

TUTORIAL KREO HMI Integrazione con Codesys

Tutorial dedicato alla integrazione tra KREO runtime ed il runtime Codesys 3.5

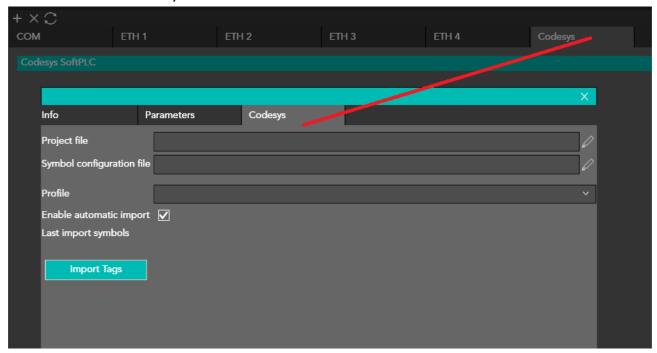
Connect Ideas.
Shape solutions.



Introduzione

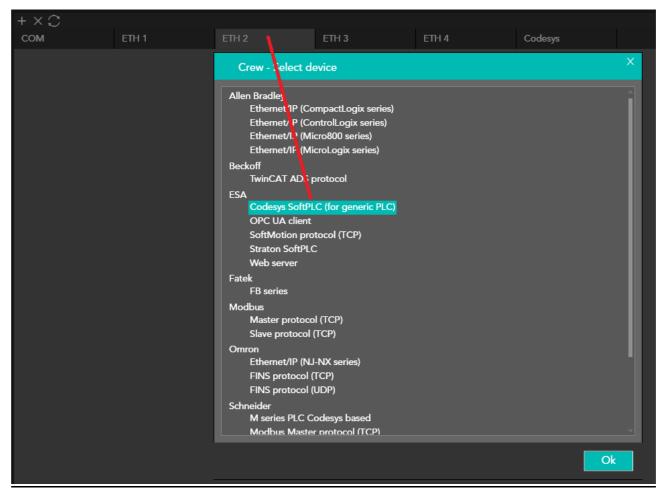
KREO supporta il driver CODESYS. La comunicazione in tal senso è gestita in 2 modalità e 2 driver diversi:

 CODESYS SOFT PLC (il driver per comunicare con il soft-plc CODESYS, sia esso quello integrato all'interno dei pannelli HMI sia quello per PC Control Win V3).



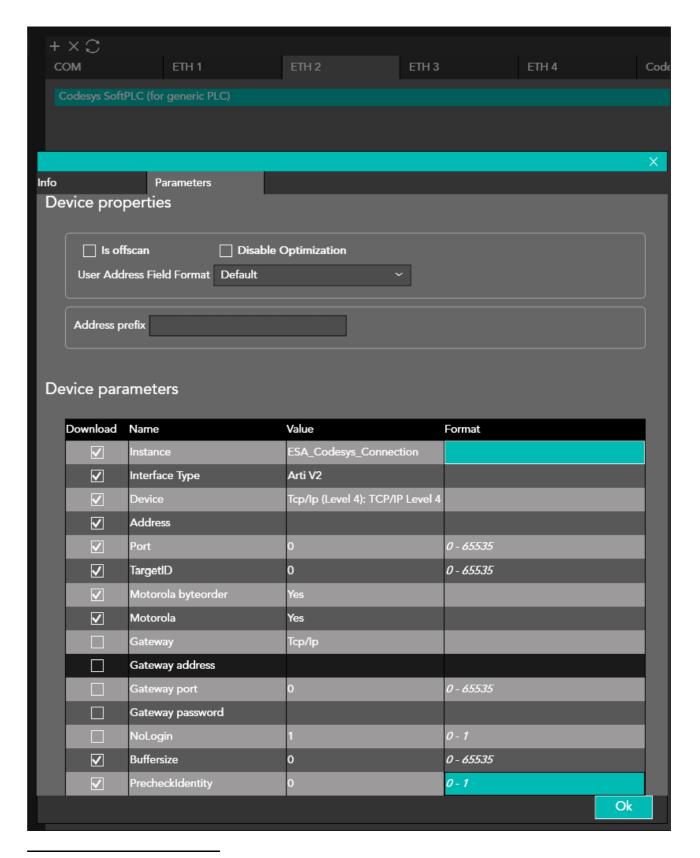
2. CODESYS SOFT PLC FOR GENERIC PLC (il driver per comunicare con un dispositivo con run time CODESYS sia esso un PLC fisico oppure un PC con a bordo CODESYS RTE).





L'unica differenza nei 2 drivers è nei parametri di comunicazione in quanto il primo richiede di indicare il .PROJECT-File, .XML-SymbolicFile, PROFILE TYPE del progetto CODESYS mentre il secondo ha delle impostazioni di parametri generici adattabili ai vari tipi di plc/RTE CODESYS 2.x, 3.x.





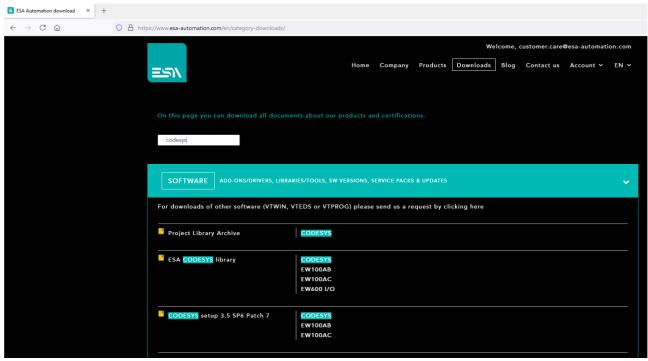


Nota1: Il firmware CODESYS cablato negli HMI-ESA ha profilo CODESYS 3.5SP6P7. Si richiede pertanto tale editor CODESYS nello sviluppare il progetto. Nessuna limitazione invece per quanto riguarda i progetti per PC in tal senso

PREMESSA (files ESA per editor CODESYS)

Prima di iniziare il progetto occorre configurare l'editor CODESYS con alcuni files ESA che permetteranno di usare librerie e configurare dispositivi tipici di ESA, appunto.

Dal sito ESA si scaricano dunque i files in figura:

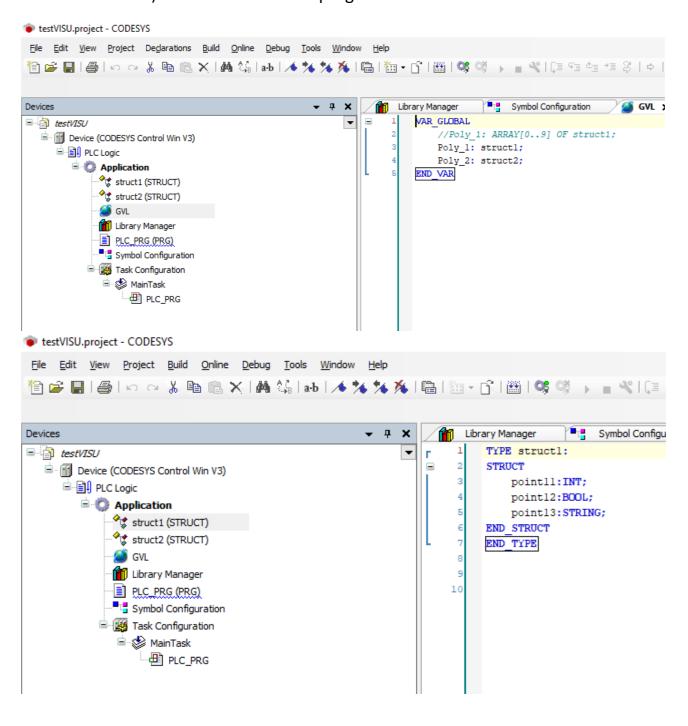


- <u>Project Library Archive</u>: progetto archivio da aprire su proprio PC. Questa operazione è sufficiente per installare su PC le librerie di sistema ESA necessarie per CODESYS.
- ESA Codesys library: files xml per la configurazione dei dispositivi ESA in CODESYS. Da installare in CODESYS>>TOOLS>>DEVICES REPOSITORY
- <u>Codesys setup 3.5sp6 Patch7</u>: setup dell'editor CODESYS richiesto per progetti HMI-ESA, come detto.

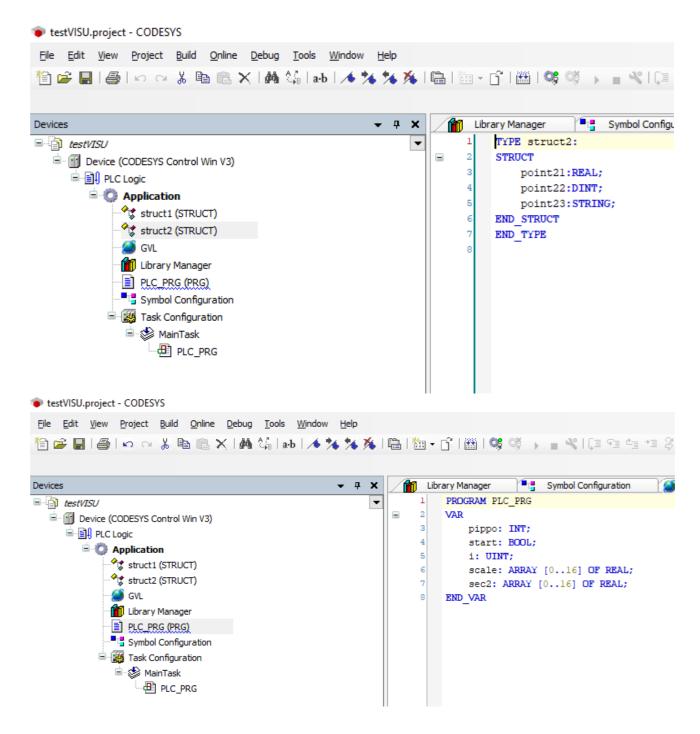


Come fare:

1) Supponiamo di avere un progetto CODESYS con diversi tipi di tag (singole e strutture) nelle varie sezioni di progetto:



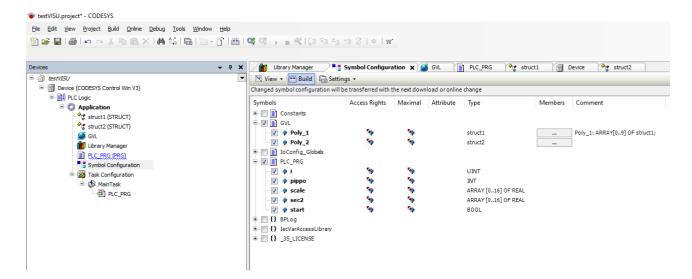




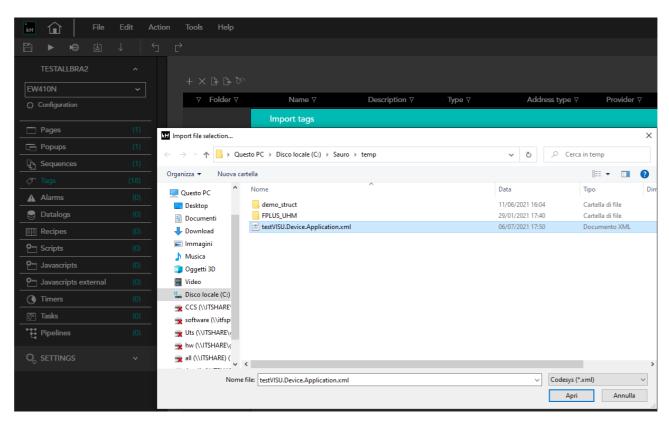
2) Seleziono nell'oggetto SYMBOL CONFIGURATION le tags che occorre importare nel progetto KREO.

CODESYS crea così un file .XML nella propria cartella di progetto (lo fa tramite comando BUILD>>GENERATE CODE oppure dopo un semplice download di progetto).



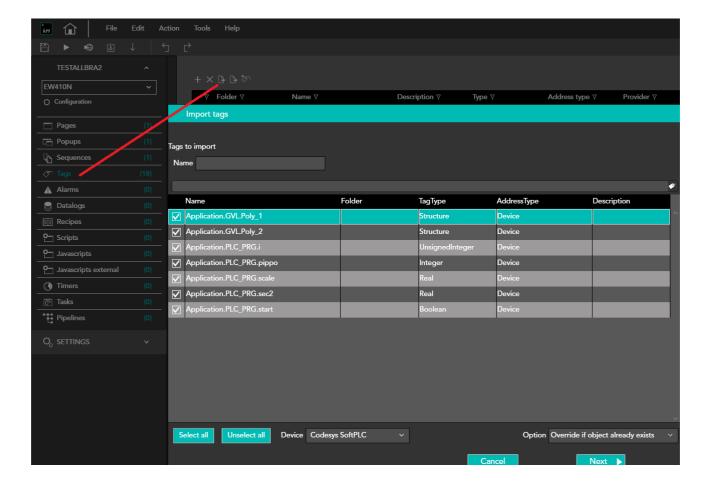


3) Dalla sezione tags di KREO importo così il file XML creato da CODESYS.



4) Le tags vengono così importate in KREO, siano esse strutture, array, singole tags e diventano disponibili per essere usate nel progetto.

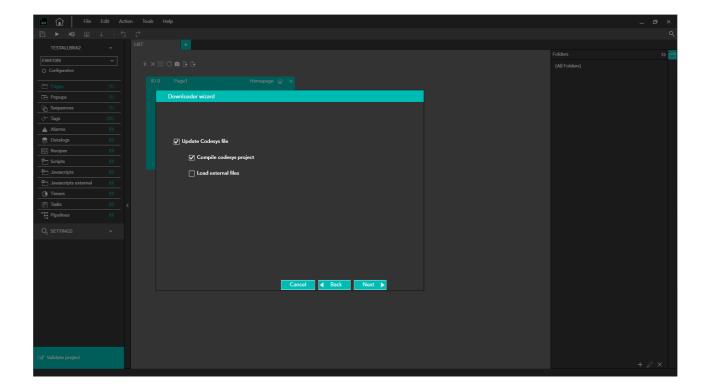




5) Nella procedura di download dei progetti KREO+CODESYS SOFT-PLC (embedded nel HMI-ESA) si puo' scegliere se trasferire i 2 progetti separatamente, ognuno con i rispettivi editor, in 2 operazioni quindi distinte oppure in un'unica operazione KREO.

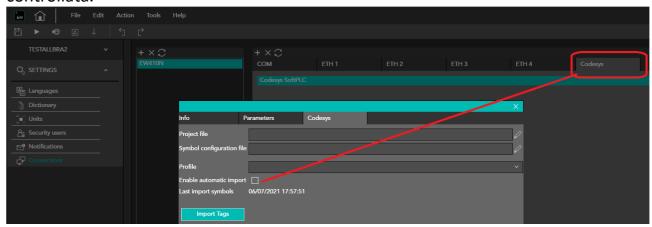
Nella procedura di download occorre in questo caso selezionare o meno il check UPDATE CODESYS FILES in figura:





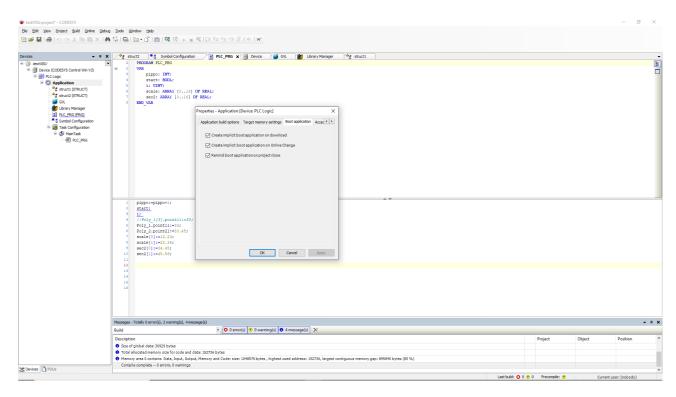
<u>Nota2</u>: Nel realizzare il progetto KREO si puo' utilizzare l'import automatico delle tag (vedi figura) ma attenzione che questo non porti a import indesiderati di tag-CODESYS nel progetto KREO.

Si consiglia dunque di disabilitarlo nel caso di un'operazione di import più controllata.



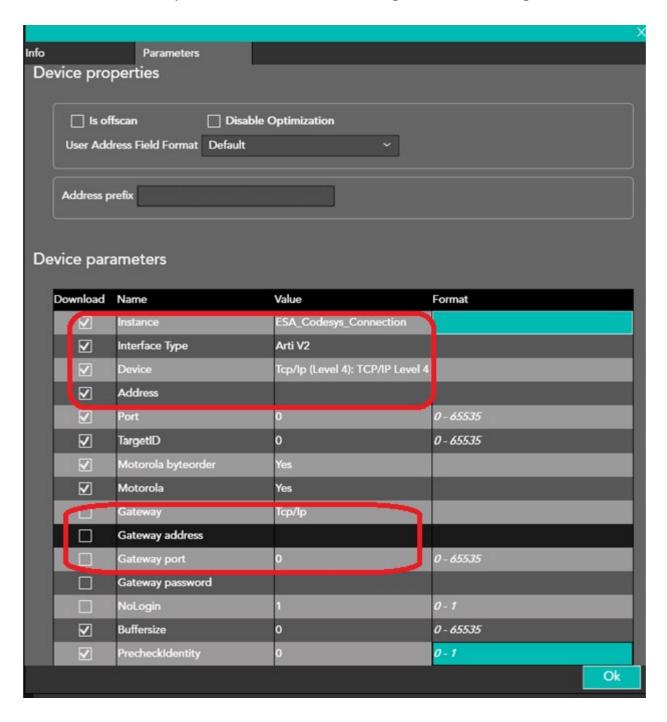


<u>Nota3</u>: Nel realizzare il progetto CODESYS si consiglia di abilitare i 3 check in figura (right click su Application >> Properties) per mantenere allineato progetto e boot-app





<u>Nota4</u>: Nei progetti KREO con driver CODESYS SOFT PLC FOR GENERIC PLC abbiamo visto che i parametri di comunicazione generici sono i seguenti:



Si vuole far notare che, in caso di progetti con sw basati su CODESYS 3.x, i parametri di comunicazione coinvolti sono quelli evidenziati in rosso (e non tutti necessariamente).



<u>Nota5</u>: Nella stessa figura il parametro ADDRESS PREFIX permette di importare tags-CODESYS eliminando il prefisso iniziale del nome.

Quindi per esempio, una tag con nome *Application.GVL.Poly_2* e con ADDRESS PREFIX= *Application.* sarà importata in KREO con nome: *GVL.Poly_2*.

<u>Nota6</u>: Nella stessa figura il parametro DISABLE OPTIMIZATION diventa utile in fase di debug.

Se abbiamo una situazione di 1 tag non definita nel plc si potrebbe avere a RT una serie di ??? su altre tag di pagina cmq configurate correttamente. Questo perché il protocollo di comunicazione ottimizza le richieste impacchettando nel frame inviato al plc il max. nr. di tags possibili. Questo però puo' inficiare la comunicazione su tutte le tag del pacchetto in caso di errore e cio' non aiuta l'utente a identificare quale tag causa il problema.

Il parametro DISABLE OPTIMIZATION forza invece una singola richiesta per ogni tag della pagina rallentando sì la comunicazione con il plc ma evidenziando esattamente i ??? sulle sole tag che presentano errore di comunicazione.



Connect ideas. shape solutions.

ESA S.p.A. | www.esa-automation.com |