

TwinCAT 3 Premier Projet

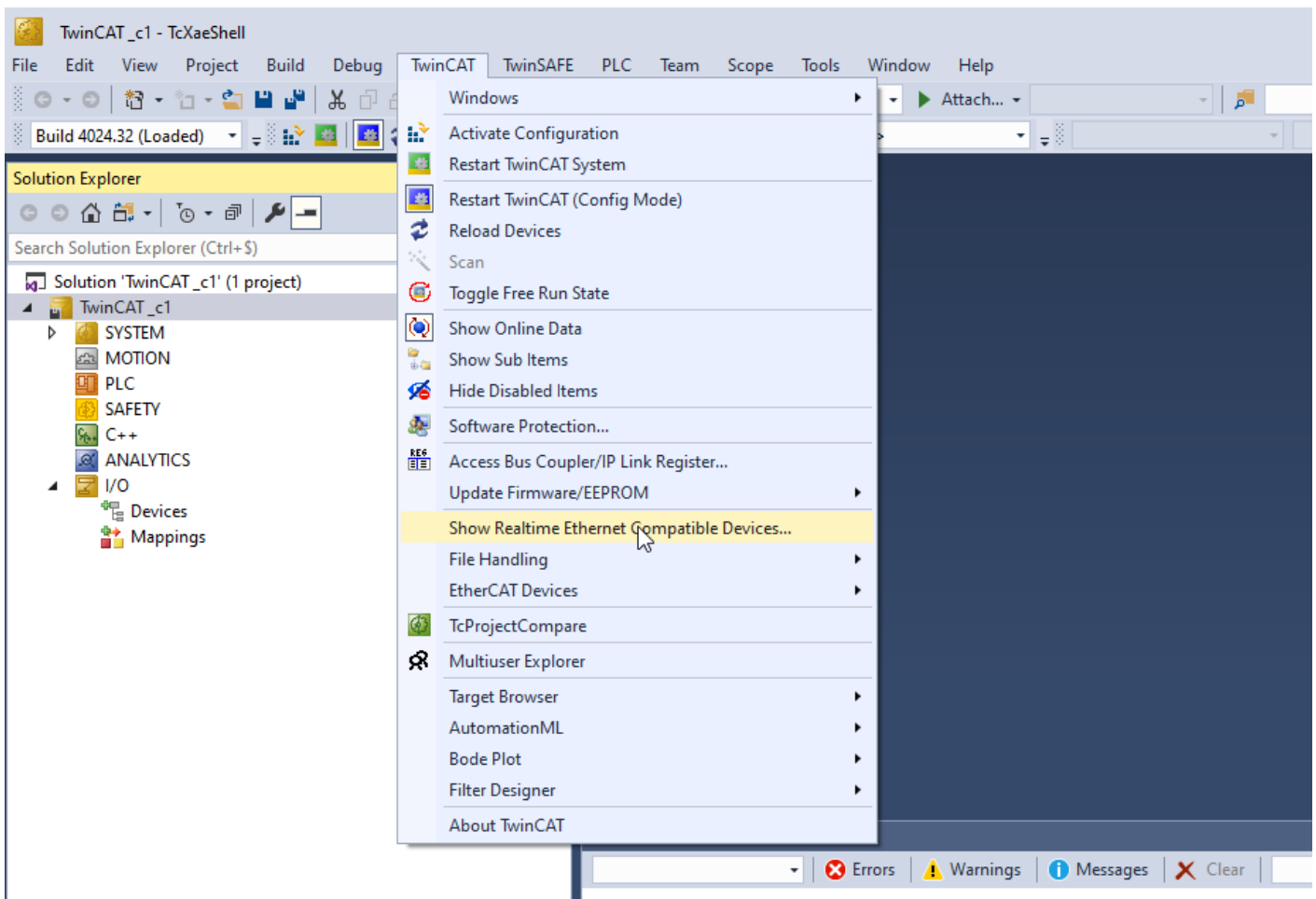
Utilisation de TwinCAT XAE Shell

Information importante : TwinCAT peut fonctionner depuis une VirtualBox, mais dans ce cas, il est nécessaire de passer sur des cores isolés. De plus on oublie les performances temps réel! Les méthodes détaillées ci-dessous correspondent à une installation classique.

Nous allons dans un premier temps nous focaliser sur l'utilisation de TwinCAT XAE Shell que vous pouvez lancer depuis le menu démarrer.

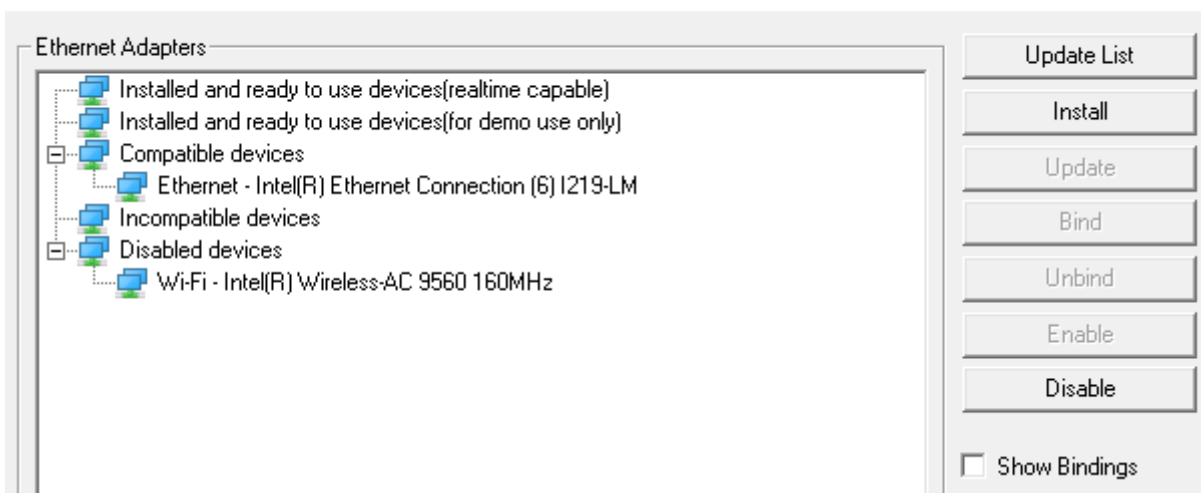
Configuration de l'interface Ethernet

Nous allons tout de suite configurer l'interface réseau afin qu'elle puisse détecter les Runtime d'IPC ou les I/O déportés sur bus EtherCAT. Dans l'onglet TwinCAT, allez dans Show Ethernet Compatible Devices...



On installe la carte réseau filaire compatible

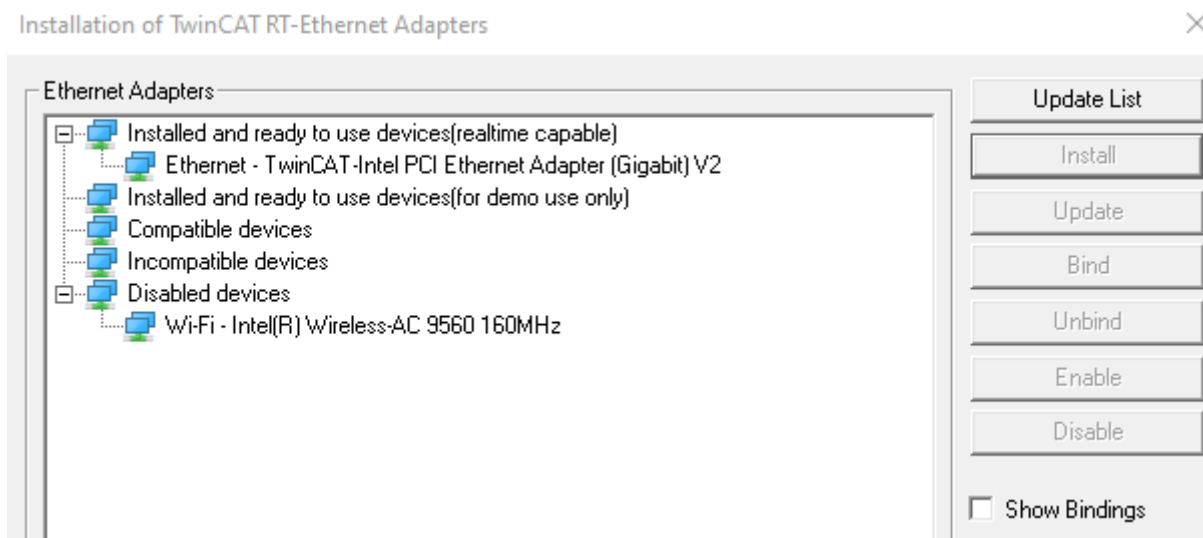
Installation of TwinCAT RT-Ethernet Adapters



Deux possibilités:

- vous avez une carte réseau Ethernet Intel compatible avec Beckhoff et vous pourrez profiter des performances temps réel. La carte apparaîtra dans les "realtime capable"
- votre carte est d'un autre fabricant, vous ne profiterez pas du real-time, mais c'est suffisant pour les essais que nous ferons. Votre carte apparaîtra dans les for demos use only.

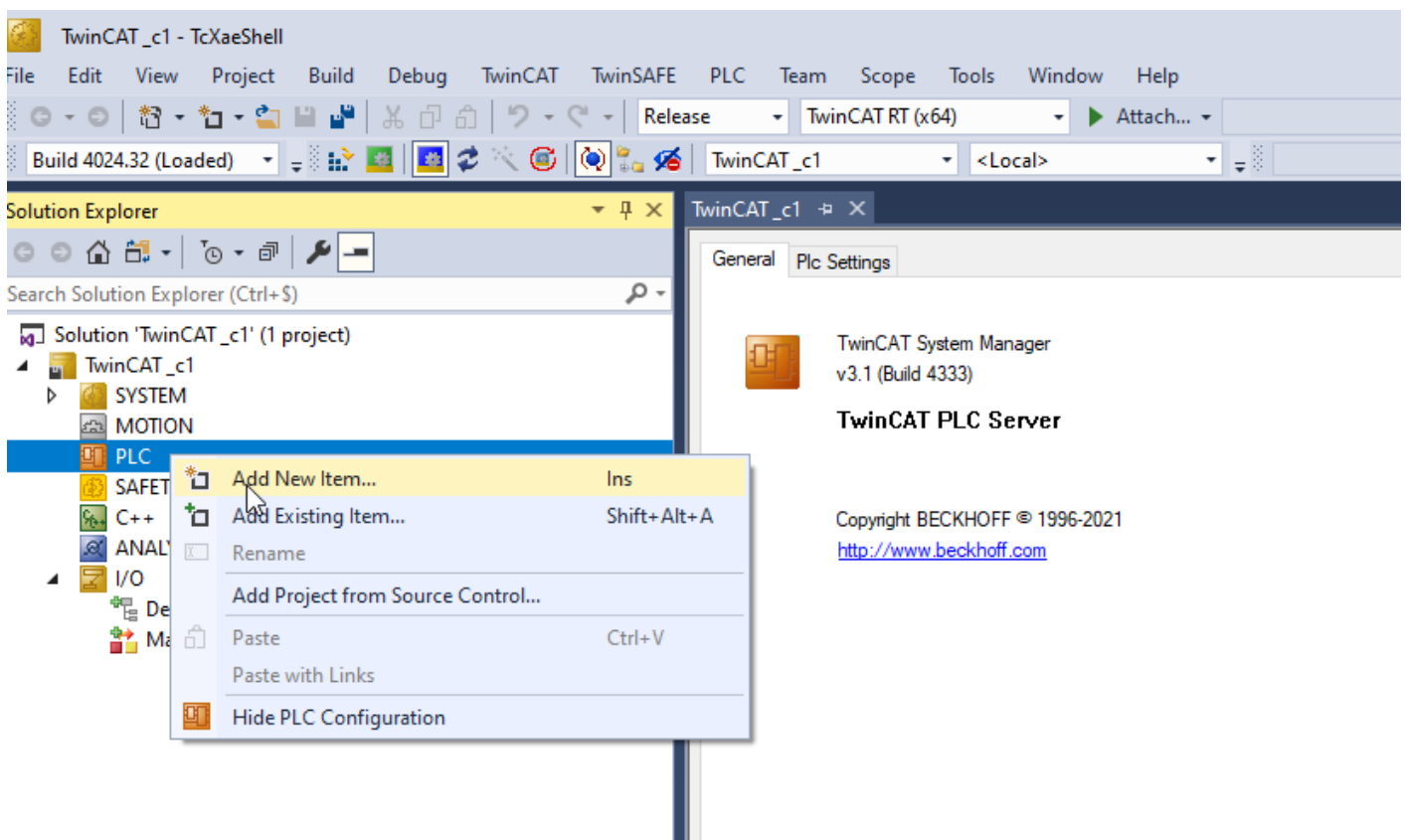
Sur mon Dell Pro, il y a une interface Intel compatible Real-Time. Les essais avec un PC fonctionnant avec une carte Realtek n'ont pas posé de soucis également. Tant que l'on ne cherche pas les performances Realtime il n'y a pas d'inquiétude à avoir sur le type de carte Ethernet utilisé.



Premier programme PLC

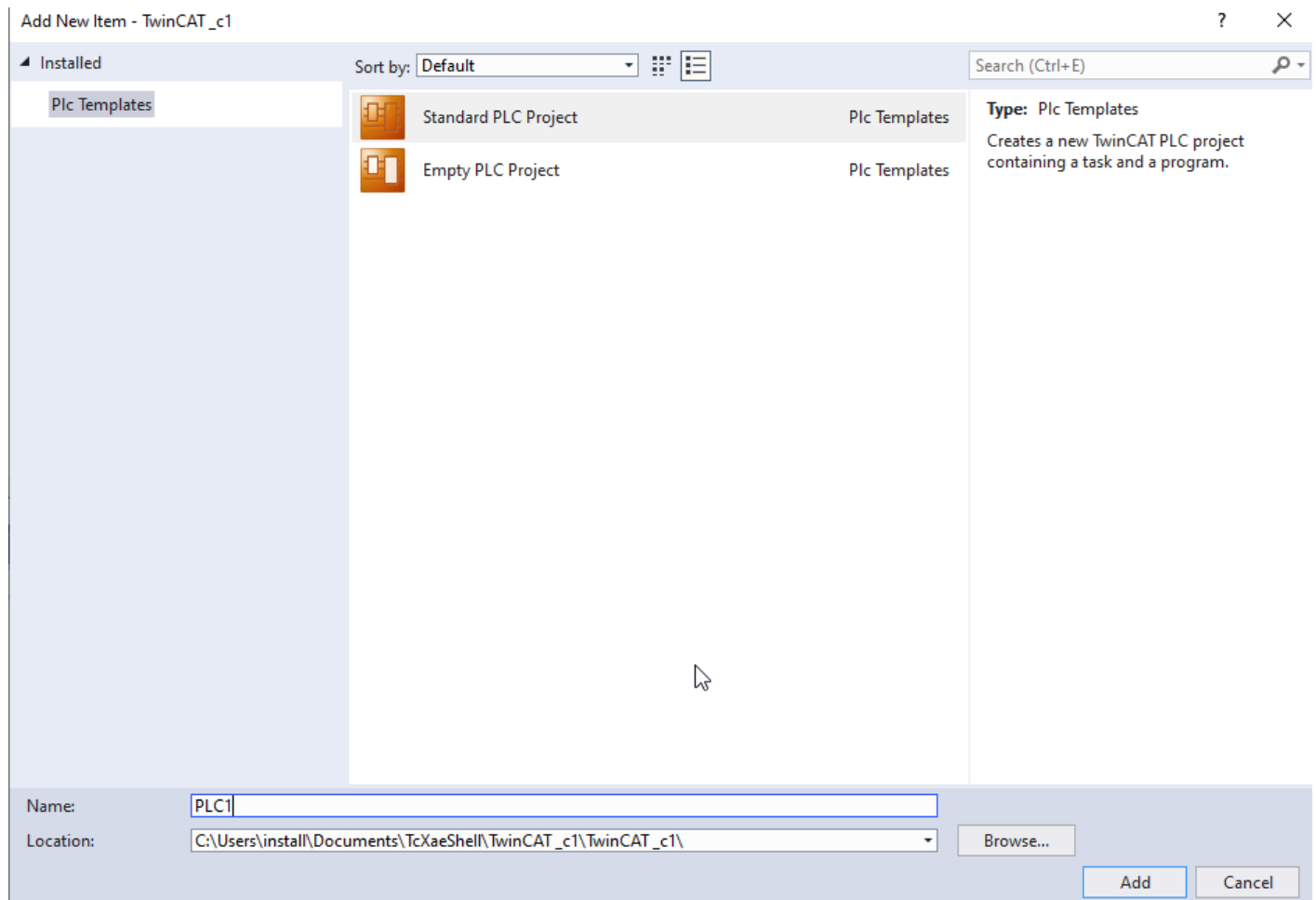
Dans cette section, nous allons mettre en oeuvre un automate. Ceux qui ont suivi les articles sur Codesys vont remarquer que l'interface de programmation est très proche. C'est normal car TwinCAT est basé sur CodeSys pour la partie programmation automate en langage IEC-61131-3.

On crée un automate en faisant un clic droit sur PLC puis Add New Item



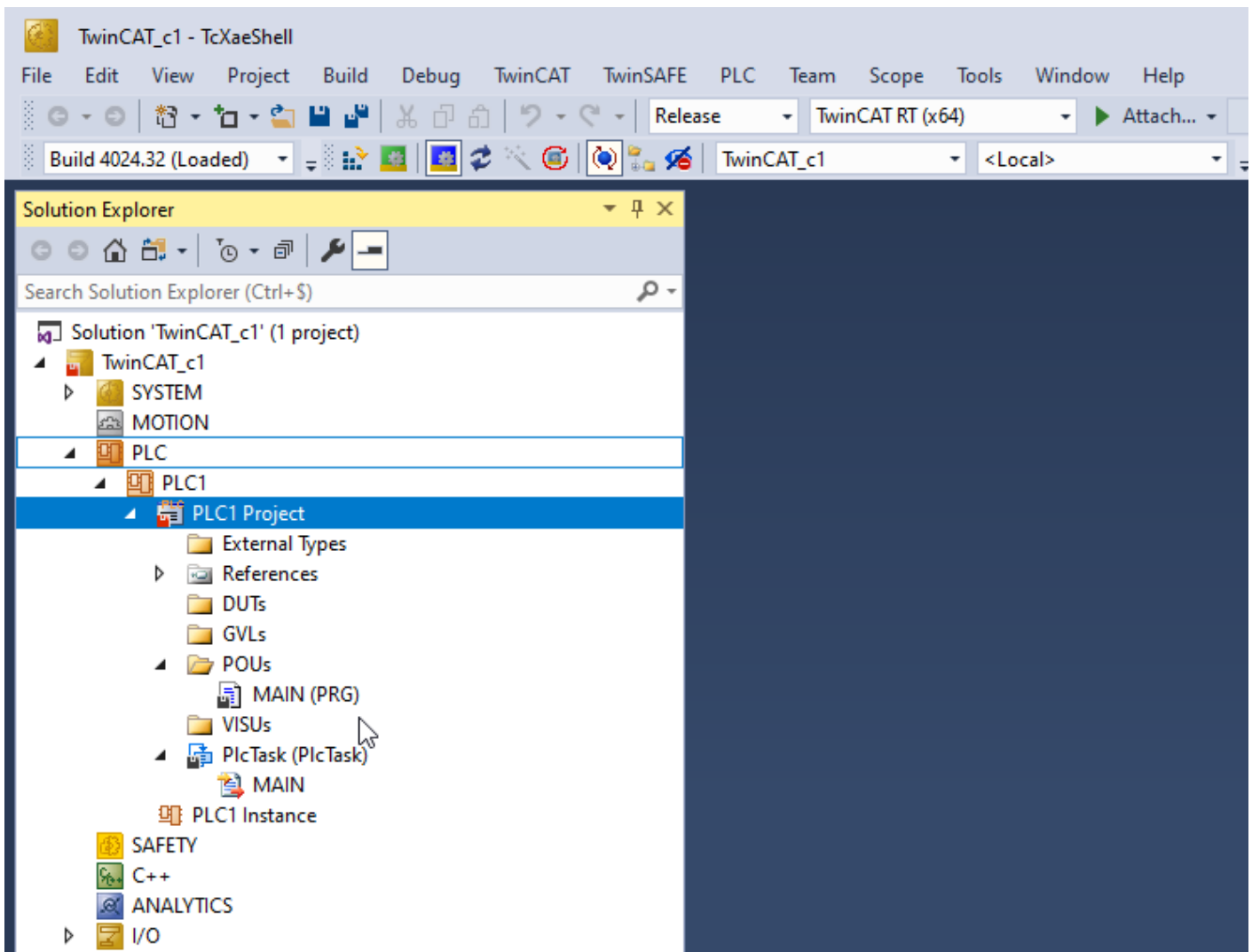
On choisit :

- Standard PLC Model,
- que l'on nomme PLC1



On déploie PLC1, puis POU pour Program Organization Unit.

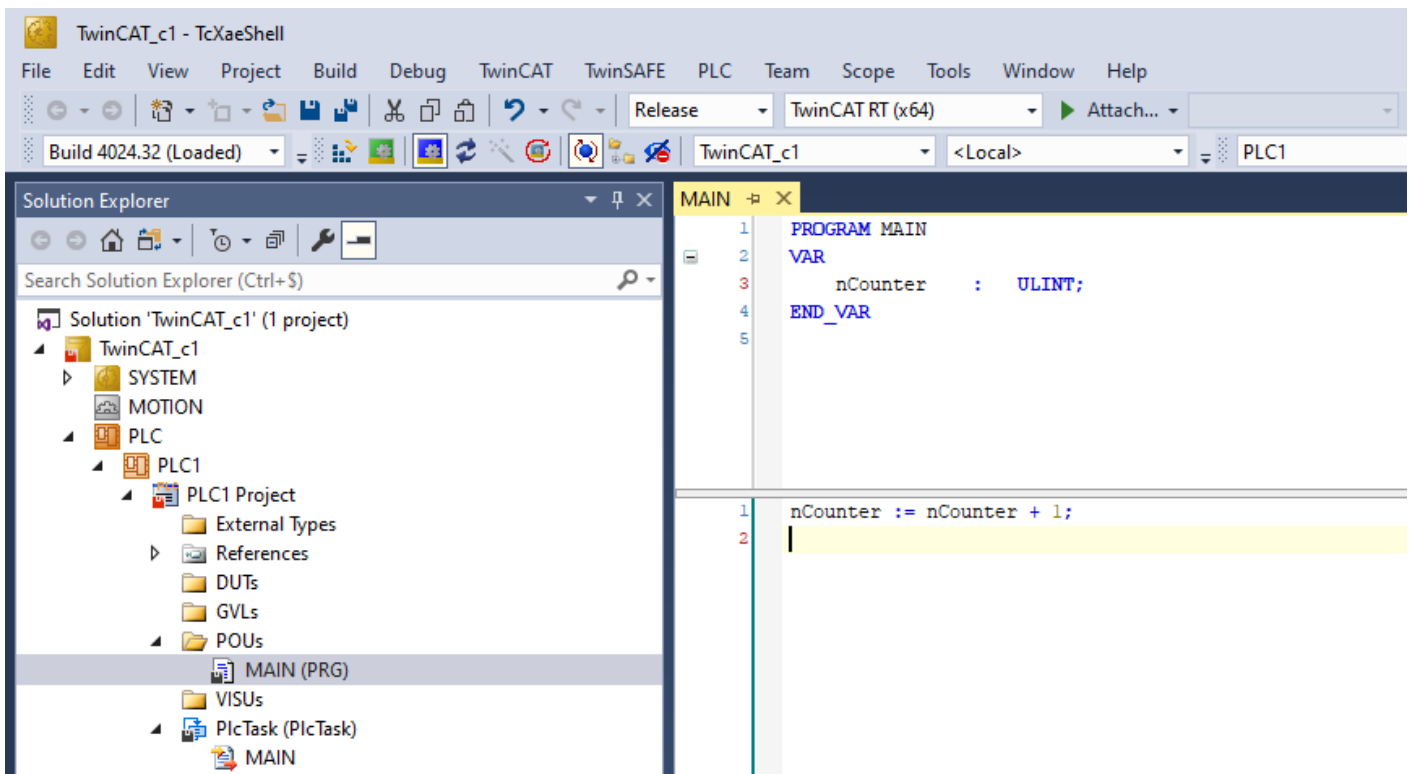
Un fichier MAIN(PRG) s'y trouve.



Programmation du Main

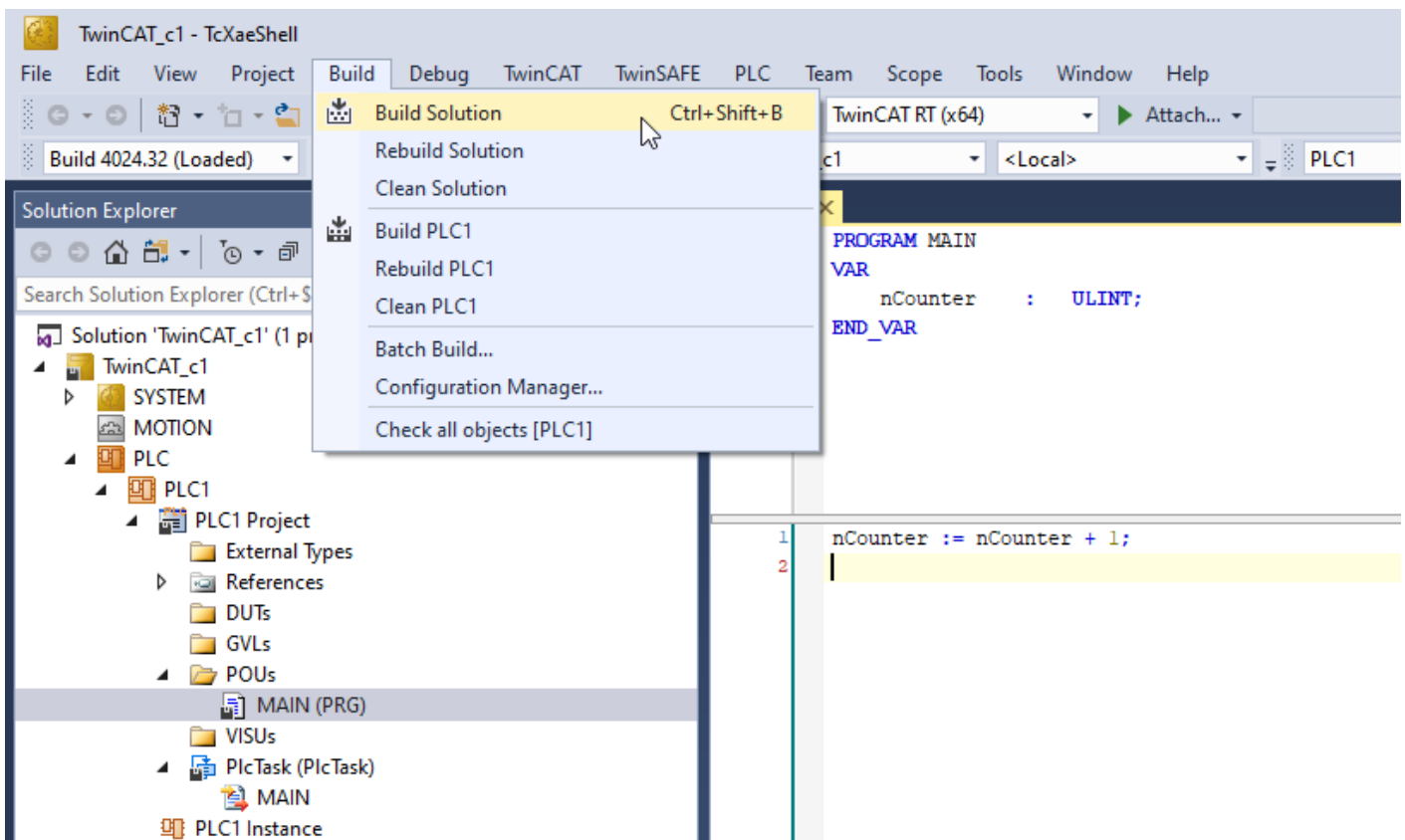
Le premier programme que nous allons réaliser pour valider le bon fonctionnement de l'installation est un compteur qui va s'incrémenter toutes les 10 ms, c.à.d., tous les temps de cycle associés à la tâche PLC1.

Une variable nCounter de type Unsigned Long Int est déclarée, et dans le main, nous ajouterons +1 à cette variable. Si tout se déroule correctement, au démarrage, la variable sera initialisée à 0 et toutes les 10 ms, elle va s'incrémenter.

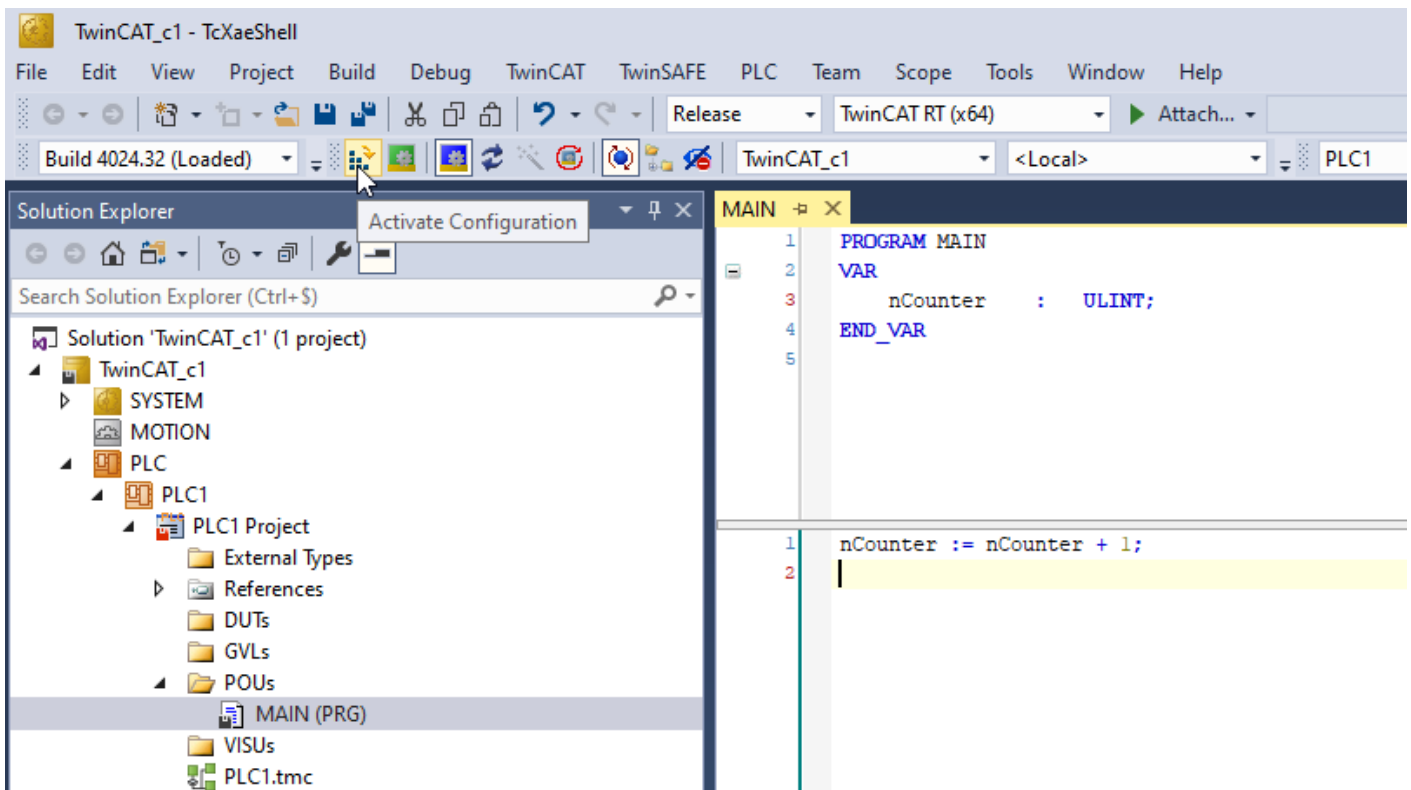


On n'oublie pas de sauvegarder le projet.

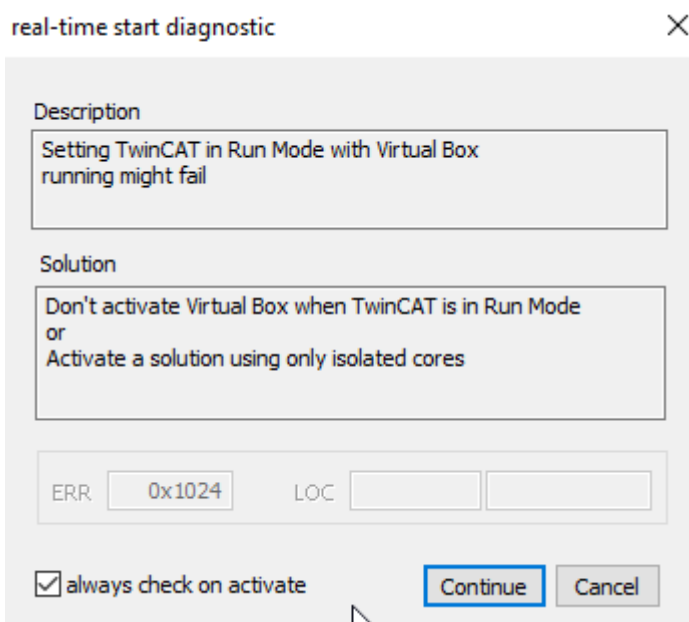
On lance le Build du projet



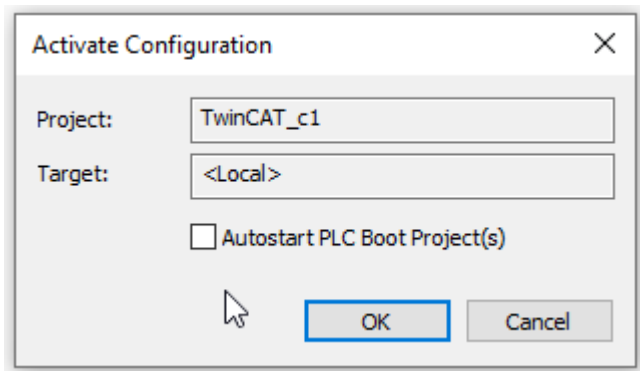
On active le projet avec Activate Configuration pour que ce soit celui-ci qui sera déployé dans le Runtime d'exécution.



Il se peut également que vous ayez un message de ce type : faire Continue



On fait deux fois OK pour activer la configuration et pour accepter le passage en Mode Run (on était en mode Config actuellement)



TcXaeShell



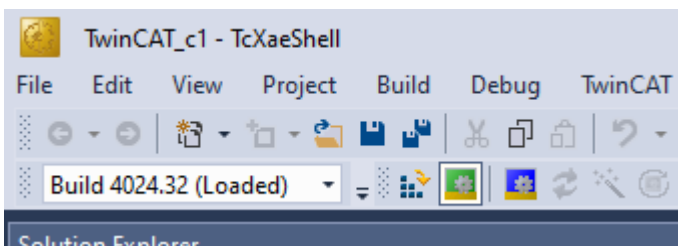
Restart TwinCAT System in Run Mode



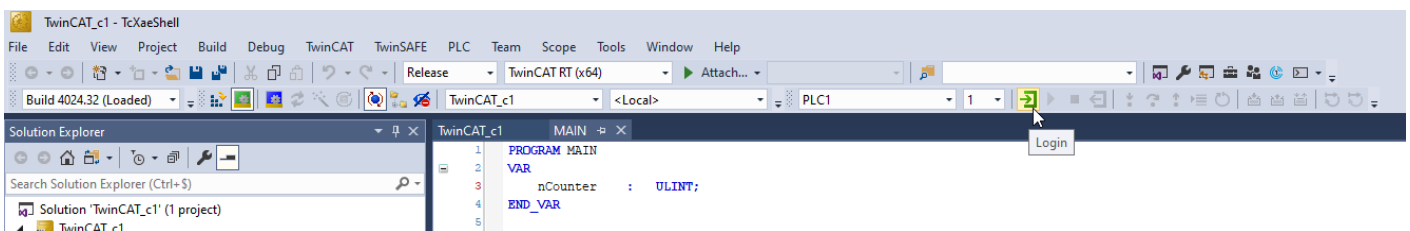
OK

Cancel

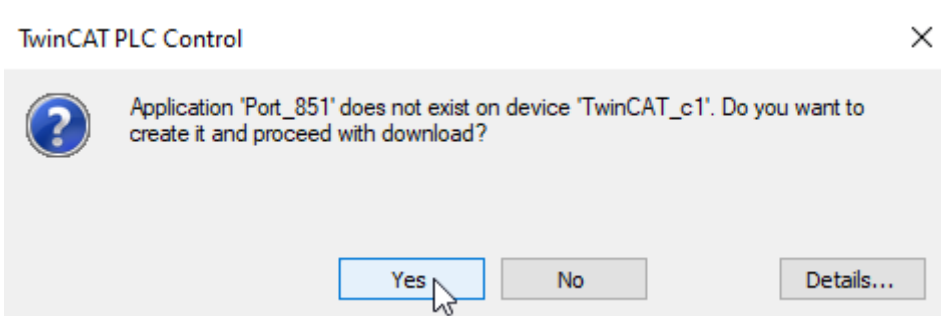
La case verte avec la roue doit être activée, pour que l'on soit dans le mode Run. (La case bleue représente le mode Config)



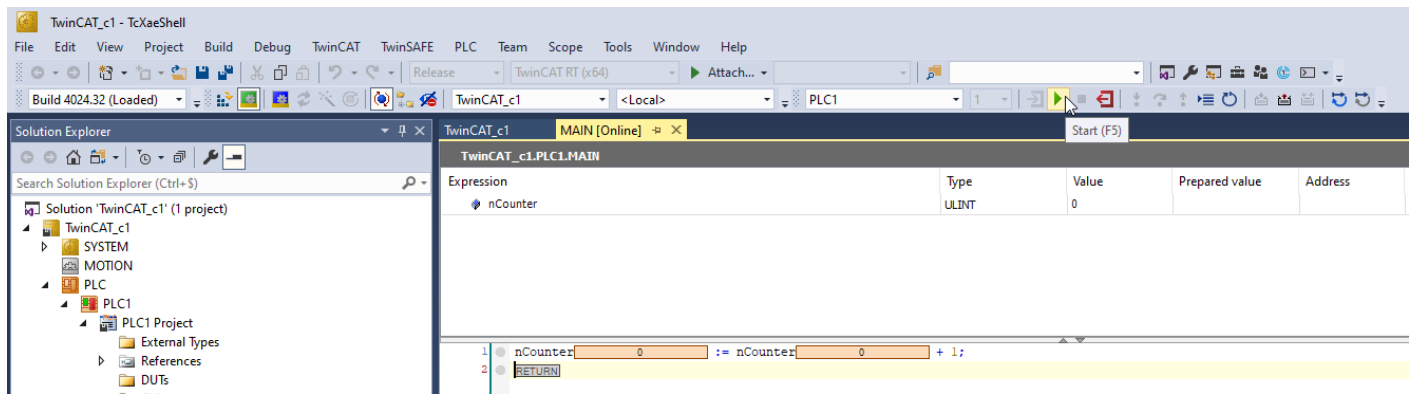
On clic sur la case verte Login



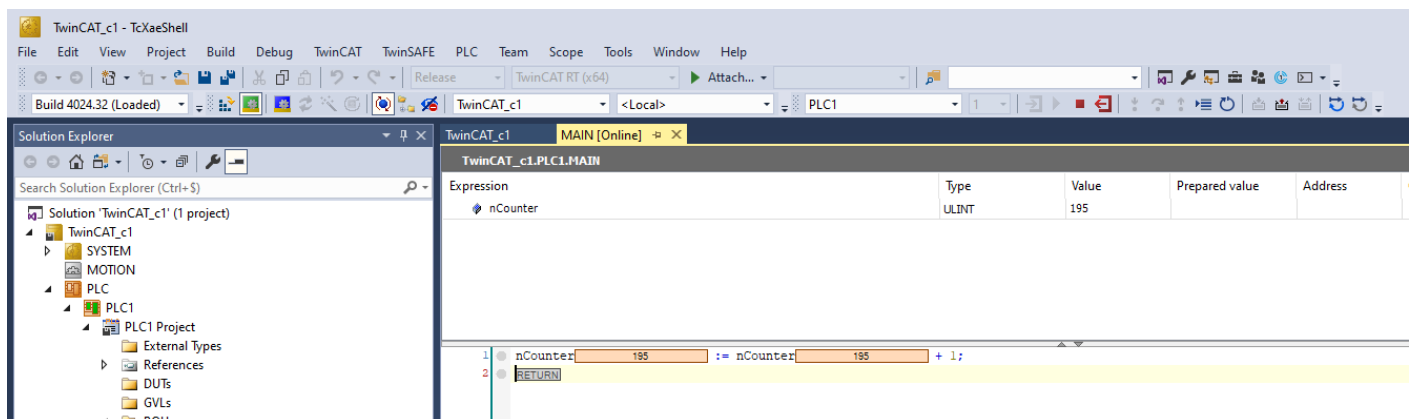
On autorise la création de l'application et le téléchargement de l'exécutable dans le Runtime en cliquant sur Yes.



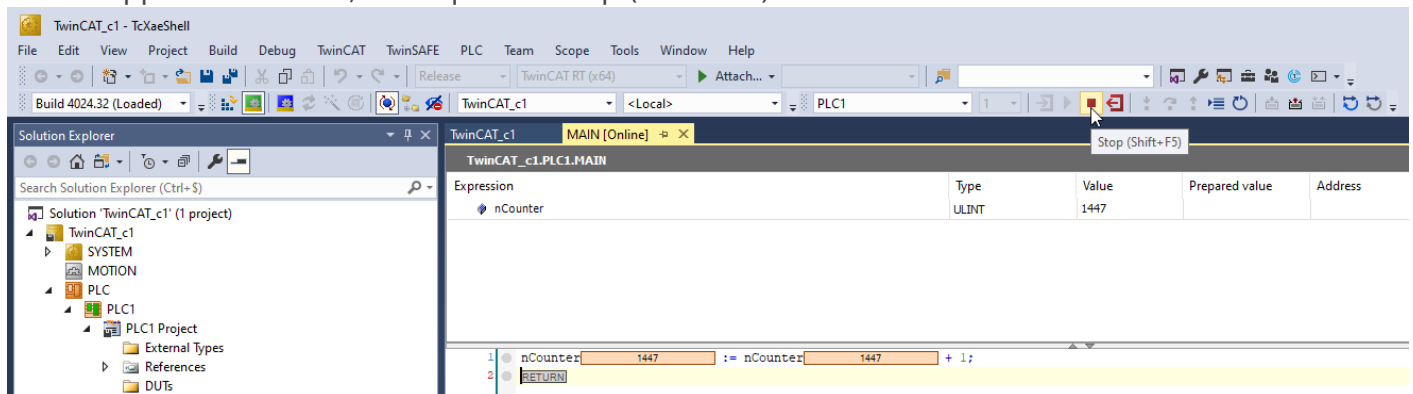
On clique sur le bouton Start (F5) pour démarrer l'application



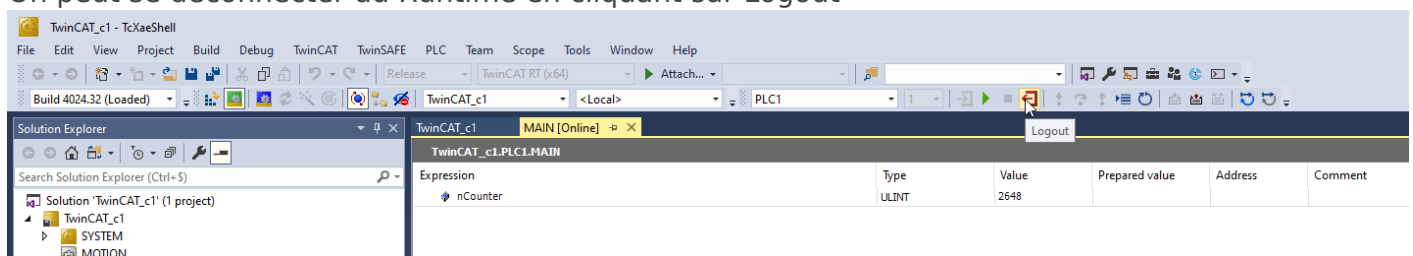
On observe l'incrémentation du compteur, (on augmente de 100 toutes les secondes)



Pour stopper le Runtime, on clique sur Stop (Shift+F5)

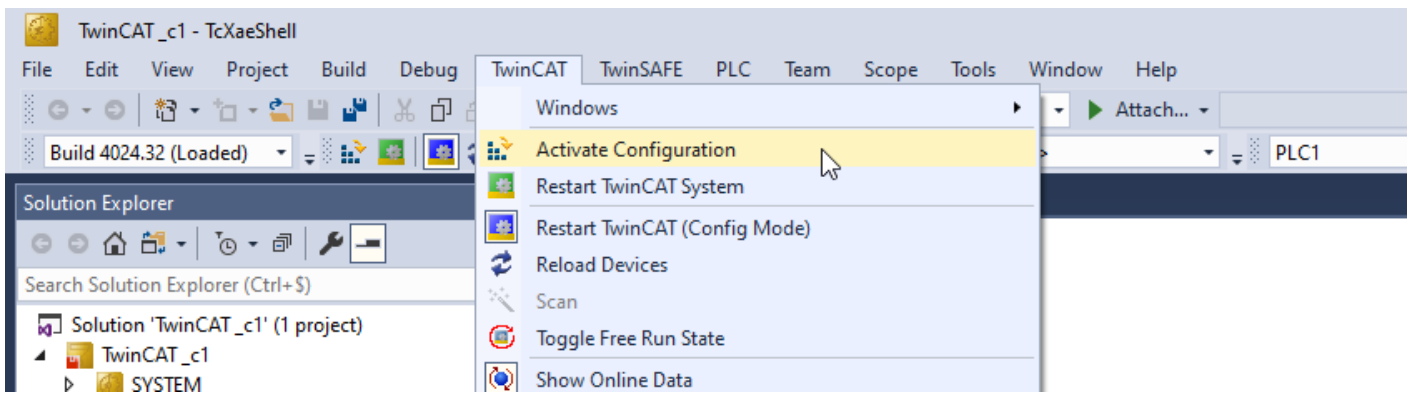


On peut se déconnecter du Runtime en cliquant sur Logout



Pour revenir dans le mode de configuration, cliquer sur :

- Restart TwinCAT Config Mode



Bilan intermédiaire :

L'objectif de ce premier programme était de vérifier la bonne installation de TwinCAT3. Si vous n'arrivez pas à exécuter le code, vérifiez que :

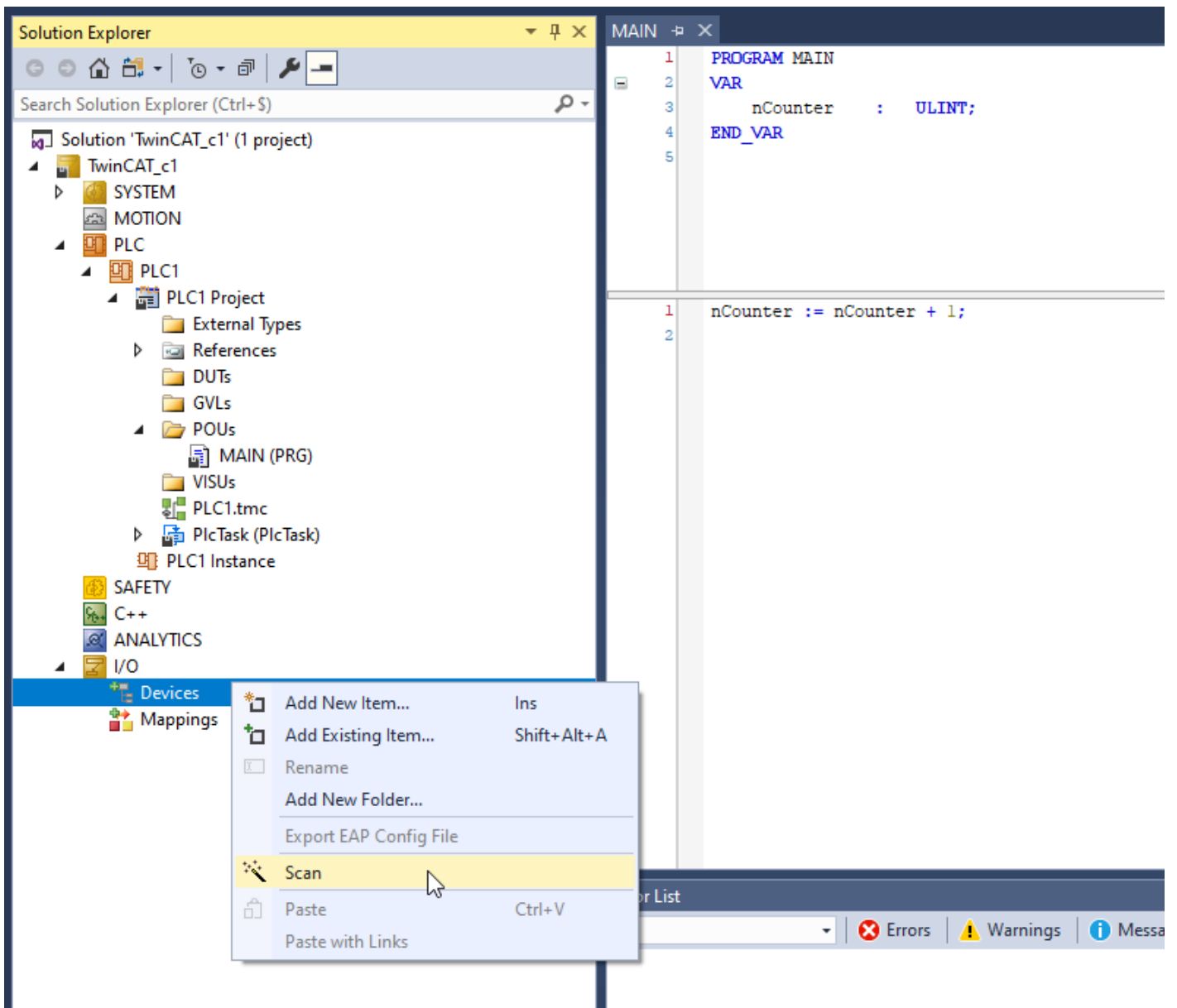
- le driver pour la carte Ethernet est installé et que celle ci apparaisse soit en compatible Real-Time, soit en use for Demos Only
- le build de votre projet soit effectué après avoir enregistré le programme
- le projet est activé, ne pas le faire génère des erreurs difficiles à interpréter
- que vous ayez passé Twincat en Mode Run (la petite roue dans la case verte)

Ajout d'Entrées/Sorties déportées sur bus EtherCAT

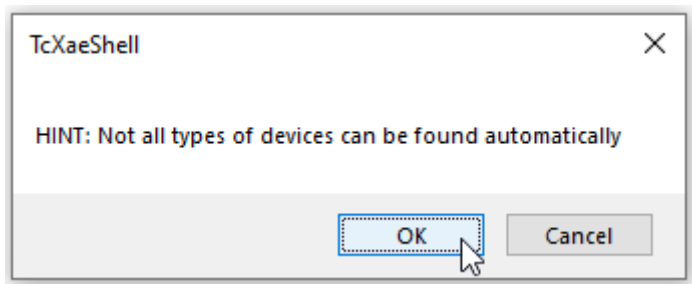
Présentation du module BK1120

Scan du bus EtherCAT

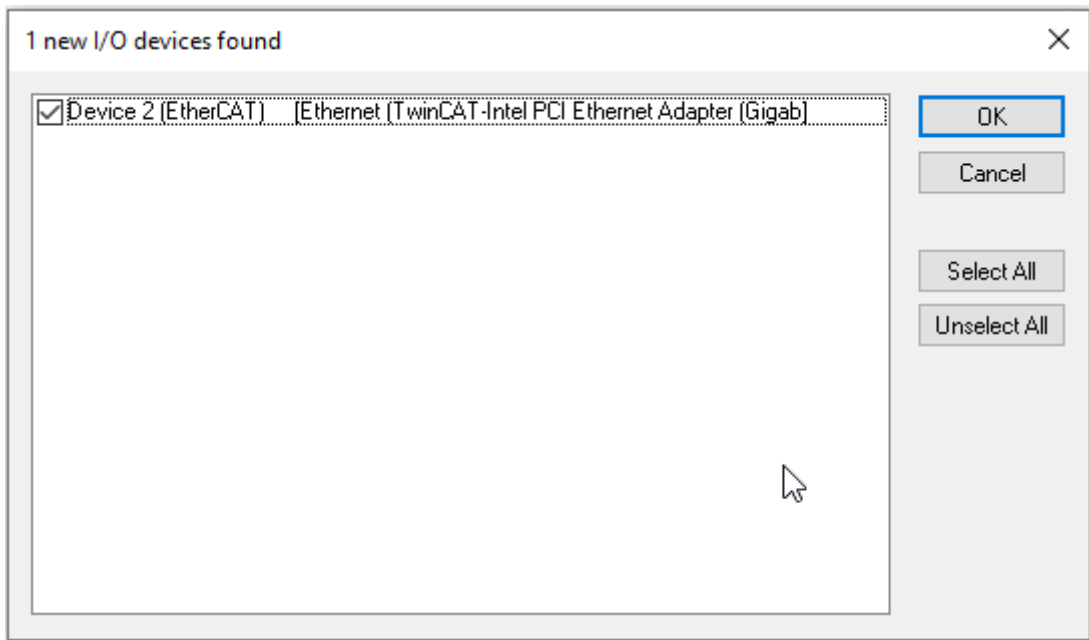
Pour scanner les Entrées/Sorties déportées qui sont connectées sur le bus EtherCAT, il faut déployer l'onglet IO et faire un Scan. N'oubliez pas que sur un bus EtherCAT, il est nécessaire de brancher directement la carte Ethernet du PC sur le module d'Entrées/Sorties déportées sans passer par un Switch! On branche en direct!



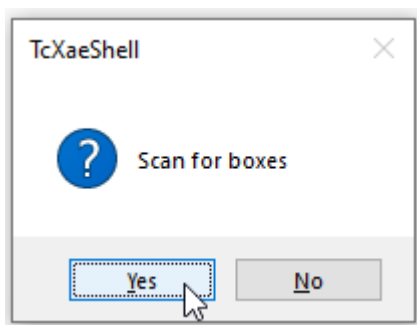
Faire OK sur le HINT. Les I/O de chez Beckhoff sont reconnues automatiquement, les I/O de fabricants tiers peuvent nécessiter une configurationn manuelle.



Un I/O a été trouvé.

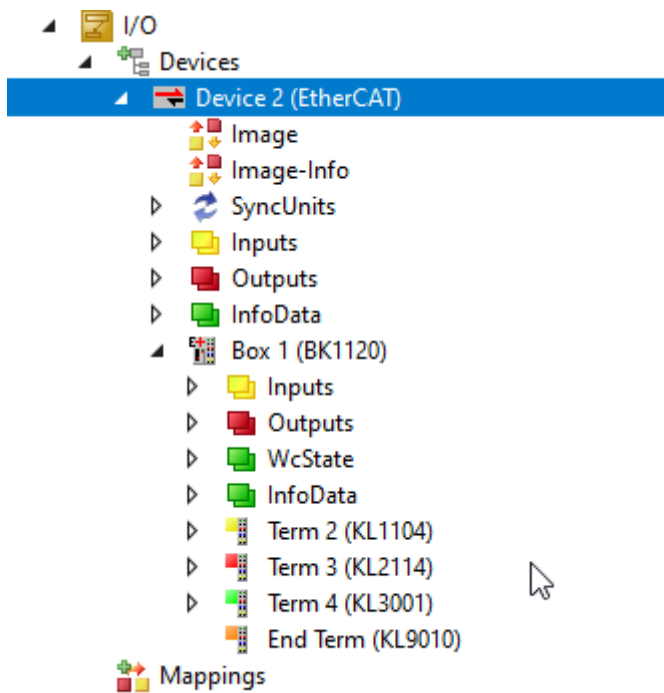


On autorise le Scan des Box et le Free Run (Le Free Run permet de piloter/forcer directement les Entrées/Sorties)



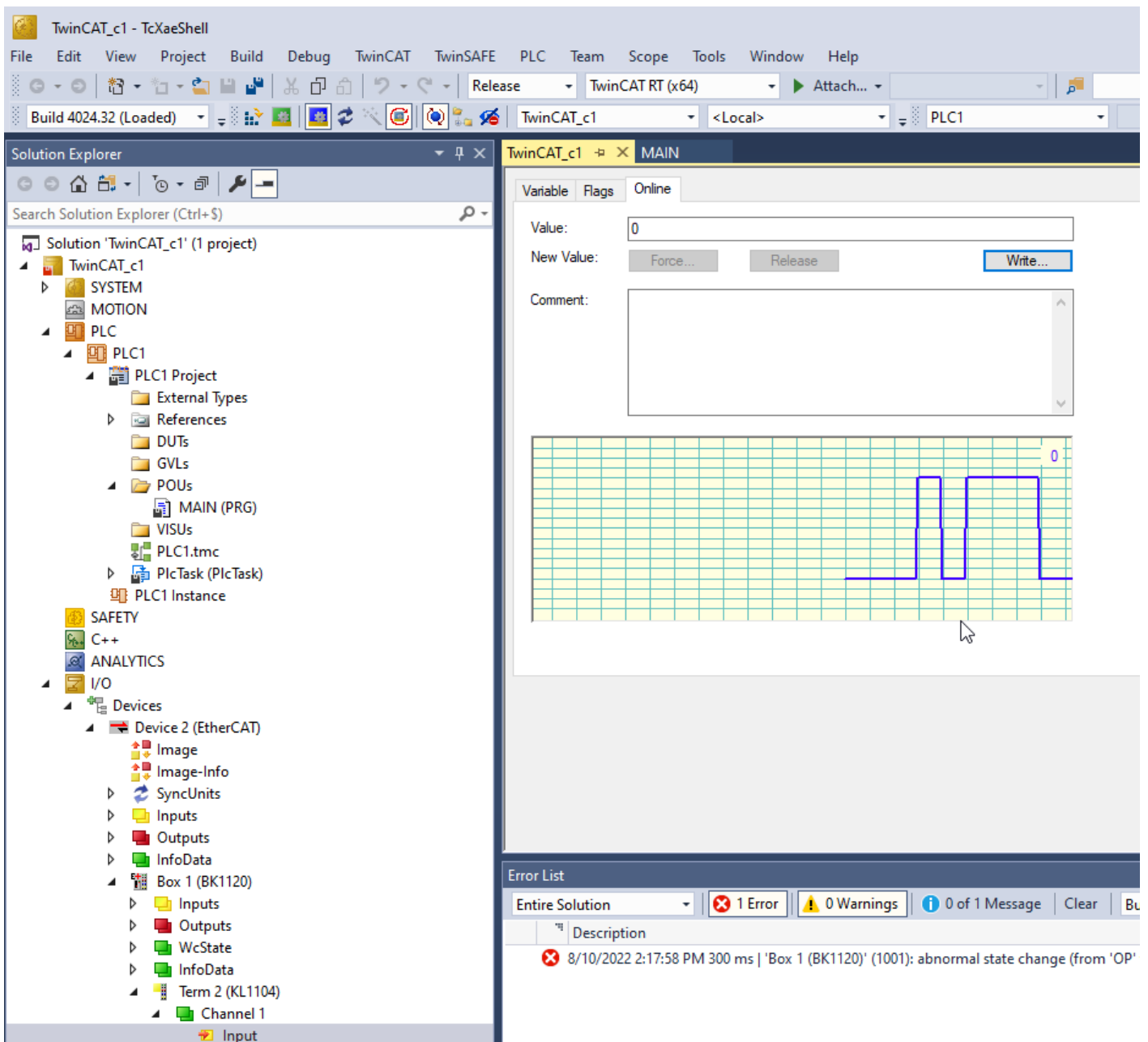
Le module d'I/O déporté BK1120 est bien reconnu, ainsi que les cartes d'entrées-sorties associées :

- KL1104 -> 4 channel Digital Input
- KL2114 -> 4 channel Digital Output
- KL3001 -> 1 channel Analog Input
- KL9010 -> carte de terminaison du bus interne

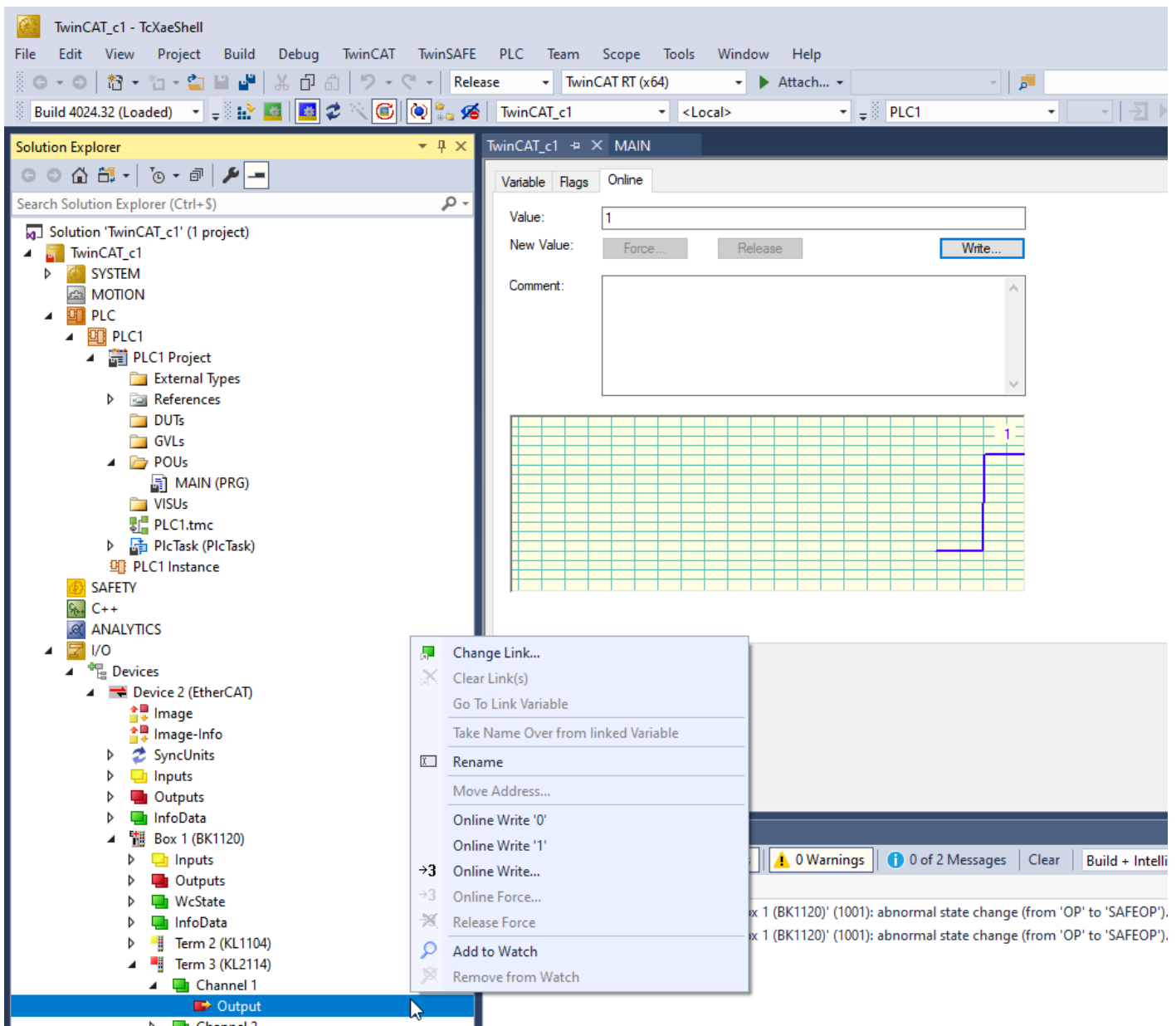


Lecture et forçage des I/O

Avant de démarrer l'association (mappage) des entrées/sorties, vérifions que celles-ci puissent être lues dans le cas des entrées digitales et forcées pour les sorties digitales. L'activation du de l'interrupteur sur l'entrée digitale doit être vue dans le signal déroulant si vous avez activé le free RUN comme précisé précédemment.

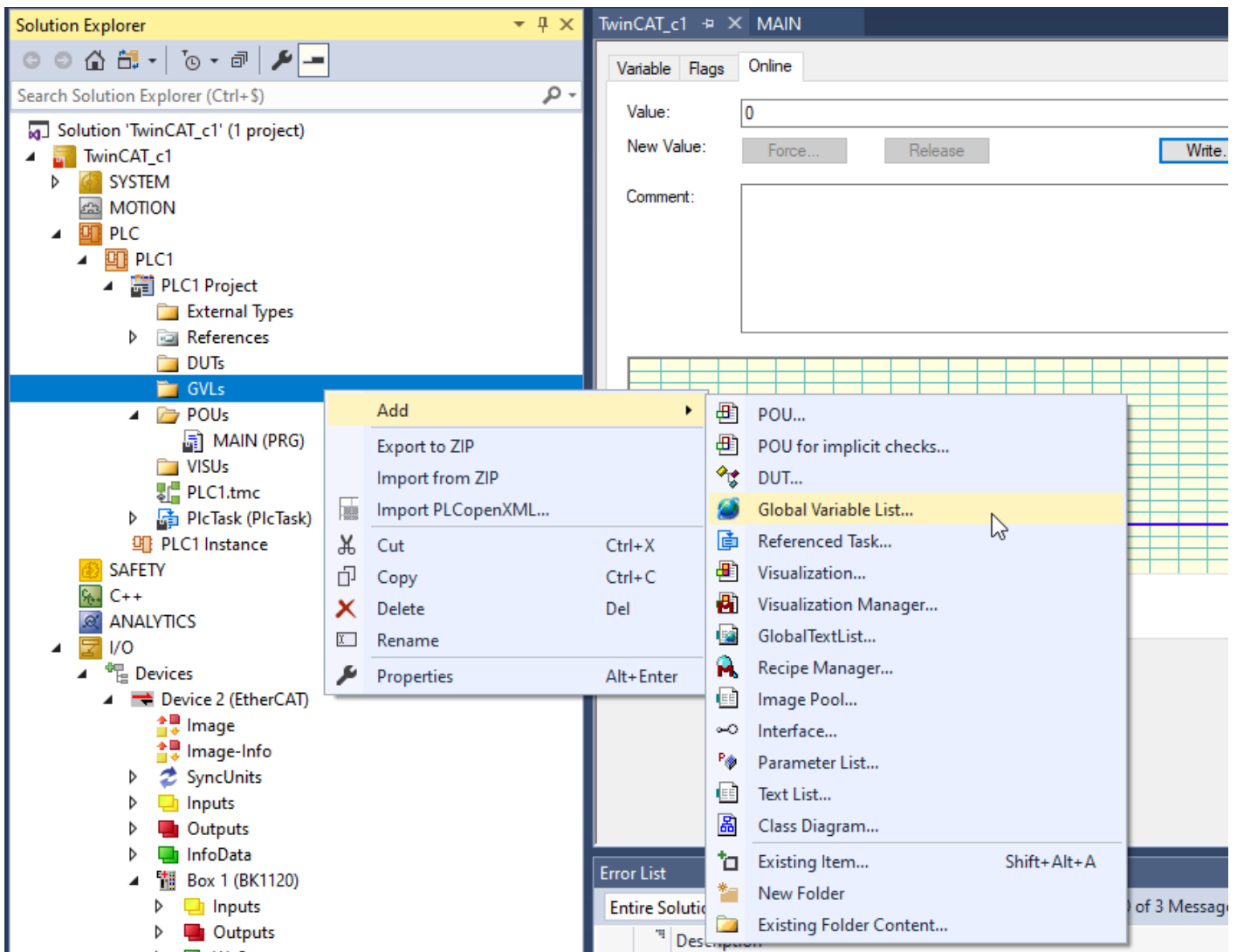


Les sorties digitales peuvent être forcées en faisant des Online Write '0' ou '1'




Liens entre I/O déportées et variables PLC

Nous pouvons maintenant réaliser le lien entre les entrées/sorties déportées et les variable de l'automate. Dans GVLs (Global Variable Lists) faire Add et Global Variable List.



Nommer cette liste IO par exemple et faire Open.

Add Global Variable List ✕

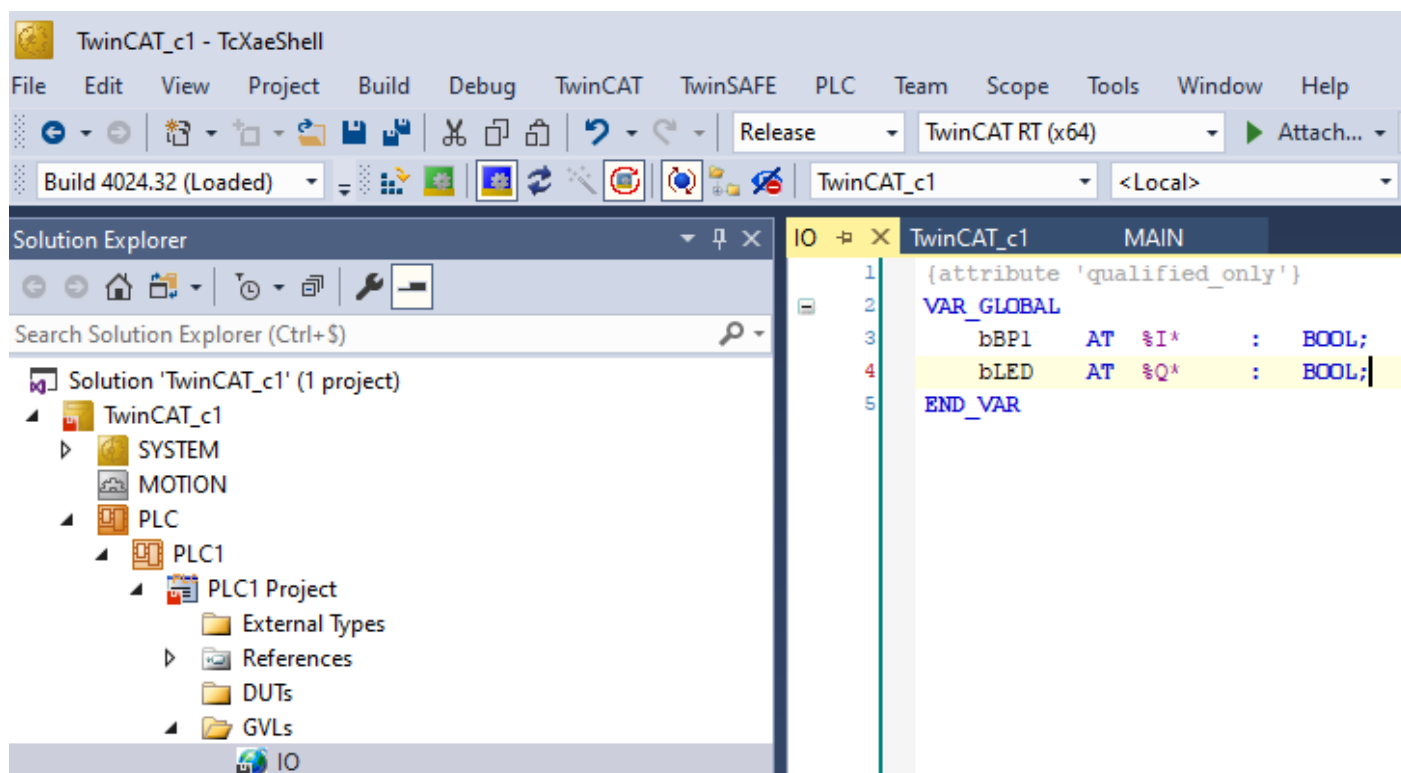
 Create a new global variable list

Name:

IO

Open Cancel

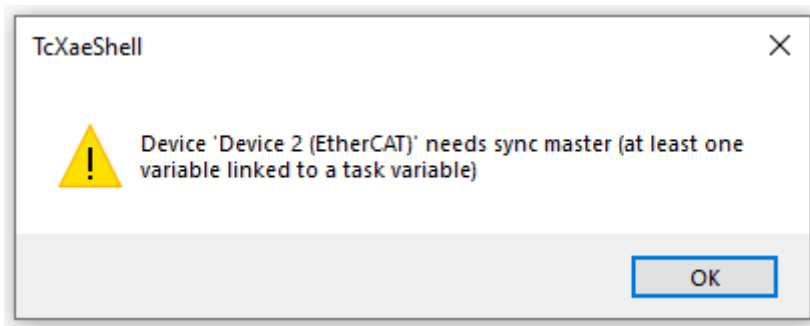
Dans la liste, ajouter les deux variables (Bouton et Led) de la manière suivante :



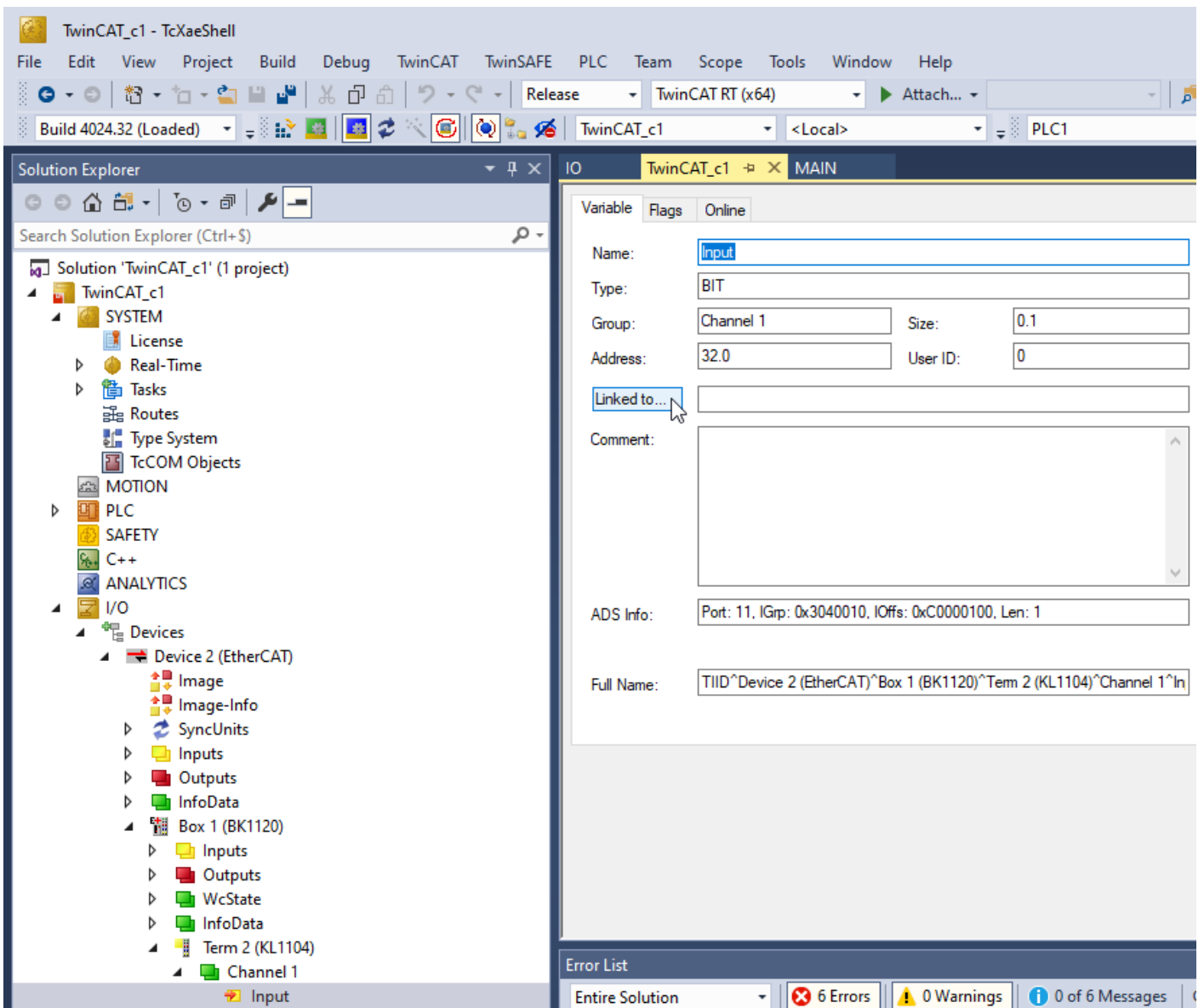
The screenshot shows the TwinCAT_c1 - TcXaeShell interface. The Solution Explorer on the left shows the project structure: Solution 'TwinCAT_c1' (1 project) > TwinCAT_c1 > SYSTEM > MOTION > PLC > PLC1 > PLC1 Project > GVLs > IO. The main editor window shows the variable declaration for the IO variable list:

```
1 {attribute 'qualified_only'}
2 VAR_GLOBAL
3     bBP1    AT  %I*    :    BOOL;
4     bLED    AT  %Q*    :    BOOL;
5 END_VAR
```

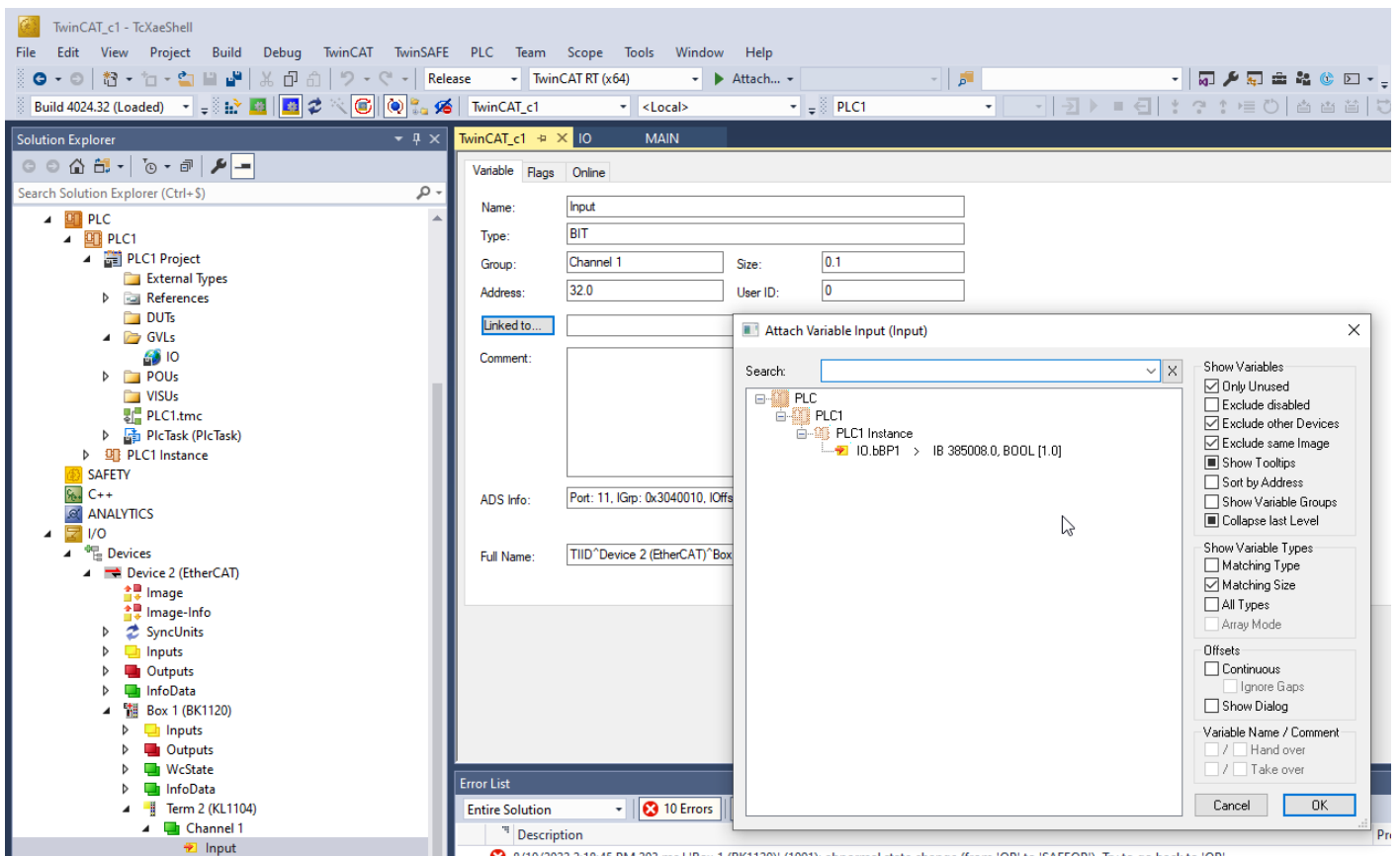
N'oubliez pas de sauvegarder et de faire un Build. Une avertissement peut apparaître comme ci-dessous, faire OK.



Nous pouvons maintenant faire le lien (link) entre les entrées-sorties physiques déportés et les variables globales de l'automate. Choisir l'Entrée du module déporté à affecter, faire Linked to ...



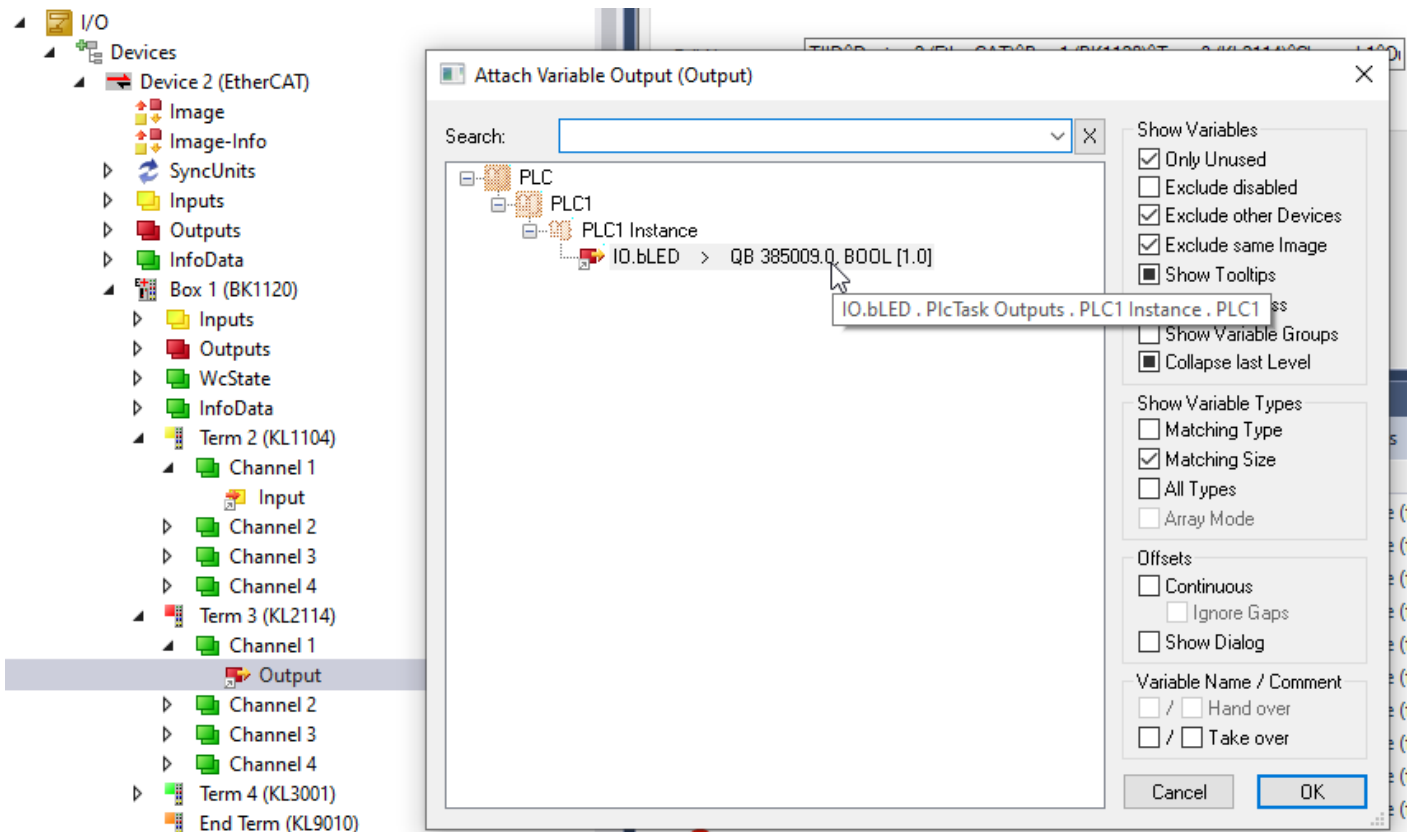
Et choisir la variable IO.bBP1



Vérifier que le Linked To ait bien pris en compte l'association de variables :

Linked to... IO.bBP1 . PlcTask Inputs . PLC1 Instance . PLC1

De la même manière, nous affectons la variable de sortie bLED à la sortie physique du module déporté :

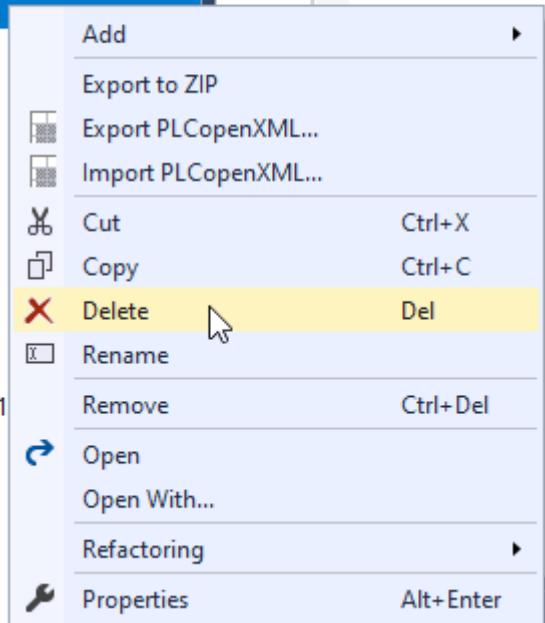


Programme en Ladder

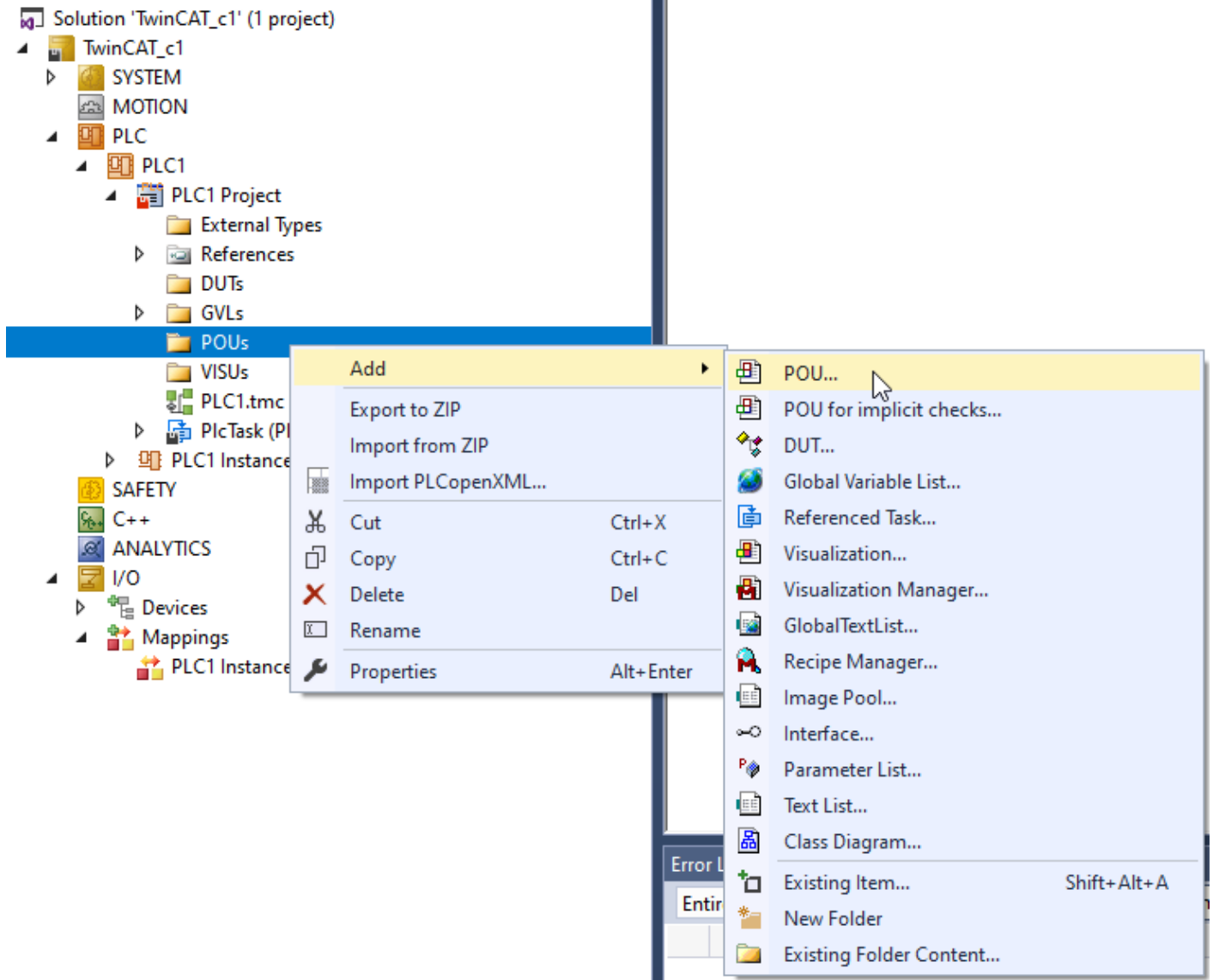
Nous allons maintenant effectuer un programme en Ladder. Effaçons le programme MAIN précédent en faisant Delete :

- PLC
 - PLC1
 - PLC1 Project
 - External Types
 - References
 - DUTs
 - GVLs
 - POUs
 - MAIN (PRG)
 - VISUs
 - PLC1.tmc
 - PlcTask (PlcTask)
 - PLC1 Instance
 - SAFETY
 - C++
 - ANALYTICS
 - I/O
 - Devices
 - Mappings
 - PLC1 Instance - Device 2 (EtherCAT) 1

```
1 nCounter := nC  
2
```



Faire Add puis POU...



Dans Add POU..., nommer le programme MAIN, vérifiez bien que

- le type est Program
- et choisir l'implémentation langage en Ladder Logic Diagram

Add POU



Create a new POU (Program Organization Unit)

Name:

Type

☒ Program

☐ Function Block

☐ Extends: ...

☐ Implements: ...

☐ Final ☐ Abstract

Access specifier: ...

Method implementation language:

☐ Function

Return type: ...

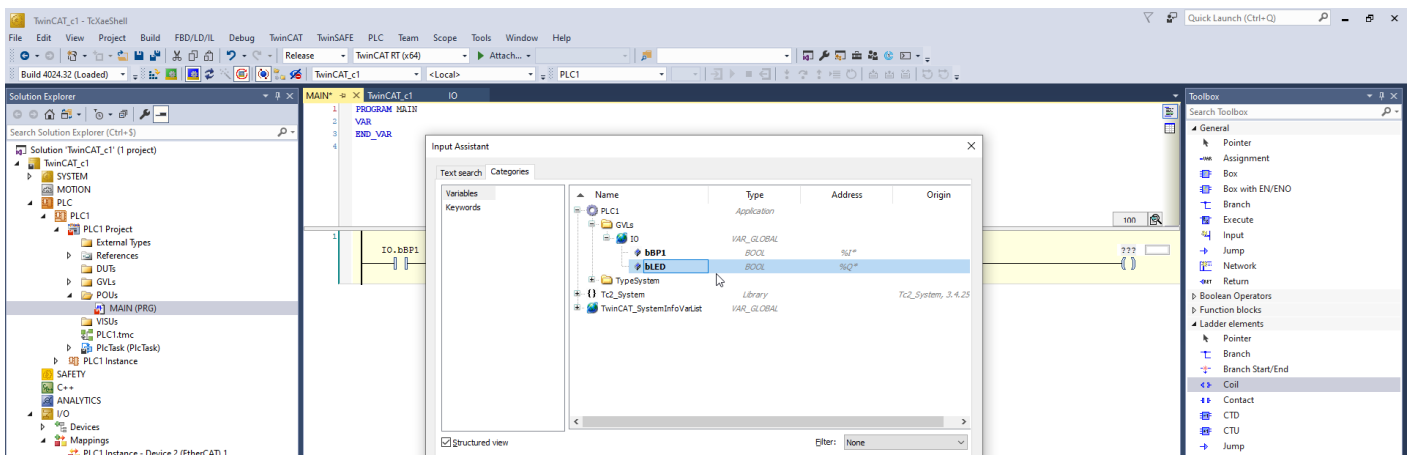
Implementation language:

Placer un Contact dans le MAIN à l'aide du menu d'outils ToolBox à droite. Cliquer sur ce contact pour l'associer à la variable bBP1 qui se trouve dans PLC1 -> GVLs -> IO -> bBP1

The screenshot shows the TwinCAT 3 IDE interface. The main window displays a Ladder Logic Diagram (LD) with a contact (bBP1) and a coil (bLED). The Solution Explorer on the left shows the project structure. The Toolbox on the right contains various ladder logic elements. The Input Assistant dialog is open, showing the variable bBP1 selected from the PLC1 -> GVLs -> IO -> bBP1 path.

Name	Type	Address	Origin
PLC1	Application		
IO	VAR_GLOBAL	%I*	
bBP1	BOOL	%I*	
bLED	BOOL	%Q*	
TypeSystem	Library		Tc2_Standard, 3.3.3
Tc2_Standard	Library		Tc2_System, 3.4.25
Tc2_System	Library		Tc2_Module, 3.3.21
Tc2_Module	Library		
TwinCAT_SystemInfoVarList	VAR_GLOBAL		

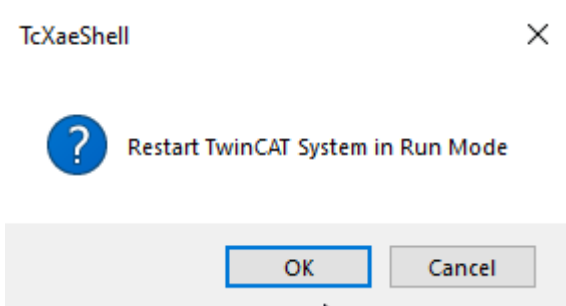
Placer un Coil à la suite du Contact et l'associer à la variable bLED.



Test

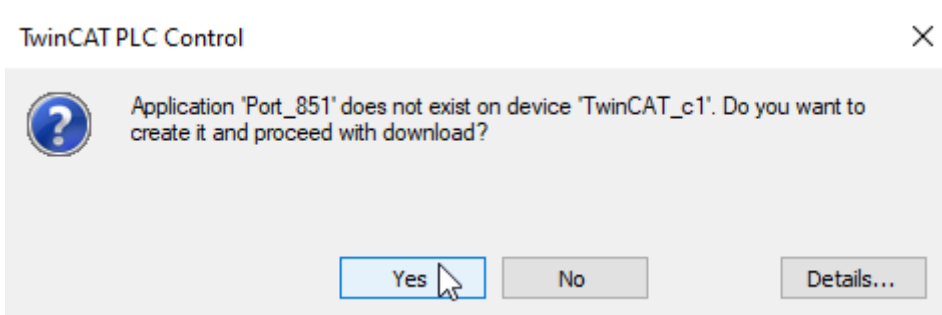
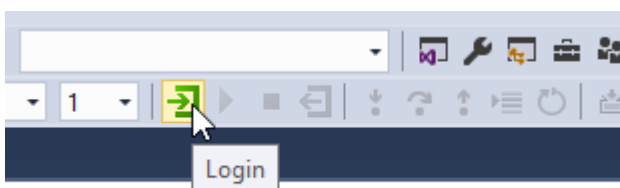
On fait un save, un Build et on active le projet.

On fait OK pour passer en mode RUN :



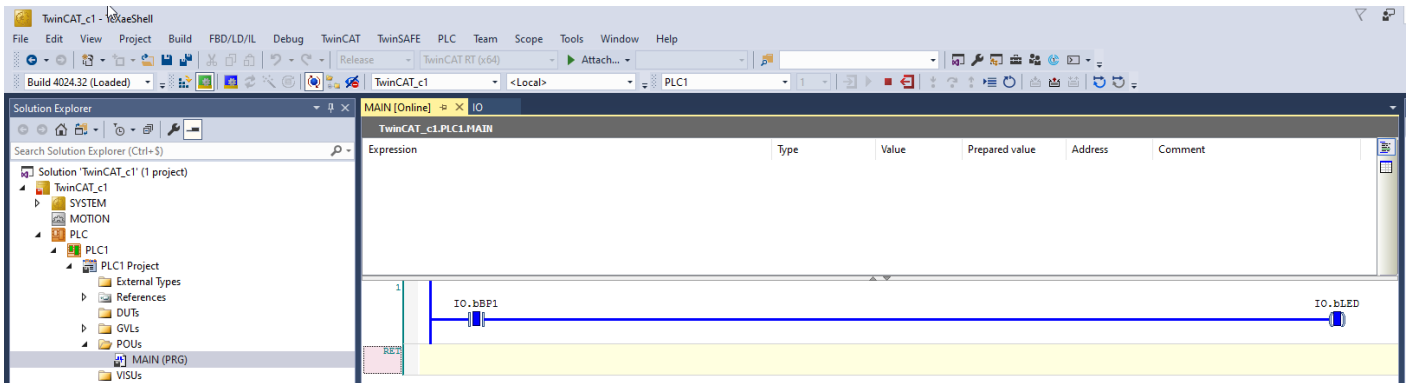
On fait :

- Login,
- puis Yes,
- puis Start (F5)

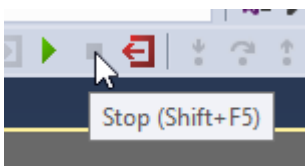




On remarque que l'activation de l'interrupteur provoque l'allumage de la LED. La désactivation de l'interrupteur provoque l'extinction de la LED. Notre programme est opérationnel et la communication avec le module d'Entrées / Sorties déportés est fonctionnelle.



Pour stopper le Runtime, faire Stop (Shift+F5) et pour sortir faire Log Out.



Conclusion

Nous venons de réaliser notre premier programme sous TwinCAT permettant de communiquer avec un module d'Entrées/Sorties déportées avec le protocole EtherCAT et le logiciel TwinCAT.

EtherCAT permet de hautes performances temporelles, comme par exemple, d'avoir une période de rafraîchissement de 100 μ s pour un ensemble de 100 axes motorisés. Pour atteindre ces performances, il est nécessaire d'être en mode Real-Time avec le hardware IPC adapté à la charge de calcul.

Revision #1

Created 4 July 2023 09:23:33 by Philippe Celka

Updated 4 July 2023 09:24:13 by Philippe Celka