

TwinCAT 3 & IO-Link IFM

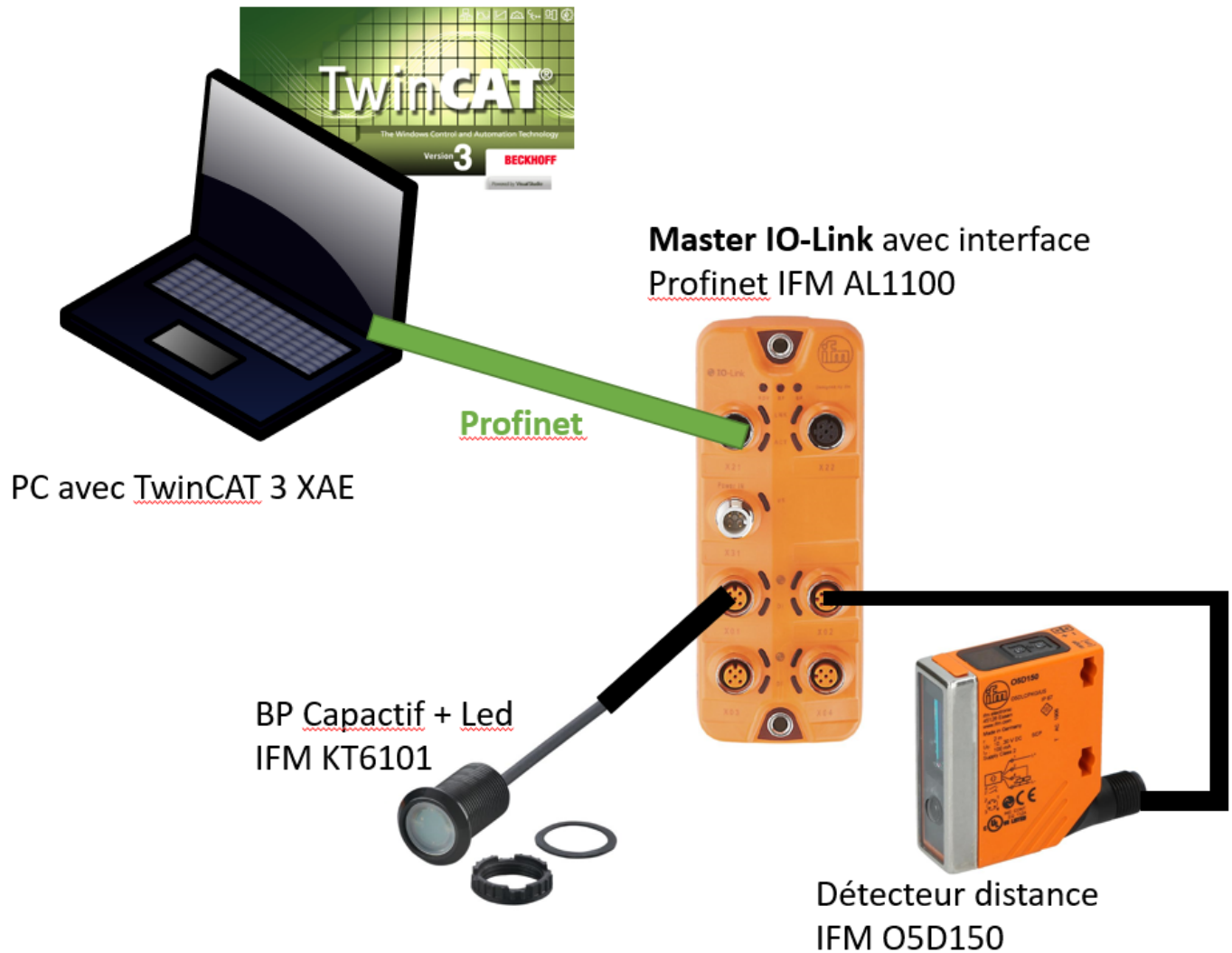
Projet TwinCAT 3 IO-Link

IO-Link est un système de communication pour la connexion de capteurs et d'actionneurs intelligents normalisé IEC 61131-9. Le système se compose d'un maître IO-Link et d'un périphérique IO-Link : un capteur, un actionneur ou une combinaison des deux. Le maître IO-Link peut être une passerelle vers un bus de plus haut niveau tel que PROFINET, EtherCAT ...

Le matériel mis en oeuvre dans cet article :

- Un PC avec l'environnement TwinCAT 3
- Master IO-Link Profinet d'IFM AL1100
- Bouton Capacitif KT6101 d'IFM : communique en IO-Link
- Détecteur Laser O5D150 d'IFM : communique en IO-Link

Le Runtime TwinCAT constituera notre PLC et va gérer la communication avec le Master IO-Link Profinet. Le schéma suivant résume le montage :



Les appareillages Profinet s'intègrent "naturellement" dans l'éco-système Siemens mais il est également possible d'associer ce matériel à TwinCAT 3, où classiquement, on utilise la communication EtherCAT. La procédure sera très proche de la mise en oeuvre d'un coupleur Ethercat vu dans l'article précédent, mais nécessitera de configurer manuellement le matériel.

Configuration du Master IO-Link

Le logiciel Moneo d'IFM

On utilisera le logiciel Moneo dont une licence est fournie avec le Kit IO-Link d'IFM.

Il permet de configurer :

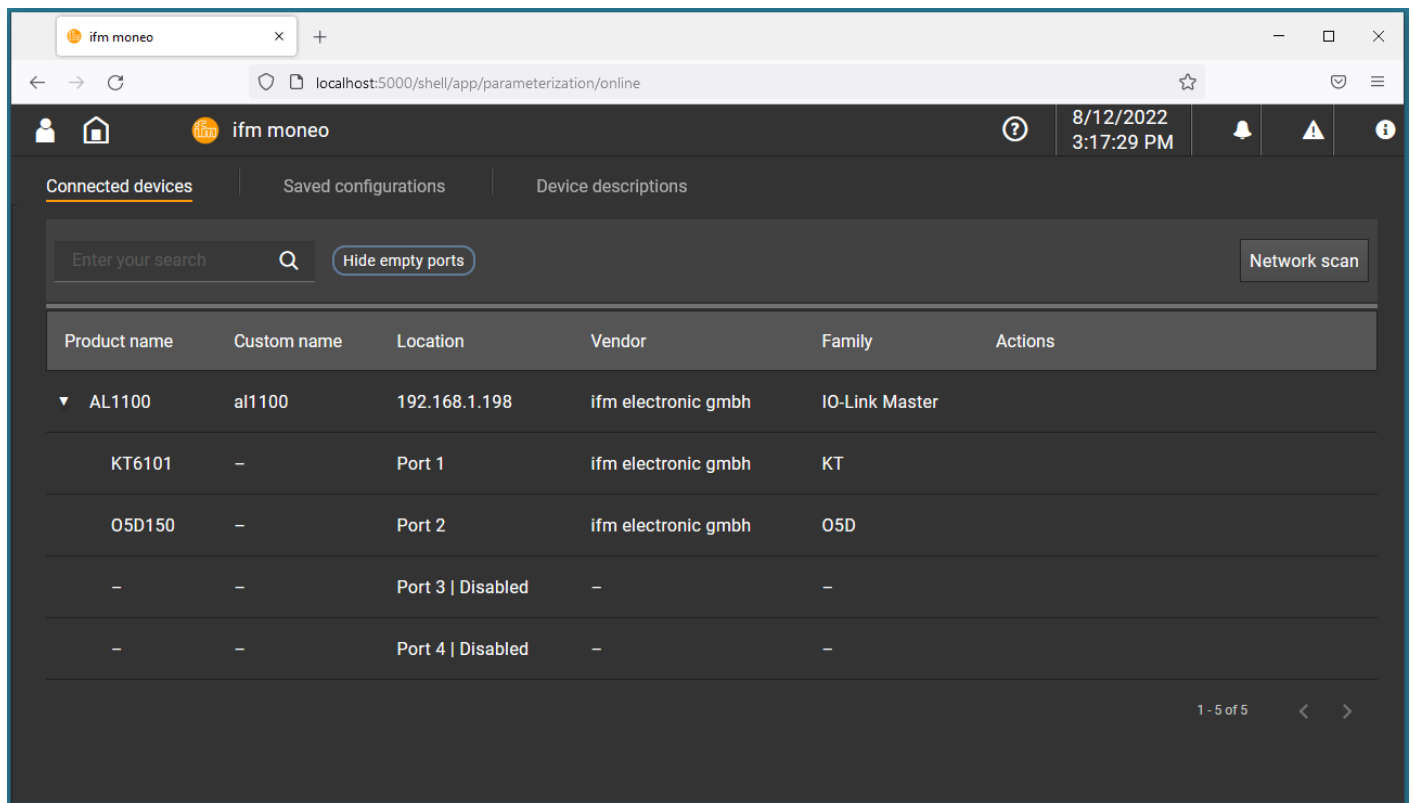
- le Master IO-Link (AL1100)
- les capteurs IO-Link associés
 - le bouton capacitif KT6101

- le détecteur de distance O5D150

La configuration réseau choisie pour le Master est :

- @IP : 192.168.1.198
- Netmask : 255.255.255.0
- Gateway : 192.168.1.1

Il est nécessaire que la carte réseau de votre PC soit sur la même plage d'adresse que le Master IO-Link. Dans cet exemple, l'@IP du PC est réglée à 192.168.1.44.



Le détecteur de distance O5D150

Pour le détecteur de distance O5D150, il n'y a grand chose à vérifier, la distance de détection est réglée à 100 cm, quand la distance mesurée est inférieure à 100 cm, la sortie SwitchState vaut 'True' et passe à 'False' quand la distance dépasse 100 cm. La plage de mesure du détecteur est comprise entre 5 cm et 200 cm.

ifm moneo

localhost:5000/shell/app/parameterization/online/42eeb49b-99d3-4618-aaf8-a760657c1424

ifm moneo 8/12/2022 3:19:10 PM

< BACK Connected: al1100 (192.168.1.198) / Port 2 / O5D150

Identification Parameter

Application Specific Tag: ***


Product name: O5D150 Serial number: 000008881411

Family: O5D Hardware / firmware revision: AD / 343

Vendor: ifm_electronic_gmbh

Vendor ID: 0x136 | 310 d | Bytes 1d 54d Description: Optical distance sensor. Visible laser light, protection class 1 laser. Connection: 2 switching outputs normally open / closed complementary DC PNP

Device ID: 0x174 | 372 d | Bytes 1d 116d



Device site

Source	Parameter name	Value	Current device value	Minimum	Maximum	Description
Identification	Application-specific Tag	***	***	0	16	Possibility to mark a device with user- or application-specific information.
Parameter						

WRITE TO DEVICE SAVE PARAMETER SET DISCARD

Le bouton capacitif KT6101

Pour le bouton capacitif, il y a plusieurs modes de fonctionnement possible

- dynamique,
- static,
- bistable.

plusieurs choix de couleurs de fonctionnement, de réglage de sensibilité, etc. Dans ce contexte, le mode Bistable avec deux couleurs prédéfinies est choisi:

- bistable: un appui, la sortie s'active, un second appui la sortie s'éteint.
- couleurs: bleu à l'état de repos, vert en marche.

The screenshot shows the 'ifm moneo' web interface. At the top, it indicates 'Connected: al1100 (192.168.1.198) / Port 1 / KT6101'. The 'Identification' tab is active, displaying the following details:

- Application Specific Tag: ***
- Function Tag: ***
- Location Tag: ***
- Product name: KT6101
- Family: KT
- Vendor: ifm electronic gmbh
- Vendor ID: 0x136 | 310 d | Bytes 1d 54d
- Device ID: 0x3EA | 1002 d | Bytes 3d 234d
- Serial number: 000009061788
- Hardware / firmware revision: 8 / 2019.09.12
- Description: Capacitive illuminated pushbutton, Plastic housing, Pig-Tail M12

On the right, there is an image of the device and a 'Device site' button. Below the details, a table lists parameters:

Source	Parameter name	Value	Current device value	Minimum	Maximum	Description
Identification	Application-specific Tag	***	***	0	32	Possibility to mark a device with user- or application-specific information.

At the bottom right, there are buttons: 'WRITE TO DEVICE', 'SAVE PARAMETER SET', and 'DISCARD'.

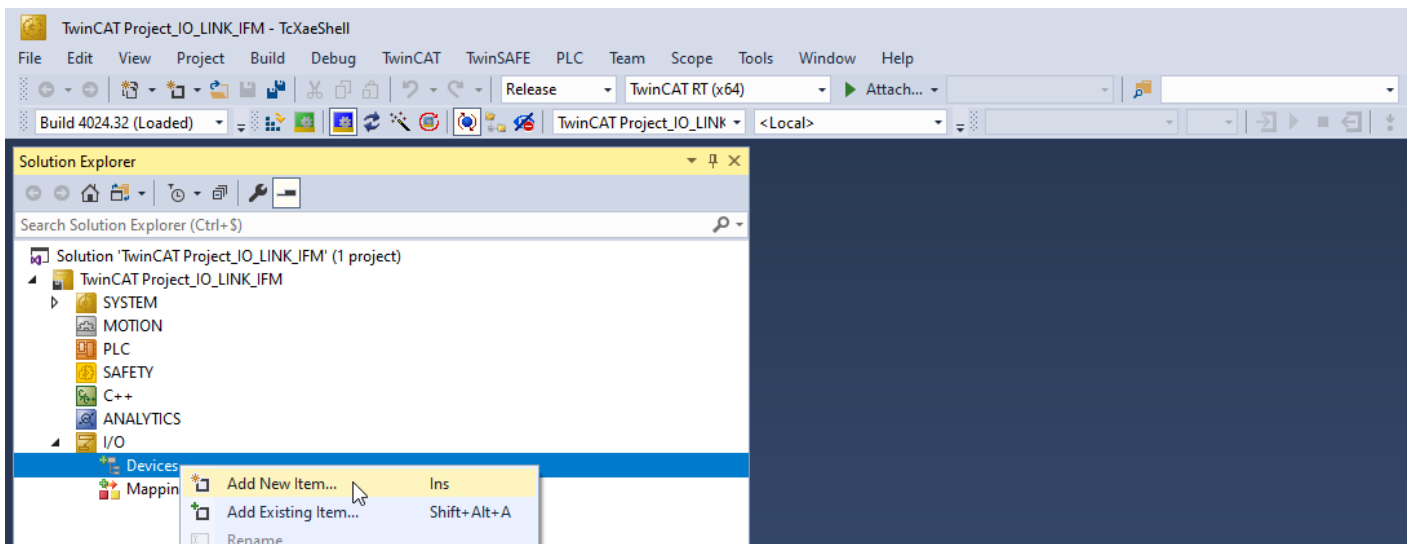
Avant de se déconnecter du logiciel Moneo, on n'oublie pas de faire un write des modifications effectuées.

Association du Master IO-Link à TwinCAT

Mise en place de la liaison Profinet

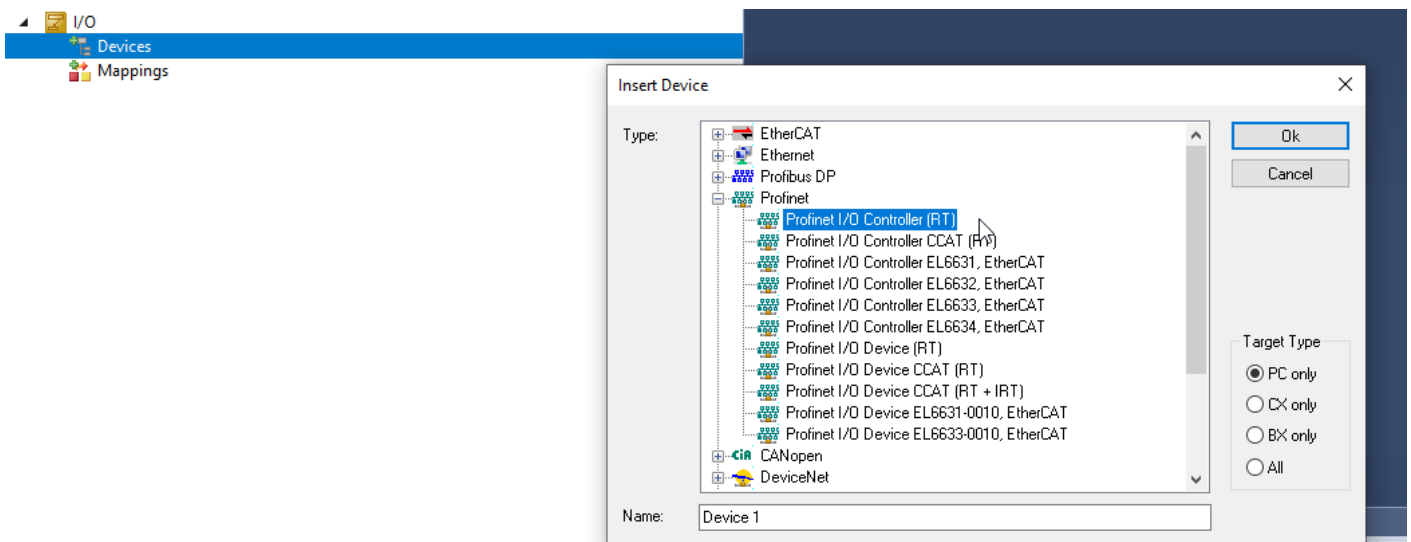
Après avoir crée un projet TwinCat, développer l'arborescence, et faire un clic droit sur :

- I/O -> Devices -> Add New Item



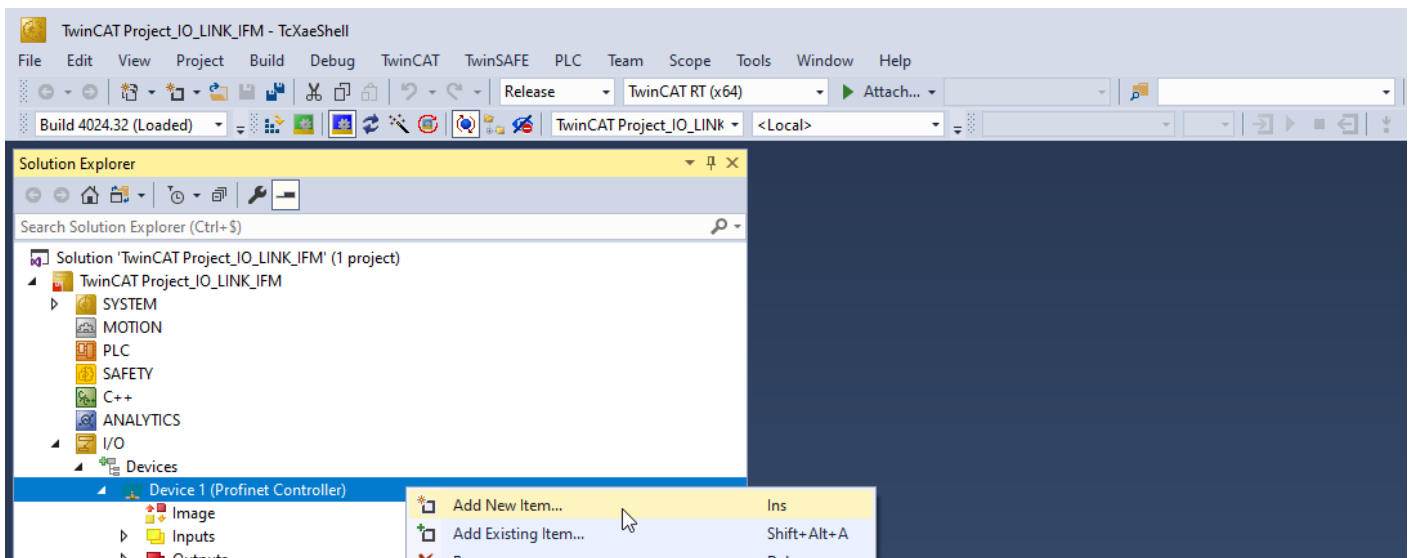
On sélectionne un Profinet I/O Controller (RT)

- on peut laisser le nom de base : Device 1



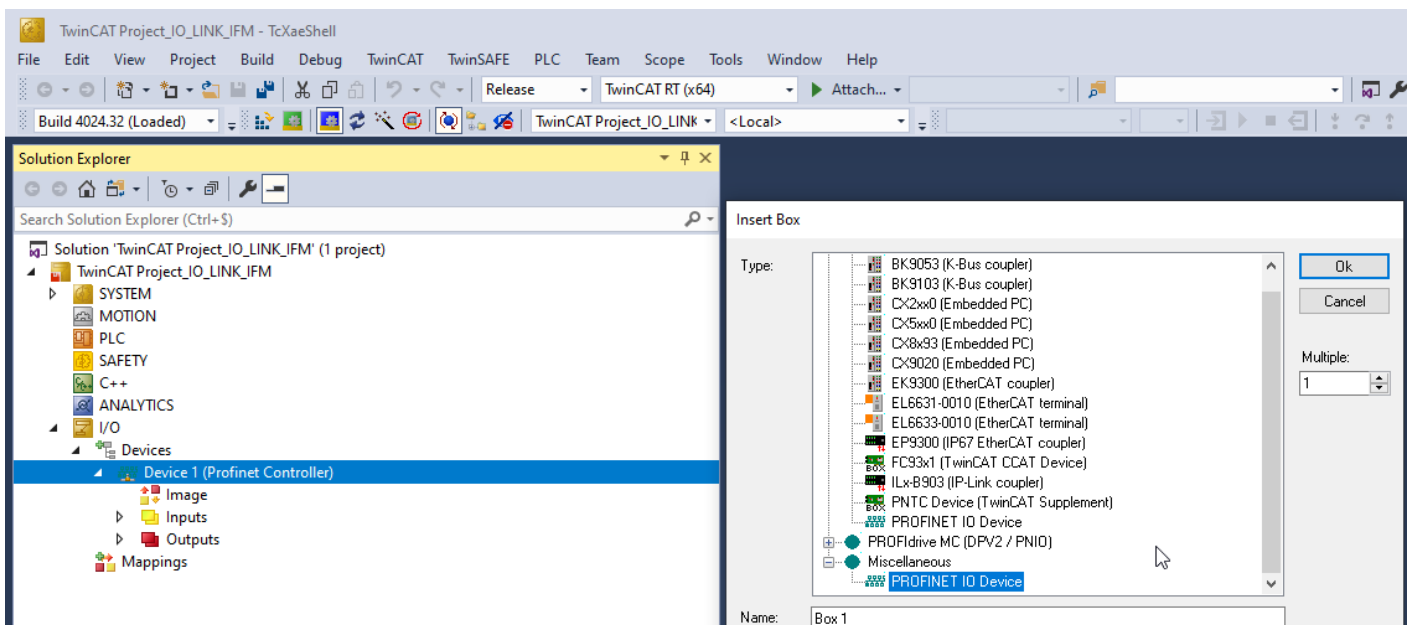
Sur Device 1, on va faire un clic droit :

- Add New Item

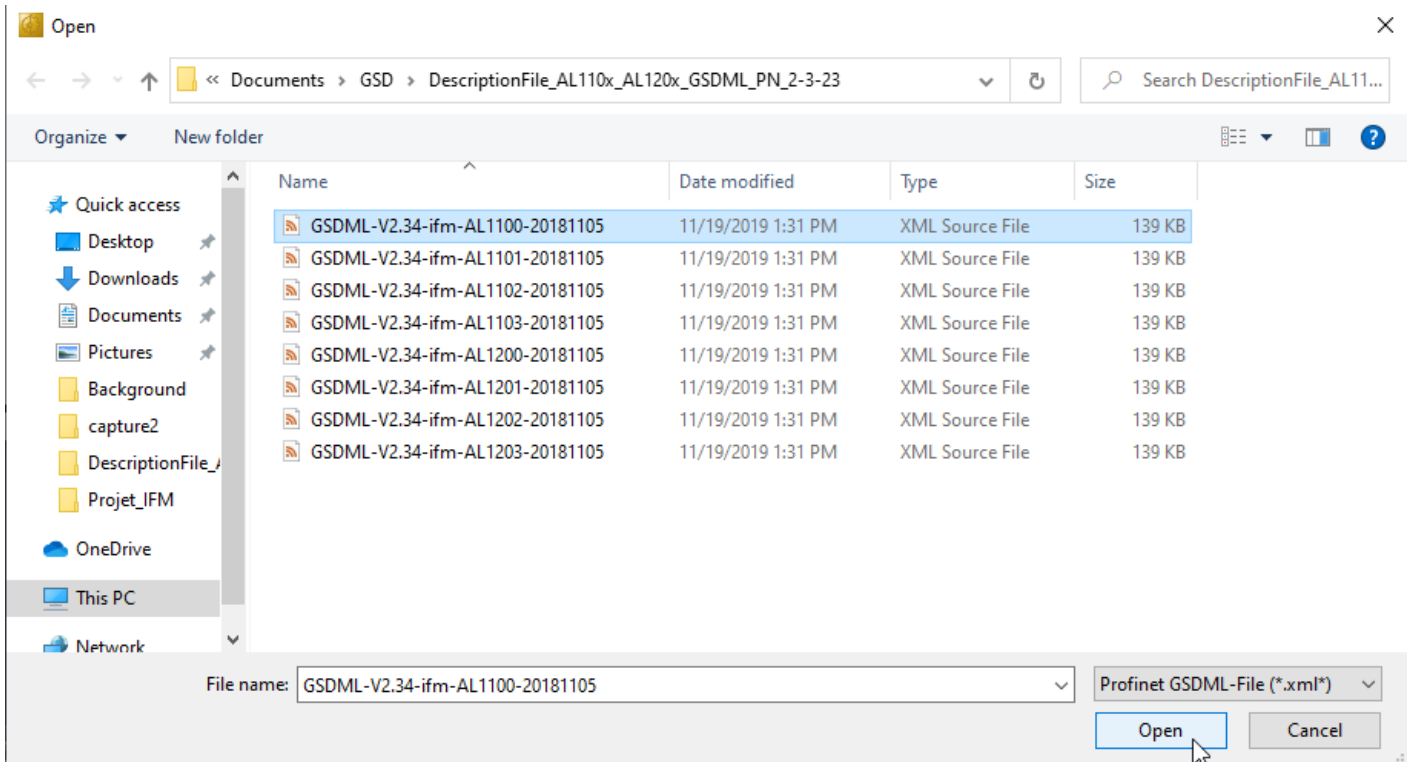


Et choisir dans Miscellaneous

- un Profinet I/O Device



A ce Profinet I/O Device, on associe le fichier GSDML du Master IO-Link Profinet AL1100. Le fichier GSDML est disponible sur le site d'IFM.

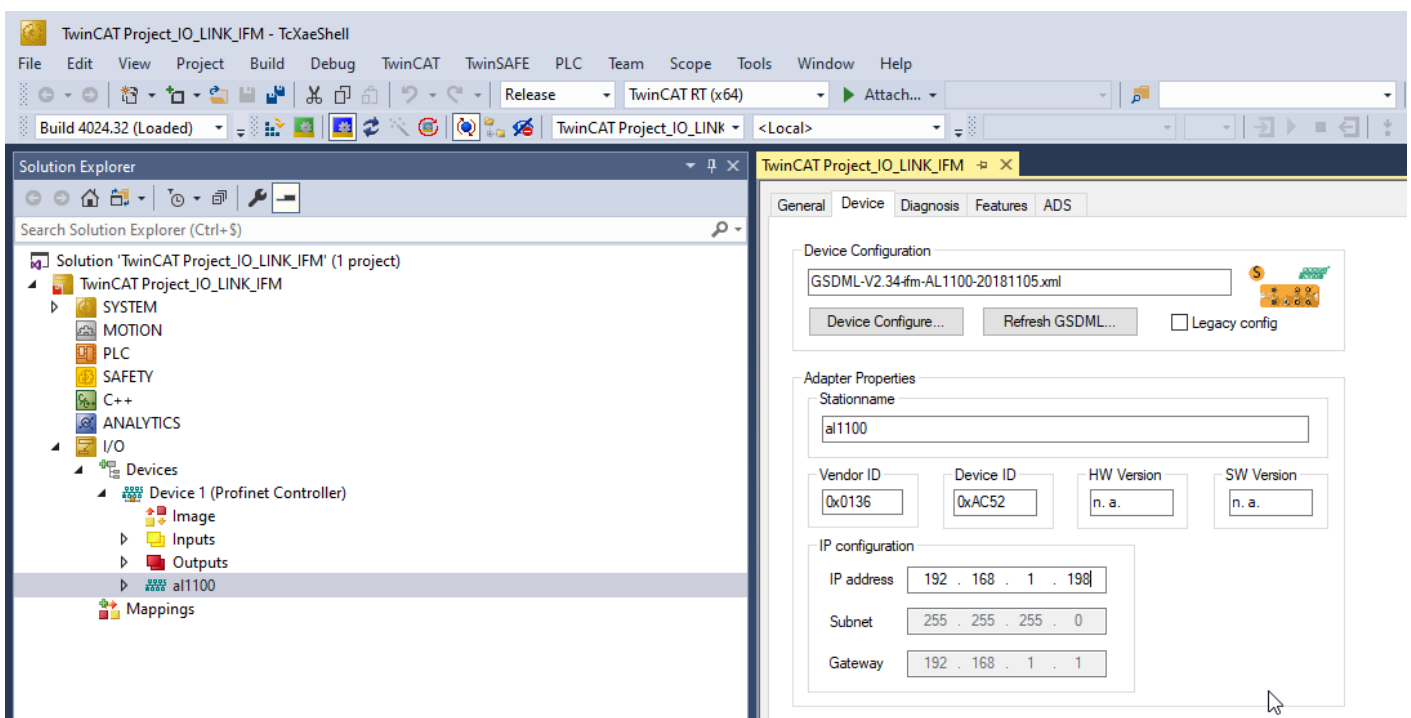


Configuration IP du Master IO-Link

Il faut modifier l'adresse IP en fonction de celle associée au Master IO-Link. Avec Moneo nous l'avions configuré à :

- @IP = 192.168.1.198

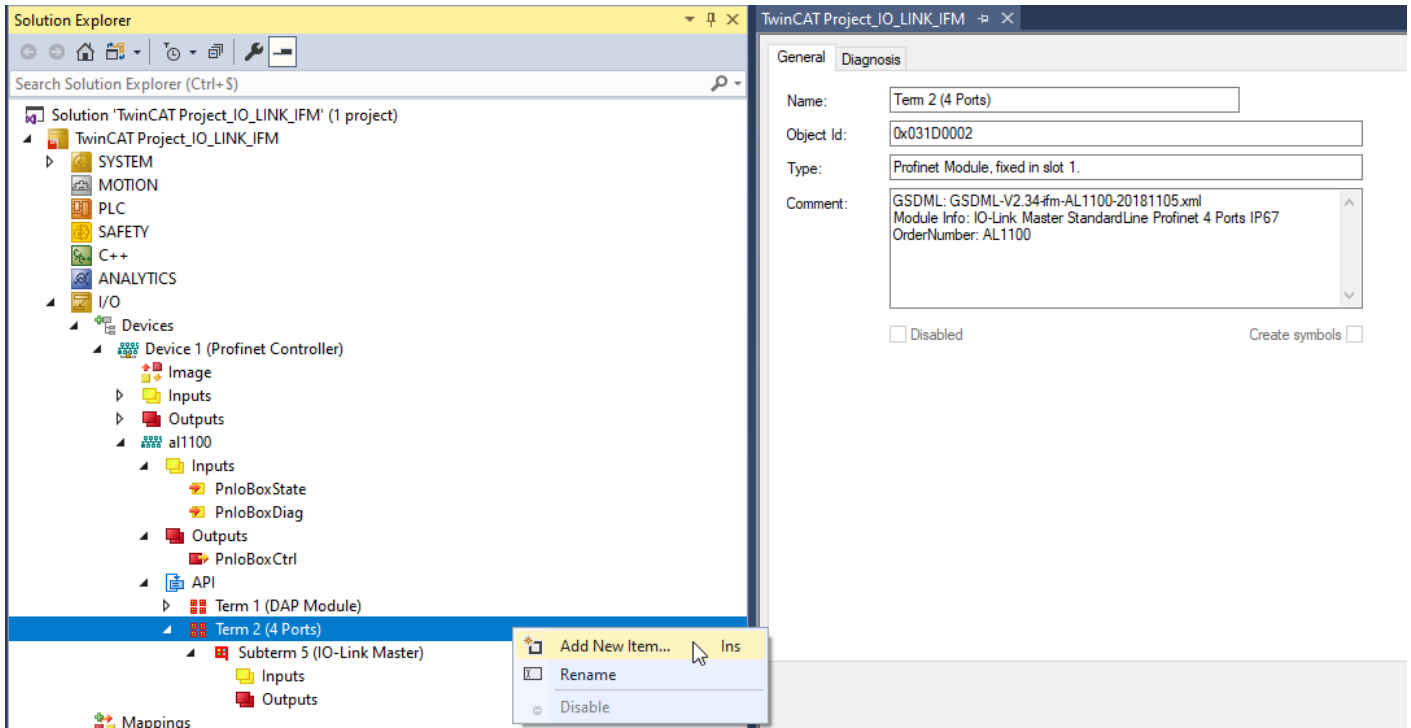
C'est cette valeur que nous allons saisir



Ajout des capteurs IO-Link

En développant l'arborescence, dans API -> Term2 (4 ports), faire :

- Add New Item



Au départ, les SubModules associés aux SubSlot 2,3,4,5 sont vides. Nous allons les configurer de la manière suivante :

- Pour le SubSlot 2 (Port 1), nous allons brancher la bouton capacitif qui nécessite 4 bytes en Input et 1 byte en output. Comme cette configuration n'existe pas, nous mettrons 4 bytes en Input, 4 bytes en Output + PQI.
- Pour le SubSlot 3 (Port 2), nous allons brancher le détecteur de distance qui nécessite 2 bytes en Input. Cette configuration existe et nous placerons 2 bytes en Input + PQI.
- Pour le SubSlot 4 (Port 3), comme aucun capteur n'est branché, nous placerons Disabled
- Pour le SubSlot 5 (Port 4), comme aucun capteur n'est branché, nous placerons Disabled

Slot	Module	SubSlot	SubModule
0	Term 1 (DAP Module)	1	Subterm 5 (IO-Link Master)
1	Term 2 (4 Ports)	2	Subterm 6 (IO-Link In/Out 4/4 Byte ...)
		3	Subterm 7 (IO-Link In 2 Byte + PQI)
		4	Subterm 8 (Disabled)
		5	Subterm 9 (Disabled)

IO-Link Master StandardLine
4 Ports

IO-Link Input + Output + PQI

IO-Link Output + PQI

IO-Link Input + PQI

IO-Link In 32 Byte + PQI

IO-Link In 16 Byte + PQI

IO-Link In 8 Byte + PQI

IO-Link In 4 Byte + PQI

IO-Link In 2 Byte + PQI

IO-Link In 1 Byte + PQI

Digital + PQI

Others

Disabled

SubModuleInfo 'Disabled':
Disabled

Configuration de la PLC

Programme Main et Fonctions

Faire un clic droit sur PLC et faire :

- Add New Item

Choisir un Standard PLC Project que nous nommerons PLC1.

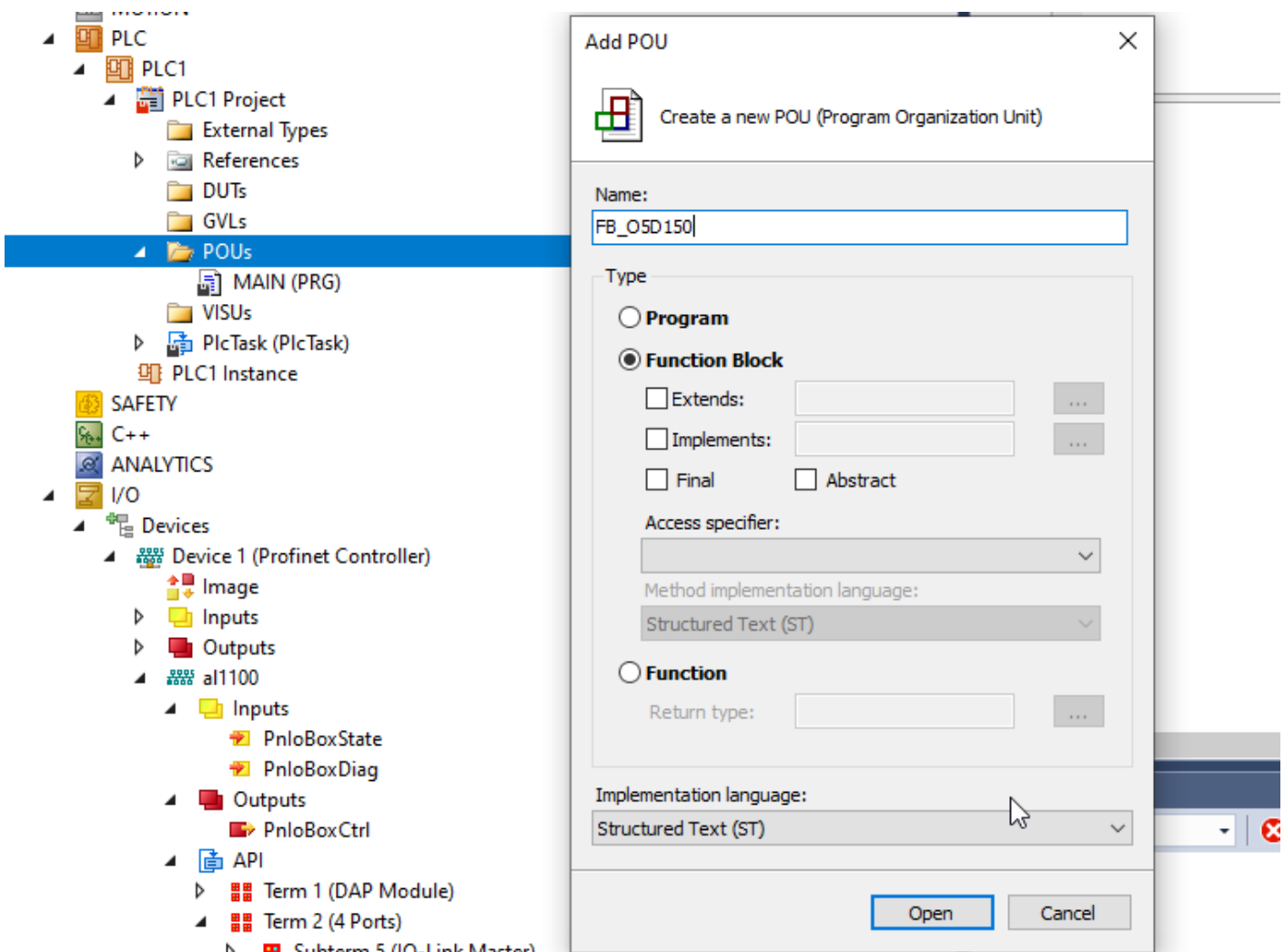
The screenshot shows the TwinCAT Project Manager interface. The 'Solution Explorer' on the left displays the project structure for 'TwinCAT Project_IO_LINK_IFM'. The 'PLC' folder is selected, and a context menu is open with the option 'Add New Item...' highlighted. The 'General' tab of the 'TwinCAT Project_IO_LINK_IFM' properties is visible on the right, showing the 'TwinCAT System Manager v3.1 (Build 4333)' and 'TwinCAT PLC Server'.

Fonction pour le capteur de distance

Nous allons dans un premier temps créer une fonction permettant de récupérer et mettre en forme des données du capteur de distance O5D150. Après avoir développé l'arborescence de PLC1, sur POU's, faire un clic droit et faire un Add POU :

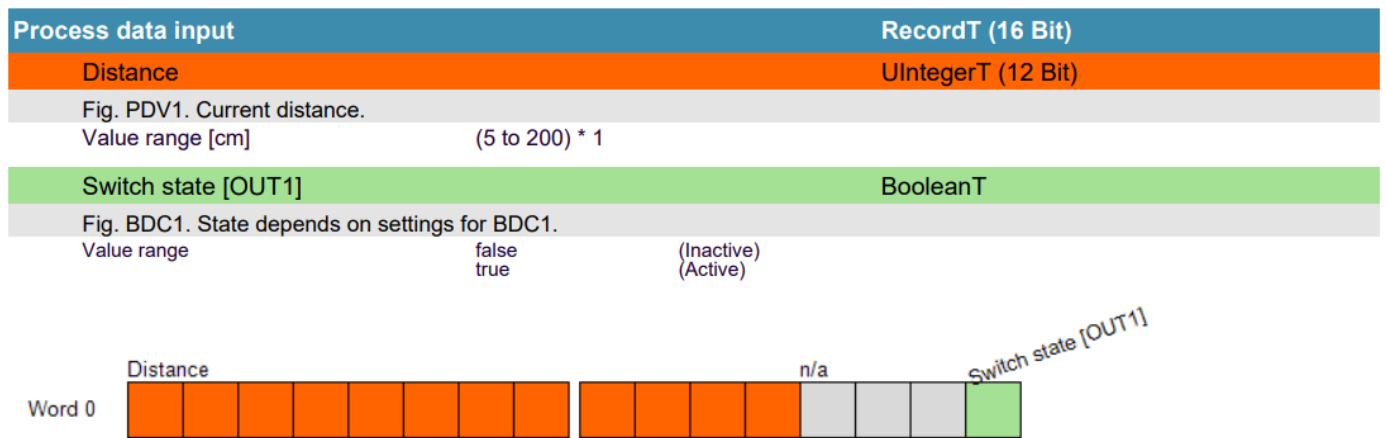
- donner le nom FB_O5D150
- dans type, choisir Function Block
- pour le langage, Structured Text (ST)

puis faire Open.



Avant de commencer à coder la fonction, il est nécessaire de comprendre le formatage des données du capteur O5D150.

6 Observation



La documentation IFM montre que deux octets (Bytes) sont utilisés pour former un mot (Word 0) de 16 bits.

Il faut faire attention à la lecture des données, dans ce cas précis :

- le premier octet contient
 - 8 bits de distance, il s'agit de l'octet de poids fort pour le calcul de la distance que l'on peut noter [x7x6x5x4_x3x2x1x0]
- le second octet que l'on peut noter [y7y6y5y4_y3y2y1y0]
 - 4 bits de poids fort [y7y6y5y4] qui constituent l'information de poids faible de la distance
 - 4 bits de poids faible
 - le bit de poids le plus faible [y0] correspond au Switch state

La distance est codée sur 12 bits et correspond au regroupement suivant :

- [x7x6x5x4_x3x2x1x0_y7y6y5y4]

Différentes opérations seront à effectuer :

- récupérer les deux octets depuis le Master IO-Link
- l'information du détecteur de distance est codé sur un bit, le Switch State [Out1]
- former à partir des deux octets le mot Word 0 sur 16 bits
- réaliser un décalage de 4 bits vers la droite pour récupérer les 12 bits de distance

Fonction O5D150 : variables

Deux variables de sortie seront créées:

- `nCurrentDistance` de type INT pour afficher la distance mesurée entre 5 et 200 cm. Le petit n devant la variable précise qu'il s'agit d'une variable de type number (nombre).

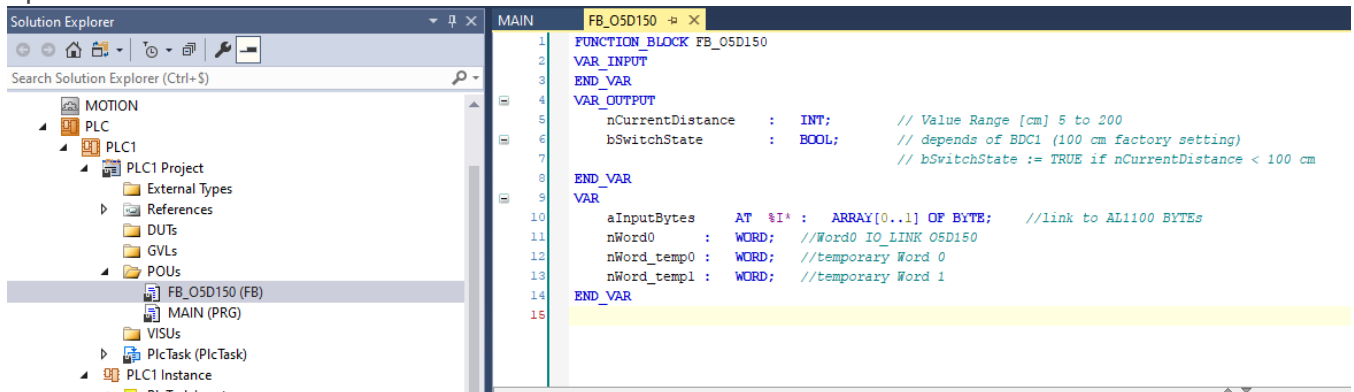
- `bSwitchState` de type `BOOL` pour afficher l'état `TRUE` ou `FALSE` de la valeur de mesure de distance (`TRUE` quand la distance mesurée est inférieure à 100 cm). Le petit `b` indique qu'il s'agit d'une variable de type booléenne.

Variables d'affectation

```
aInputBytes  AT  %I* : ARRAY[ 0..1] OF BYTE;
```

c'est les variables que nous allons utiliser pour associer les deux octets d'Input contenus dans la trame IO-Link du O5D150.

- `AT %I*` va permettre l'affectation
- `ARRAY[0..1] OF BYTE` signifie que nous avons un tableau de 2 octets
- `aInputBytes[0]` constituera dans cet exemple l'octet de poids fort
- `aInputBytes[1]` constituera l'octet de poids faible
- `nWord0` de type `Word` permettra de créer le mot de 16 bits contenant les informations de distance
- `nWord_temp0` et `nWord_temp1` de type `Word` sont des variables tampon pour exécuter les opérations



Fonction O5D150 : instantiation et appel de fonction

Pour que le fonction O5D150 puisse être utilisée par le main, il est nécessaire

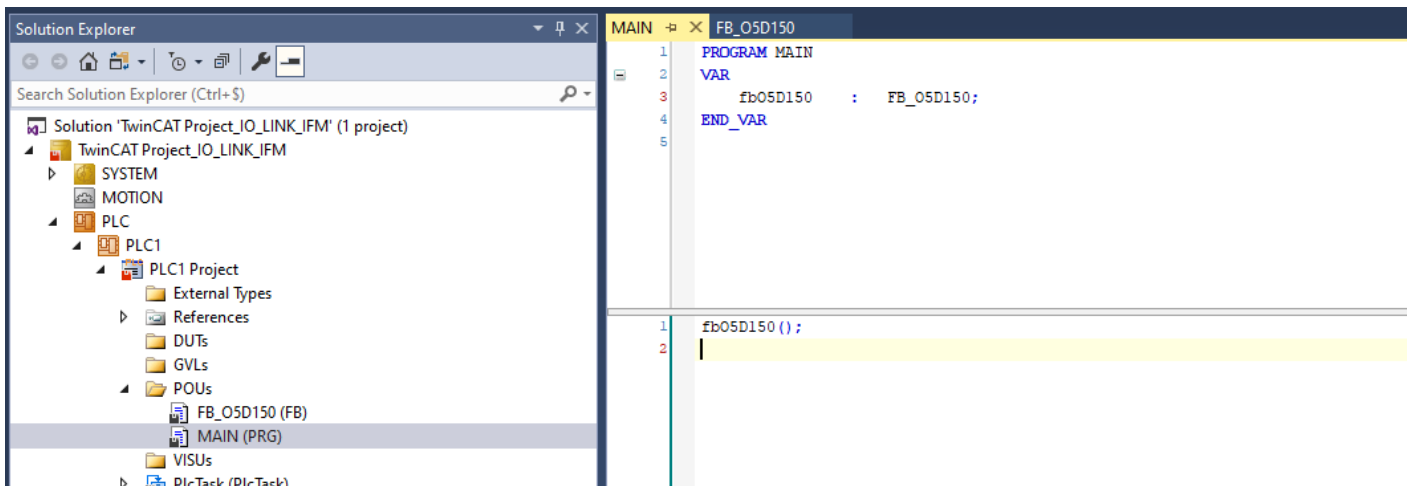
- qu'elle soit instanciée

```
VAR
    fb05D150 : FB_O5D150;
END_VAR
```

- qu'elle soit appelée dans le main

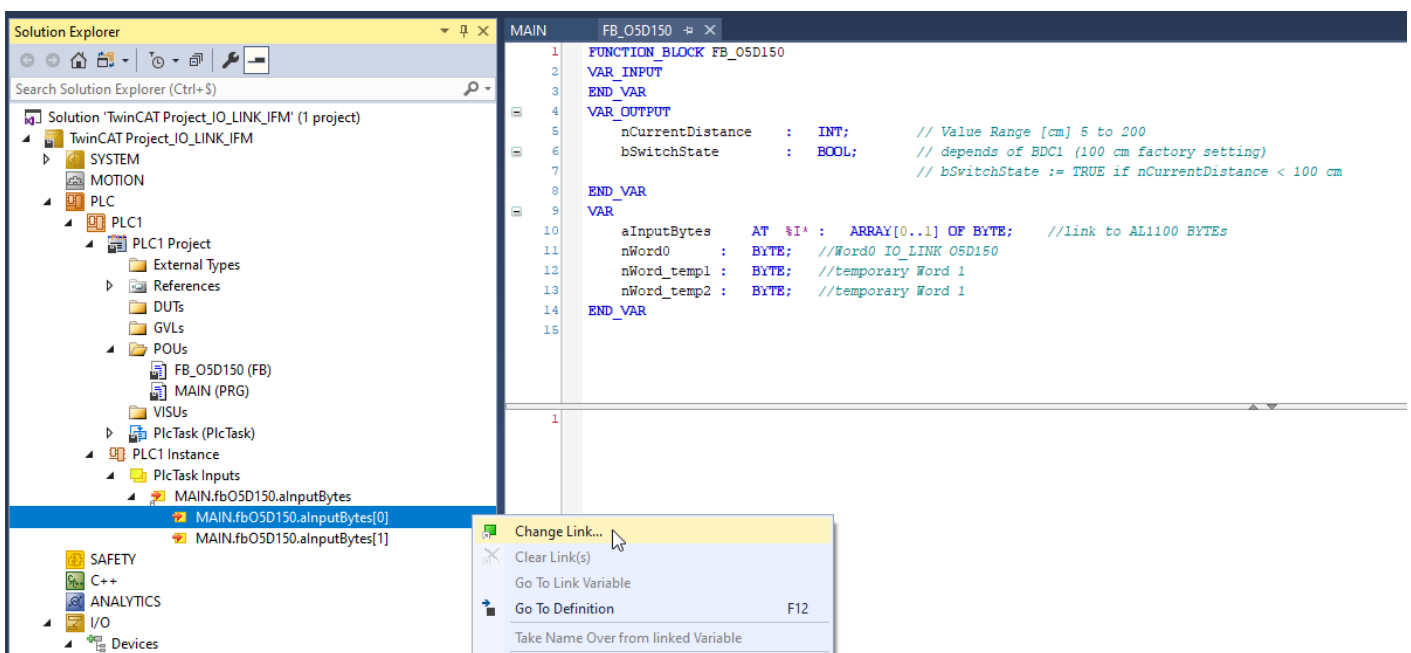
```
fb05D150( );
```

La capture d'écran ci-dessous synthétise ces actions :

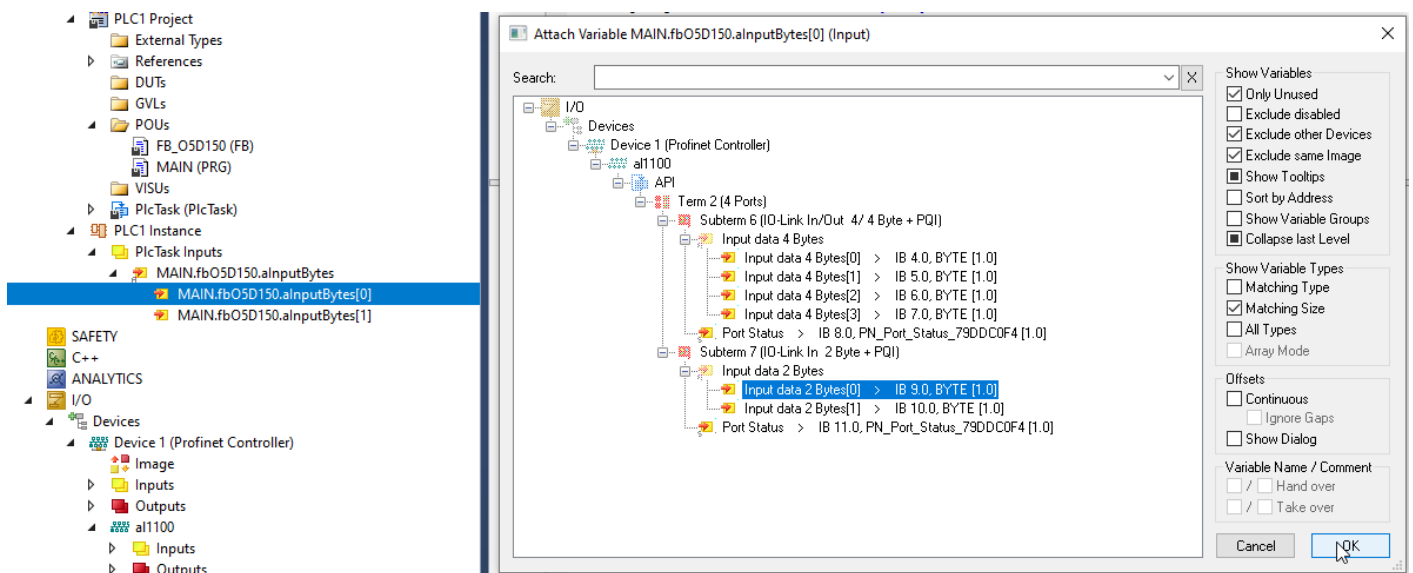


Fonction O5D150 : liaisons variables et I/O

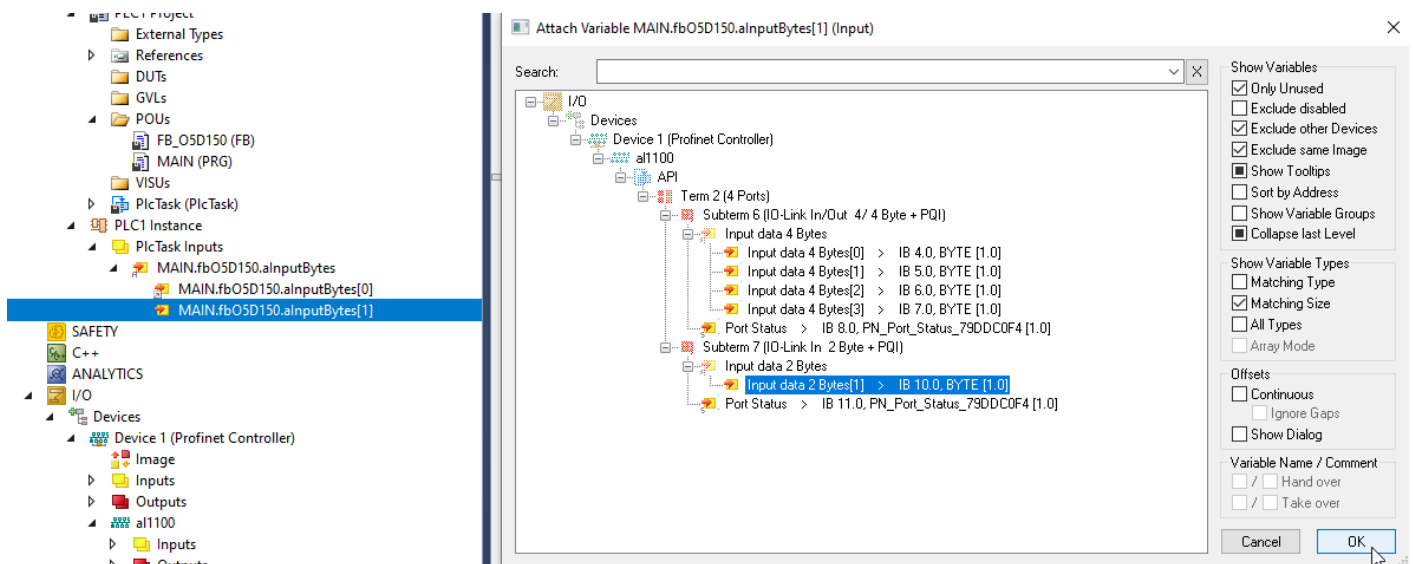
Avant de démarrer le développement du code de la fonction, il est nécessaire de créer les liaisons entre les variables physiques du Master IO-Link et les variables créées dans la fonction. Commencer par faire un activate de votre projet, suivi d'une sauvegarde et d'un build. Un onglet PLC1 Instance devrait apparaître avec les deux octets de variables d'affectation (d'instanciation) que nous venons de créer. Nous pouvons maintenant faire le Link de ces variables avec les Input du Master IO-Link.



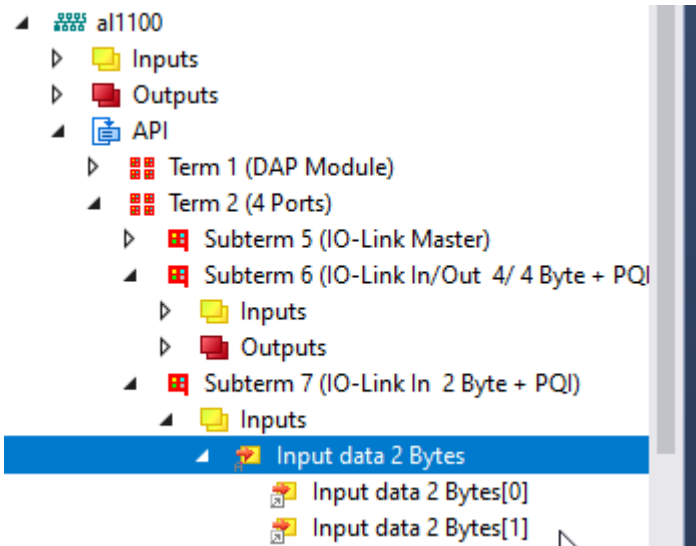
Sur la variable MAIN.fbO5D150.alnputBytes[0], faire Change Link et associer la variable Input data 2 Bytes[0] associé à la trame IO-Link du capteur O5D150.



Sur la variable MAIN.fbO5D150.alnputBytes[1], faire Change Link et associer la variable Input data 2 Bytes[1] associé à la trame IO-Link du capteur O5D150.

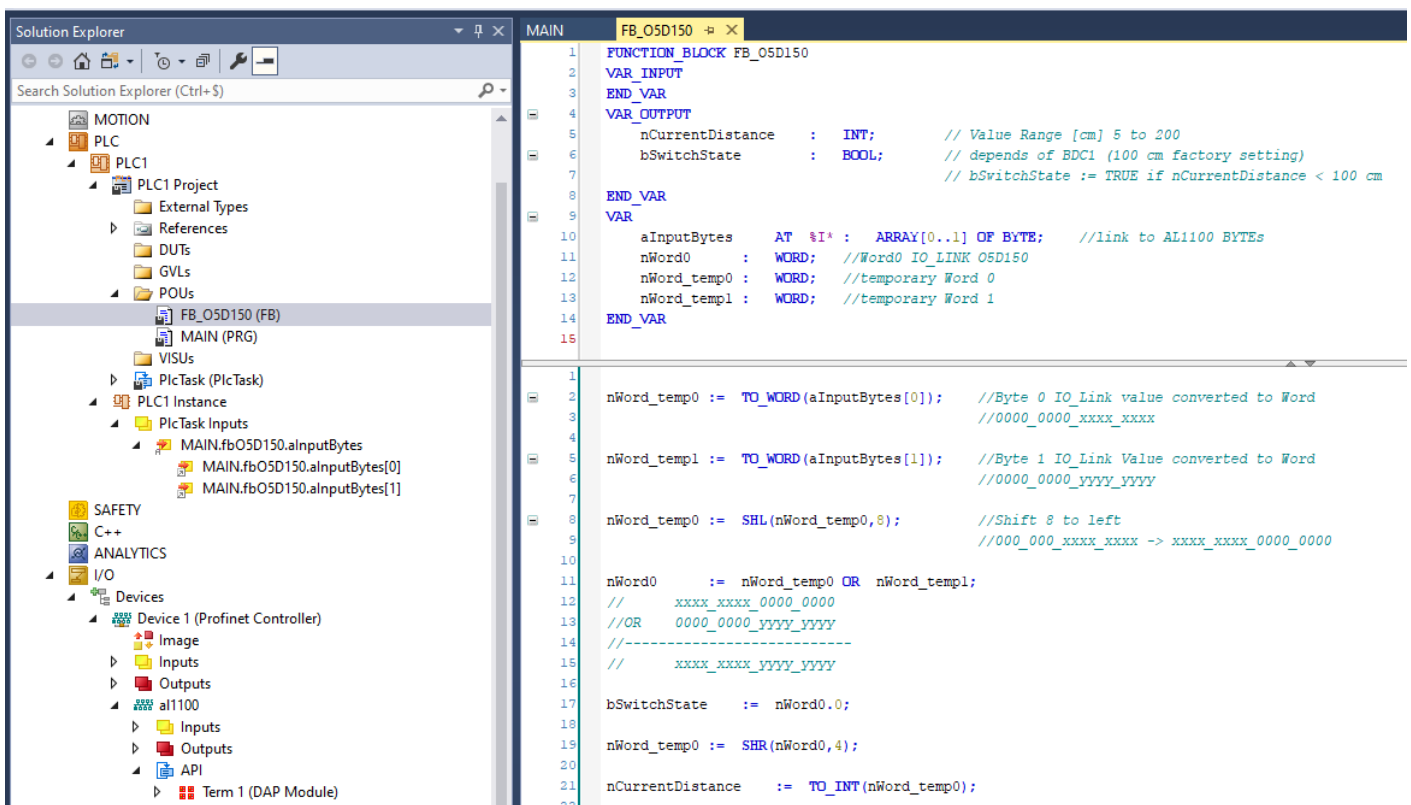


on vérifie dans l'arberescnce du al1100 que des petites apparaissent dans Input data 2 Bytes [0] [1], preuve que les liens sont pris en compte.



Fonction O5D150 : le programme

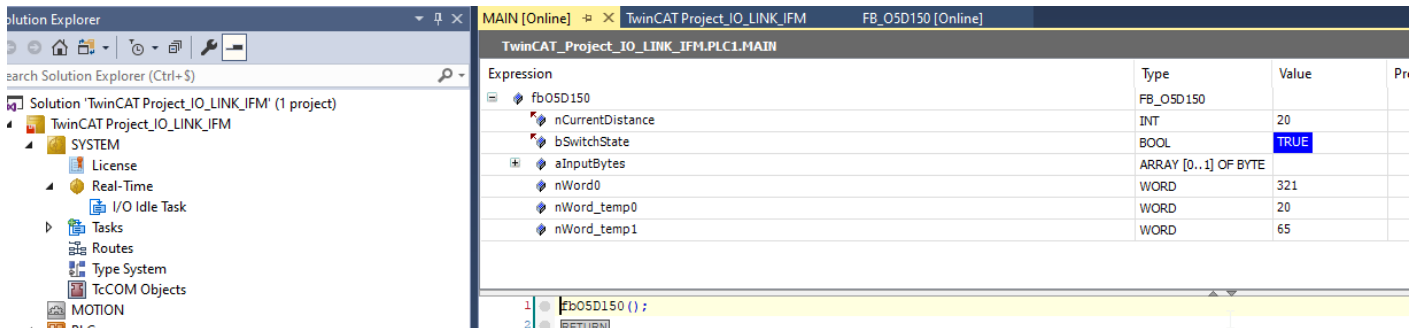
Ci-dessous, un exemple de programme permettant de construire le mot de 16 bits avec des décalages. `TO_WORD` correspond à une opération de conversion de type, ici, la variable BYTE est convertie en WORD.



Il reste à faire le build, puis cliquer sur le Mode Run (case verte avec la roue). Un message d'erreur de ce type peut apparaître. J'ai éliminé ce message en mettant un temps de cycle à 16 ms sur le cycle PLC.

The PNIO-Controller Cycle Time must be a power of 2!

OK

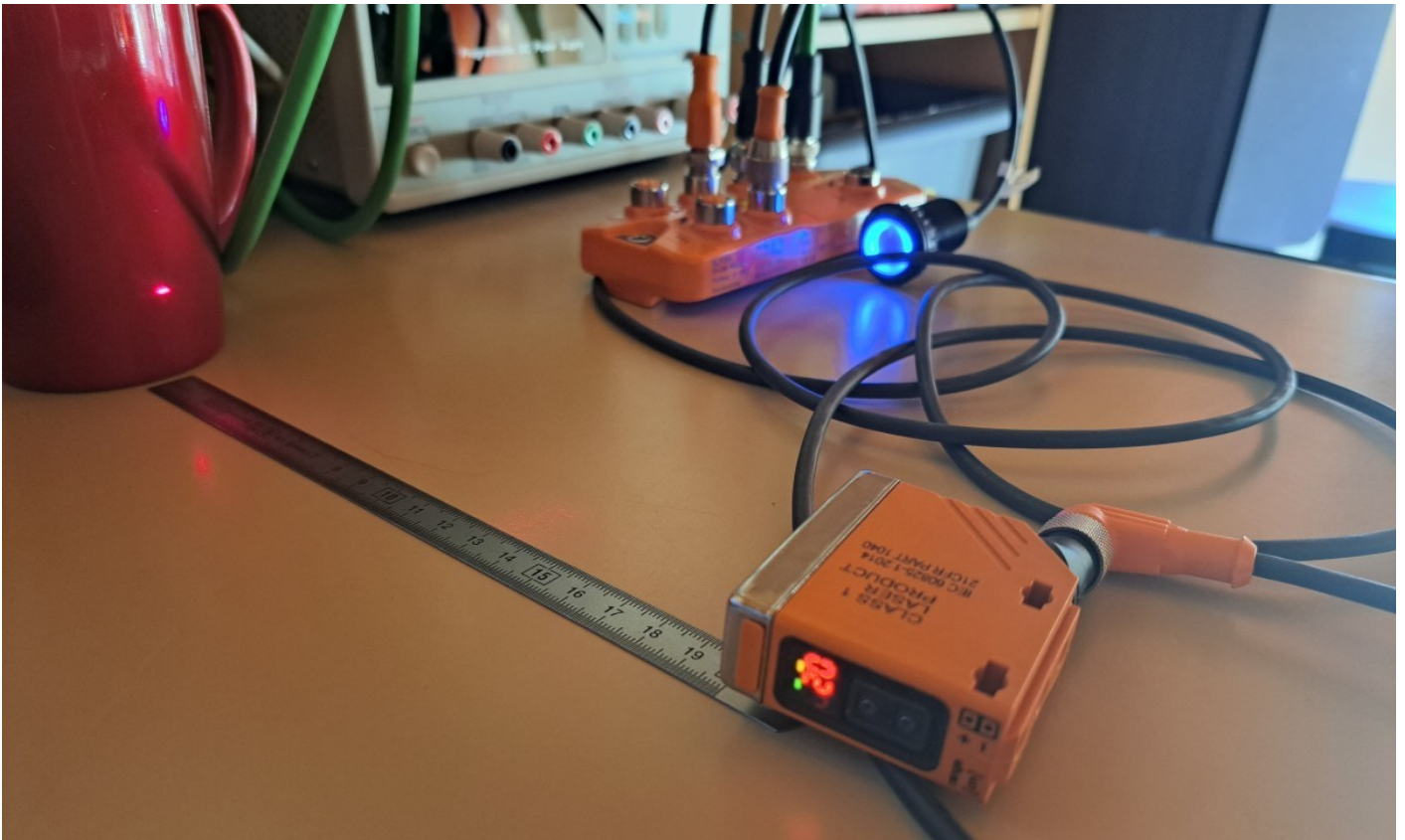


The screenshot shows the TwinCAT software interface with the 'TwinCAT_Project_IO_LINK_IFM.PLC1.MAIN' project loaded. The 'Expression' table displays the following variables and their values:

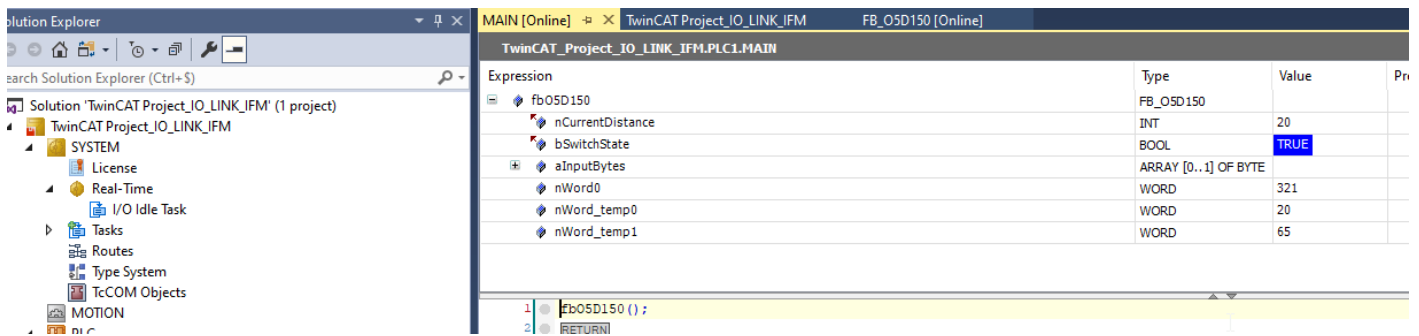
Expression	Type	Value	Pr
fbO5D150	FB_O5D150		
nCurrentDistance	INT	20	
bSwitchState	BOOL	TRUE	
aInputBytes	ARRAY [0..1] OF BYTE		
nWord0	WORD	321	
nWord_temp0	WORD	20	
nWord_temp1	WORD	65	

Below the table, the ladder logic for the 'fbO5D150' function block is visible, showing a 'RETURN' instruction.

En configuration d'essai, la distance mesurée est de 20 cm



Le programme nous renvoie bien 20 cm pour nCurrentDistance et bSwitchState est bien à TRUE car l'on est inférieur à 100 cm.

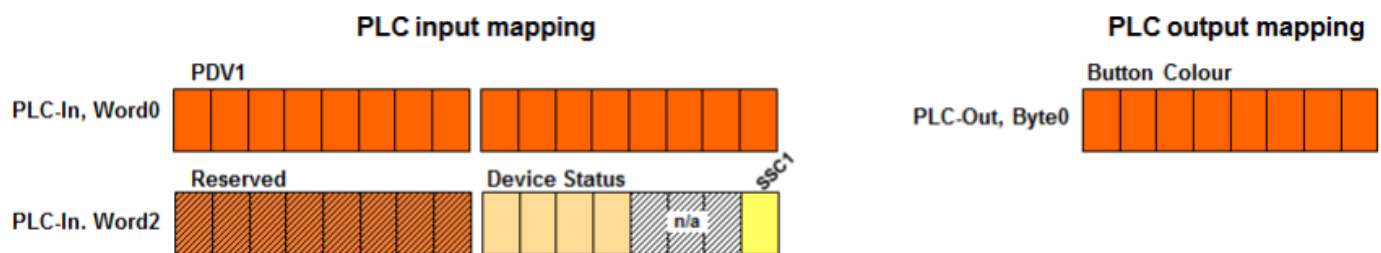


Ajout du bouton Capacitif KT6101

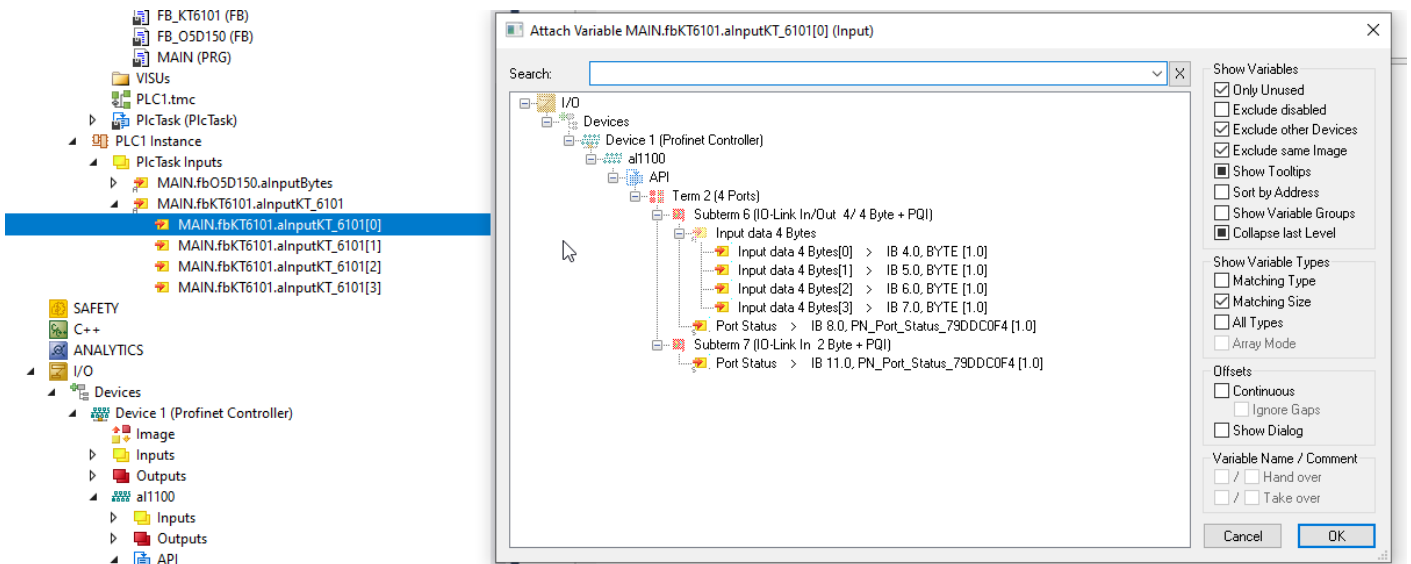
Configuration et programme

Le mapping mémoire donné par IFM pour le KT6101 est le suivant :

- 2 octets (Bytes) sont utilisés pour créer la variable PDV1.
- 1 octet pour la variable device Status, mais comme les transmissions IO-Link se font par nombre d'octets pairs, un octet supplémentaire est présent et réservé pour des développements futurs.



Les étapes sont identiques à celles présentées précédemment et ci-dessous, une proposition de code fonction pour gérer le bouton. La valeur PDV1 correspond à une valeur d'appui, avec la valeur 3000, le doigt touche complètement le bouton, en dessous de 1000, on effleure à peine.



Tests et essais

La fonction réagit correctement à l'appui du bouton.







- bSSC1 passe de False à True quand on appui sur le bouton
- nPDV1 passe de 0 à 3000 car l'appui s'est fait avec l'intégralité de la pulpe du pouce.

TwinCAT_Project_IO_LINK_IFM.PLC1.MAIN.fbKT6101		
Expression	Type	Value
nPDV1	INT	0
bSSC1	BOOL	FALSE
aInputKT_6101	ARRAY [0..3] OF BYTE	
nWord0	WORD	0

```

1
2
3 nWord0[0] := TO_WORD(aInputKT_6101[0][0]);
4 nWord0[0] := SHL(nWord0[0],8) OR TO_WORD(aInputKT_6101[1][0]);
5 nPDV1[0] := TO_INT(nWord0[0]);
6 bSSC1[FALSE] := aInputKT_6101[3][0].0 FALSE;
7

```

PLC1.tmc [TMC Editor]	TwinCAT Project_IO_LINK_IFM	FB_KT6101 [Online]  	MAIN [Online]	FB_O5D150
TwinCAT_Project_IO_LINK_IFM.PLC1.MAIN.fbKT6101				
Expression	Type	Value		
 nPDV1	INT	3000		
 bSSC1	BOOL	TRUE		
 aInputKT_6101	ARRAY [0..3] OF BYTE			
 nWord0	WORD	3000		

```

1
2
3  nWord0[3000] := TO_WORD(aInputKT_6101[0][11]);
4  nWord0[3000] := SHL(nWord0[3000],8) OR TO_WORD(aInputKT_6101[1][184]);
5  nPDV1[3000] := TO_INT(nWord0[3000]);
6  bSSC1[TRUE] := aInputKT_6101[3][1].0[TRUE];
7

```

Bilan

TwinCAT offre la possibilité d'associer "facilement" du matériel Profinet au Runtime. Il reste à vérifier :

- l'histoire du temps de cycle en puissance de 2, j'ai mis 16 ms, il faudrait faire des essais avec 8ms et 4ms. La documentation du Master IO-Link laisserait penser que c'est possible avec 4ms...
- le programme du bloc fonction est du type Quick and Dirty, il faudrait vérifier si il n'existe pas des fonctions dans TwinCAT qui permettrait d'obtenir un code plus "propre"?

TwinCAT est normalement prévu pour fonctionner avec le bus de terrain EtherCAT qui aurait permis de simplifier encore la procédure (Master IO-Link EtherCAT AL1333 d'IFM), mais il reste étonnamment convivial concernant l'intégration de protocoles tiers.