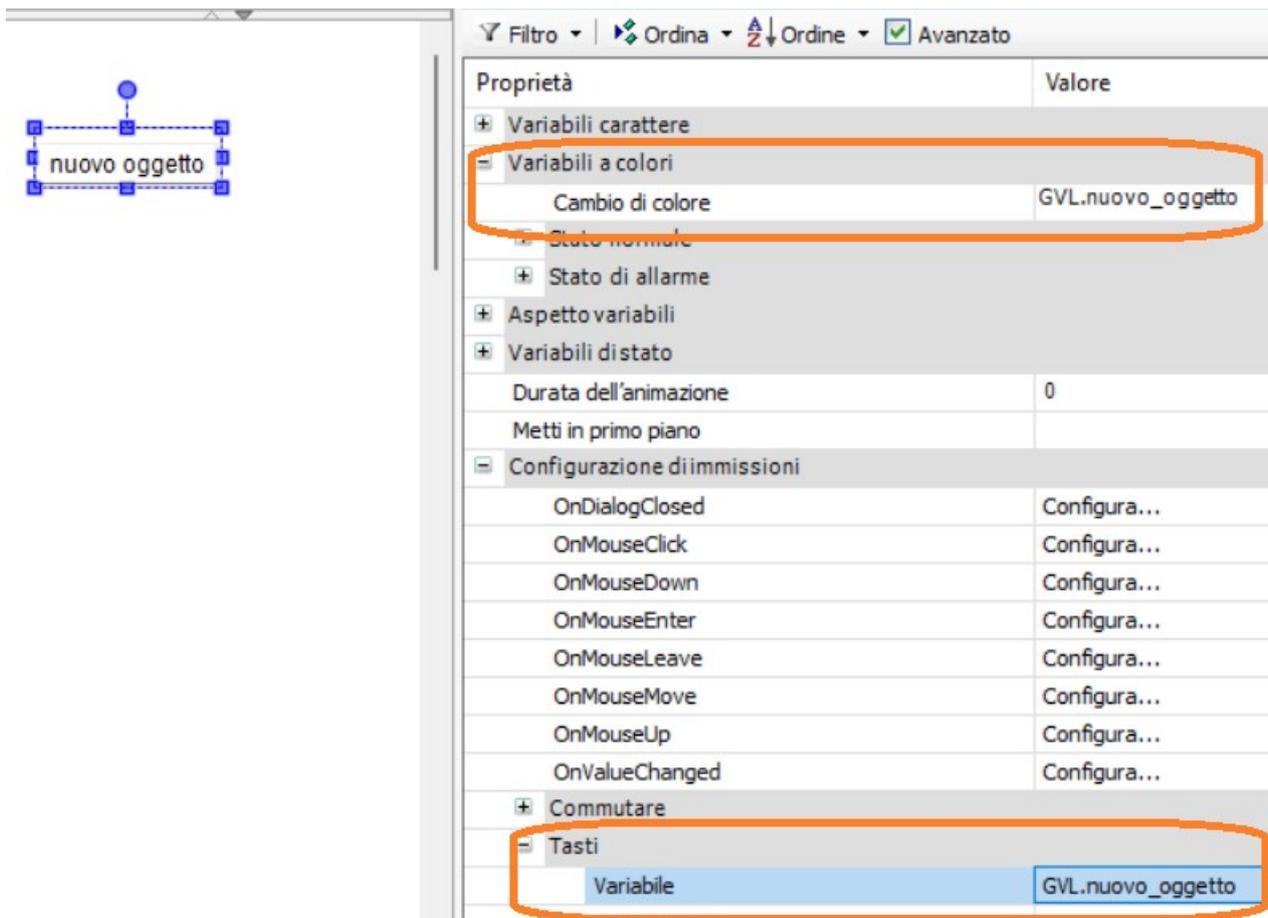


Poi, nell'altra POU, quella in ST, scriviamo quanto segue, dichiarando la variabile "i" come una WORD locale.:



A questo punto creiamo una Visualizzazione, andando su "Application" → "Aggiunge oggetto" → "Visualizzazione".

Trasciniamo sulla visualizzazione un comune rettangolo, diamogli il testo "nuovo oggetto", un colore di allarme, associamo la variabile di cambio colore a "GVL.nuovo_oggetto" e l'opzione "Configurazione di immissioni" → "Tasti" sempre alla variabile "GVL.nuovo_oggetto".



Creiamo un secondo rettangolo, che rappresenterà il primo pacco.
Diamogli come testo "pacco1".

Andiamo, tra le sue proprietà grafiche, nella sezione “Movimento assoluto” → “Movimento” → “x” e scriviamogli “GVL.pacchi[1].x”

Movimento assoluto	
Movimento	
X	GVL.pacchi[1].x
Y	

Andiamo nella sezione “Variabili di stato “ → “Invisibilità” e scriviamogli “NOT GVL.pacchi[1].visibile”

Variabili di stato	
Invisibilità	NOT GVL.pacchi[1].visibile
Disattiva immissioni	

Copiamo e incolliamo il rettangolo altre 9 volte, e per tutti i nuovi rettangoli, cui daremo il nome “pacco2”, “pacco3” etc, modifichiamo sempre l’indice racchiuso tra le parentesi quadre, in modo da avere ogni rettangolo associato alla quota in x e alla visibilità di un elemento del vettore.

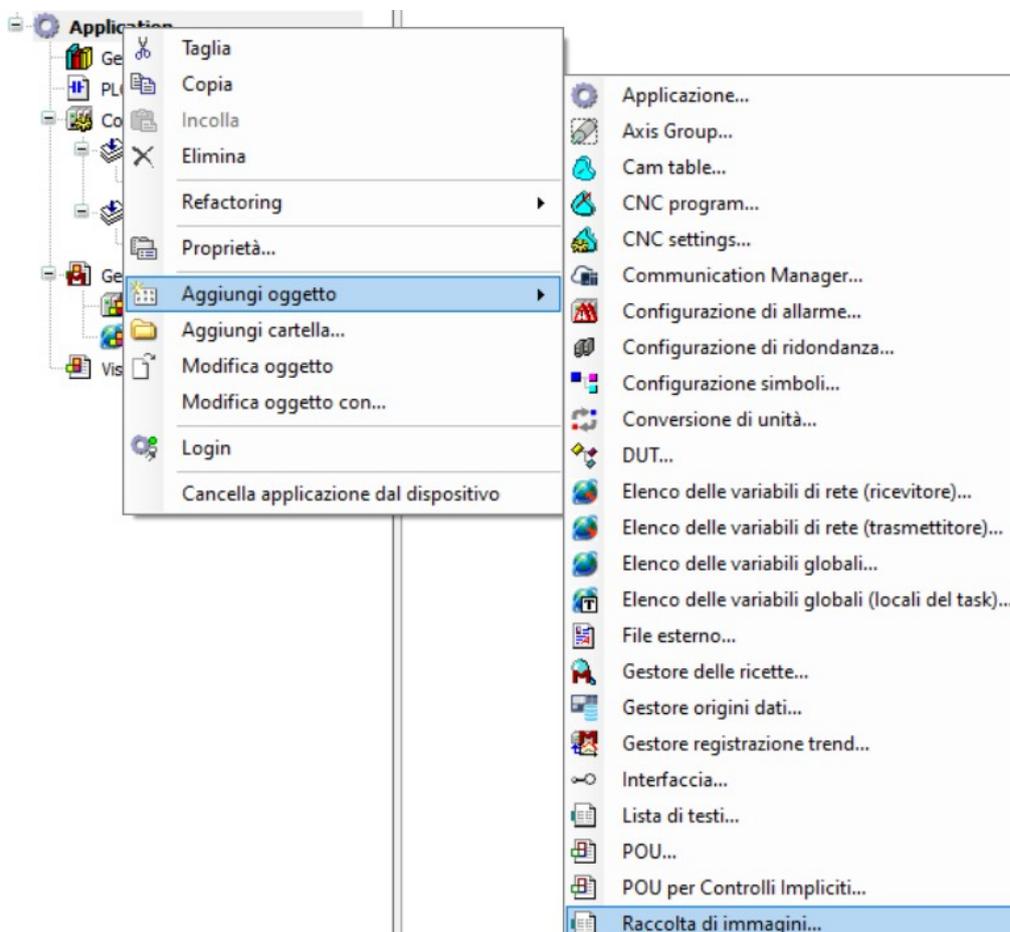
Quando abbiamo finito andiamo in Simulazione, compiliamo, diamo il Login e mettiamo in run. Alla partenza si vede solo il pulsante “nuovo oggetto”.

Se ci clicco su una volta, appare il primo rettangolo e inizia a muoversi verso destra, se clicco altre volte compaiono gli altri rettangoli ed iniziano a muoversi verso destra. Ad esempio, qui di fianco, il pulsante “nuovo oggetto” è stato premuto 3 volte, quindi vedo tre rettangoli che si muovono.

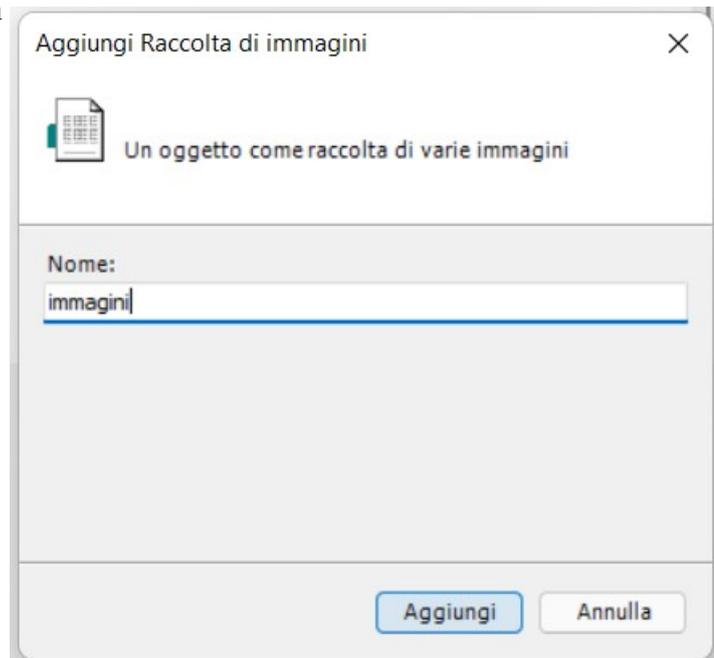


Inserire un’immagine esterna nella grafica

Per inserire immagini esterne, ad esempio nostri file .jpg o .bmp, la strada da percorrere è simile a quella usata in altri ambienti di sviluppo come Visual Studio, ovvero prima bisogna inserire le immagini nel progetto, e poi esse si possono usare nella visualizzazione. Click dx su “Application” → “aggiungi oggetto” → “raccolta di immagini” (in inglese Image Pool)



Si apre una finestra in cui chiede di dare un nome al set delle immagini. Chiamiamolo, con estrema fantasia, “immagini” e clicchiamo su Aggiungi.



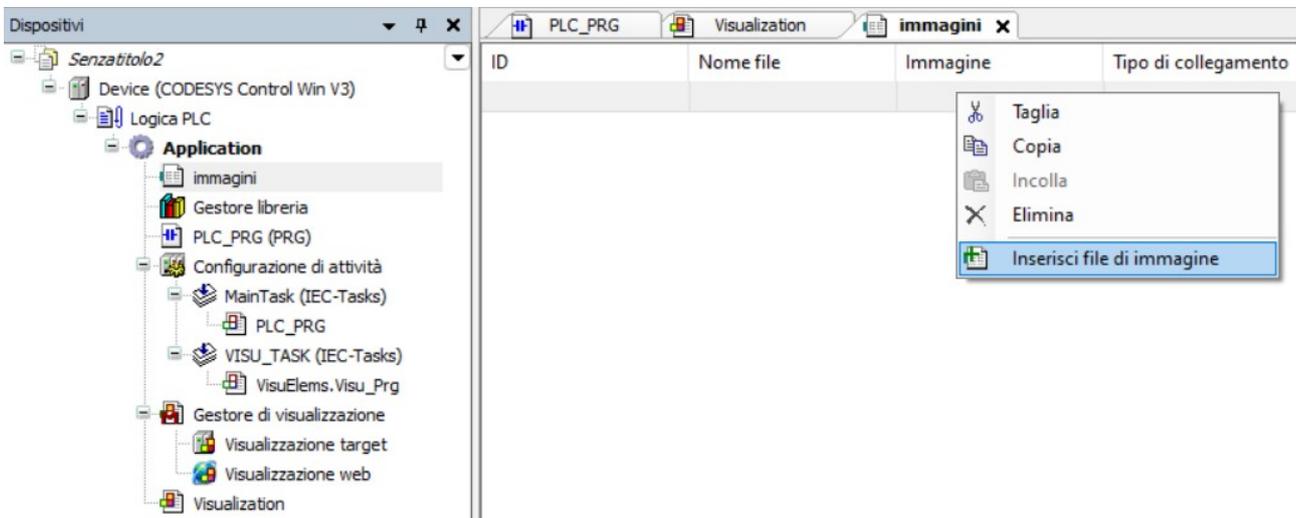
A questo punto si apre la visualizzazione del set delle immagini che possiamo voler includere nella nostra visualizzazione. Dobbiamo popolarla con i nostri file, presenti nel nostro pc.

Clicchiamo col dx del mouse sulla sezione vuota sotto alla riga

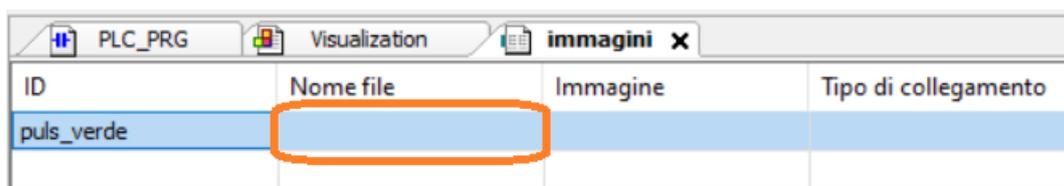
“ID Nome File Immagine Tipo di collegamento”

e selezioniamo “Inserisci file di immagine”.

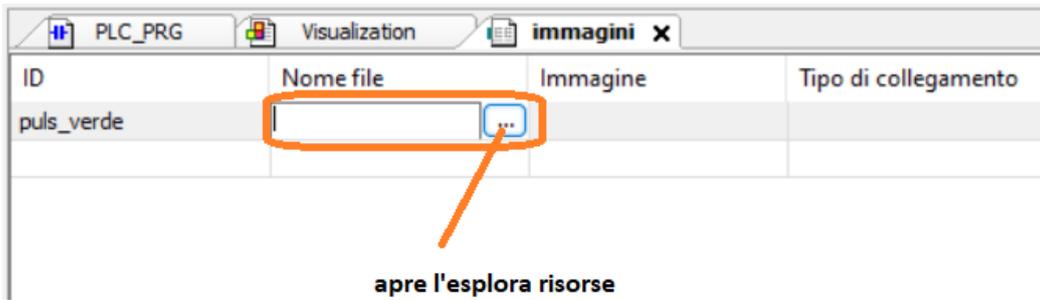
Diano un nostro ID all’immagine. Ad esempio io do il nome “puls_verde”, perché ho intenzione di inserire una mia immagine di pulsante, appunto di colore verde.



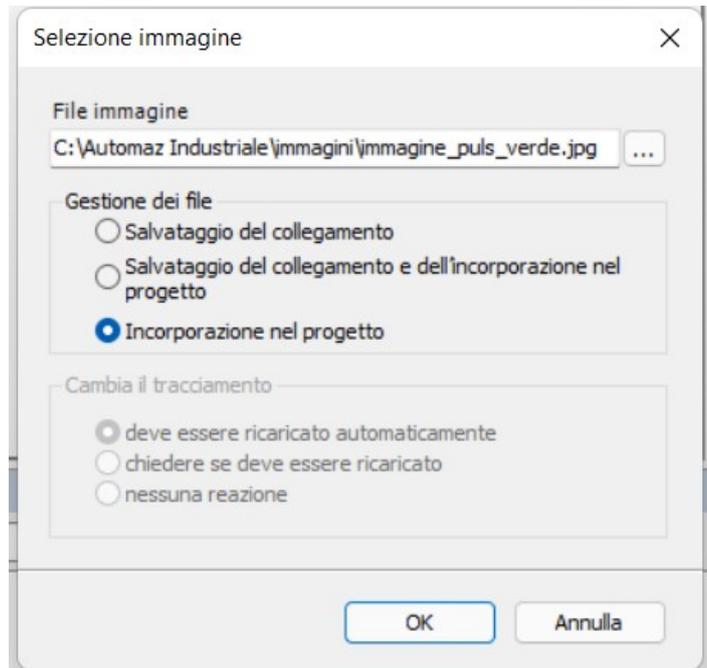
Poi clicchiamo due volte nella zona SOTTO alla voce “Nome file”.



Si apre un pulsantino con i puntini, da dove riusciamo ad accedere all’explora risorse per trovare, sul nostro pc, l’immagine che vogliamo includere.



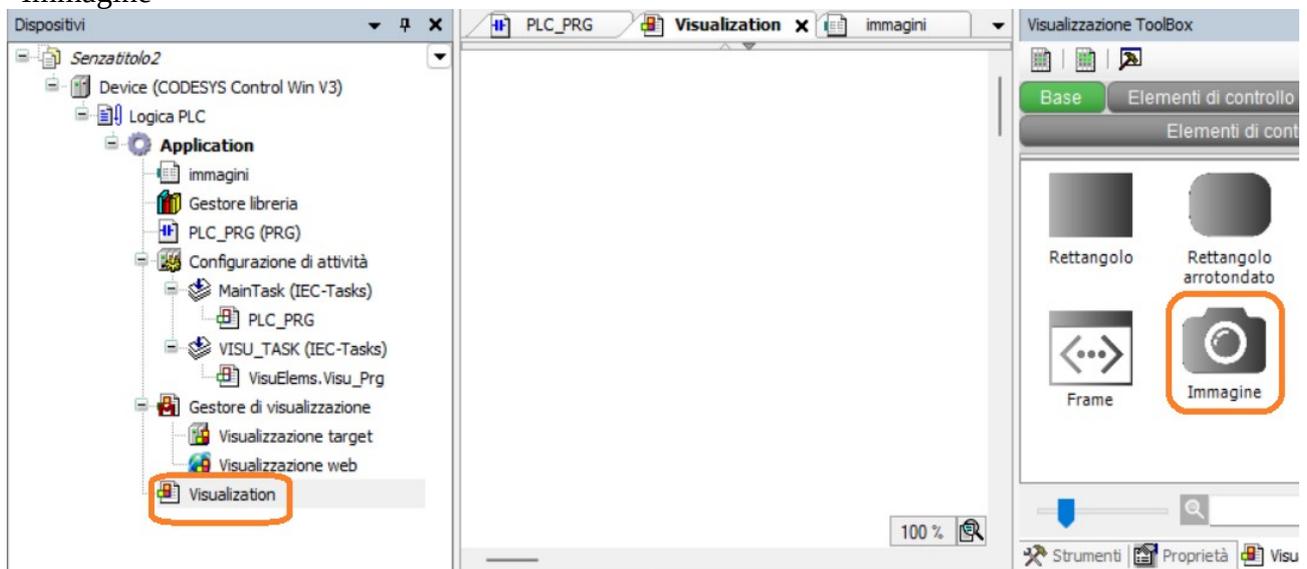
Selezioniamo il file immagine e spuntiamo la voce “Incorporazione nel progetto” (in inglese “Embed into project”). In questo modo il file immagine diventa parte del nostro progetto software.



Il tutto diventa quindi

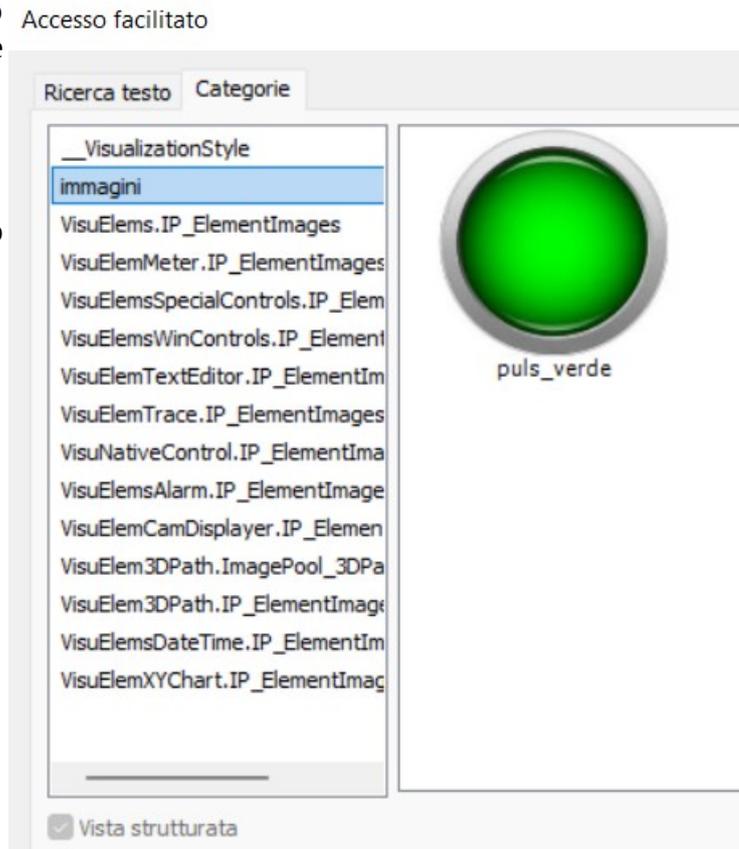


Adesso torniamo nella Visualizzazione, e tra gli strumenti di destra selezioniamo “Base” → “Immagine”

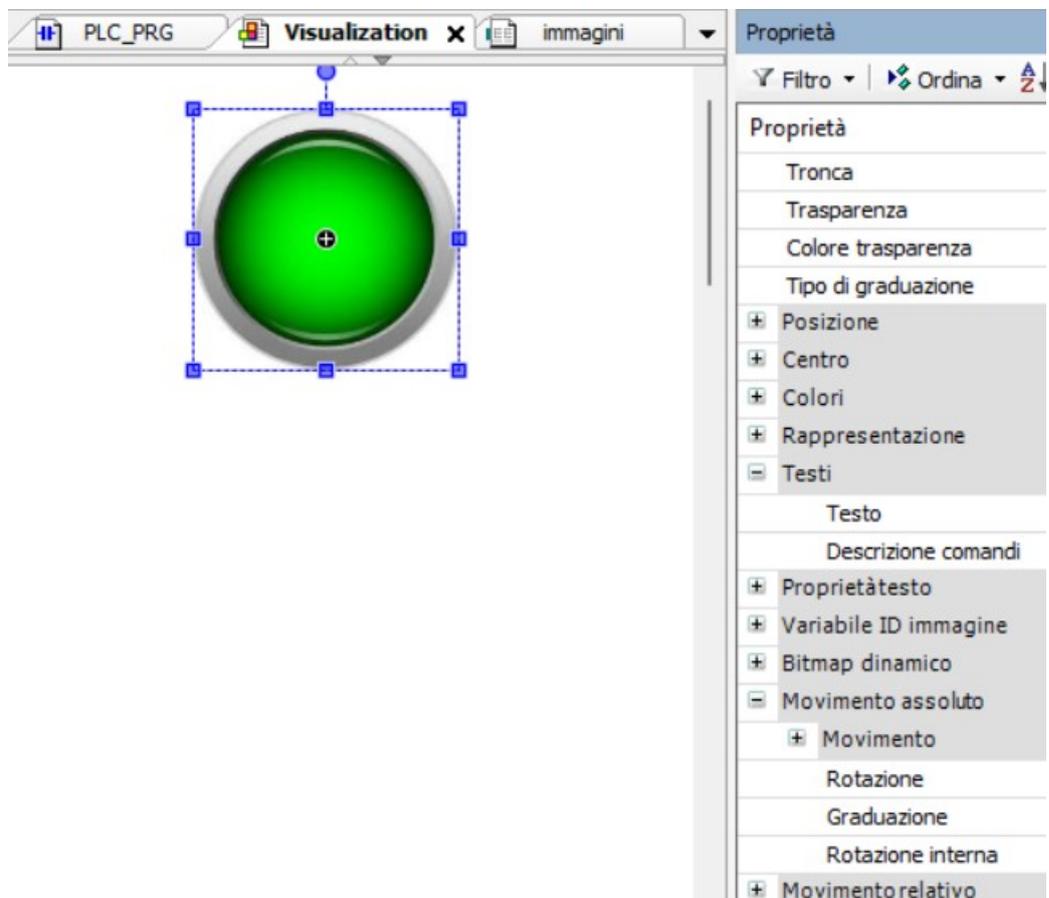


Non appena, col Drag&Drop, rilasciamo l'immagine sulla visualizzazione, si apre la finestra dell'accesso facilitato, dove una delle opzioni di sinistra dovrebbe essere proprio il nostro set di immagini (nel quale, per il momento, abbiamo inserito un'immagine sola. Ma possiamo inserire tutte quelle che vogliamo, ovviamente).

Selezioniamo l'immagine che abbiamo inserito (che da me è il pulsante verde), diamo l'OK e poi posizioniamo e scaliamo l'immagine come vogliamo.



A destra, nelle sue proprietà, posso impostare movimenti, click e altro, come per gli elementi grafici già presenti nelle librerie di Codesys.



Se voglio, posso anche inserire un'immagine di sfondo (background).
Clicco col dx del mouse sullo sfondo bianco della visualizzazione e selezione l'opzione "Sfondo".
Poi mi si apre una finestra in cui posso selezionare l'immagine.

