

TP N°3
LIAISON S7 PLC-PLC/ TIA PORTAL
(Service de communication s7)

I. OBJECTIF

Sur l'environnement SIEMENS TIA PORTAL V13 réaliser :

- Réseau Profibus DP
- Une liaison s7 PLC – PLC
- Instruction GET - PUT (Profinet ou Profibus)
- PROFINET IO

II. MATERIEL

1 Micro-ordinateur avec conditions requises minimales :
- <i>Processeur => Intel® Celeron® Dual Core 2,2 GHz (Ivy/Sandy Bridge)</i>
- <i>RAM => 2 GB</i>
- <i>Espace disque disponible => 20 GB</i>
- <i>Systèmes d'exploitation => Windows 7 (64 bits)</i>
- <i>Résolution de l'écran => 1024 x 768</i>
2 Automates programmables S7-1200_ CPU 1215C DC/DC/DC
2 câbles réseau Ethernet avec connecteurs rj45 croisés
En cas de Profibus DP : 2 modules de communication SIMATIC S7-1200
1 câble Profibus avec interface série RS-485

III. MODALITÉ DE FONCTIONNEMENT

1. Création d'une liaison s7 PLC – PLC

La configuration d'une liaison est obligatoire pour la communication S7. Les fonctions de communication intégrées sont appelées par le biais d'instructions dans le programme utilisateur.

On peut répartir ces instructions dans les classes suivantes :

- Instructions d'échange de données (GET, PUT, USEND, URCV, BSEND, BRCV et PRINT)
- Instructions de modification de l'état de fonctionnement (START, STOP et RESUME)
- Instructions d'interrogation de l'état de fonctionnement (STATUS, USTATUS)

- Respectivement une instruction pour interroger une liaison (CONTROL et C_CNTRL)

Ces fonctions sont proposées indépendamment du système de communication utilisé ; vous pouvez donc utiliser la communication S7 via PROFINET, Industrial Ethernet, **PROFIBUS** ou MPI.

Dans ce TP, on va utiliser la communication s7 via un réseau **Profibus**.

On va donc créer un sous-réseau Profibus. Pour ce faire, on va configurer les interfaces MPI/DP (DP pour Profibus DP) des 2 appareils en mettant le 1^{er} CPU en tant que maître et le second en esclave (figure 1).

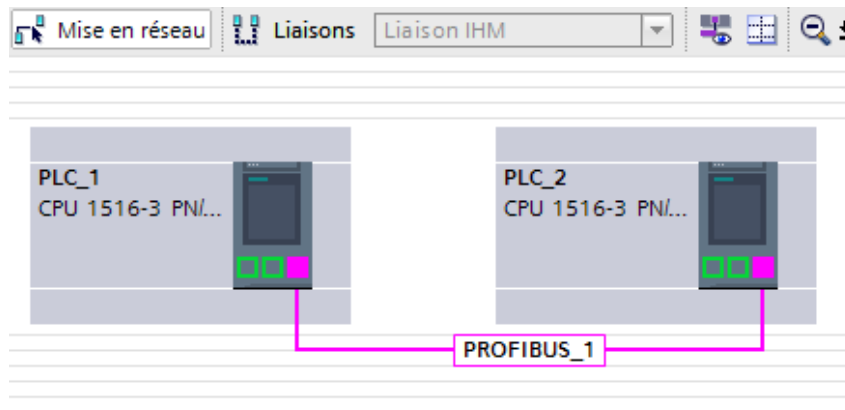


Figure 1_création d'un réseau Profibus

On a accès à **Communication S7** depuis le menu **Communication** sur le volet droit des **instructions**.

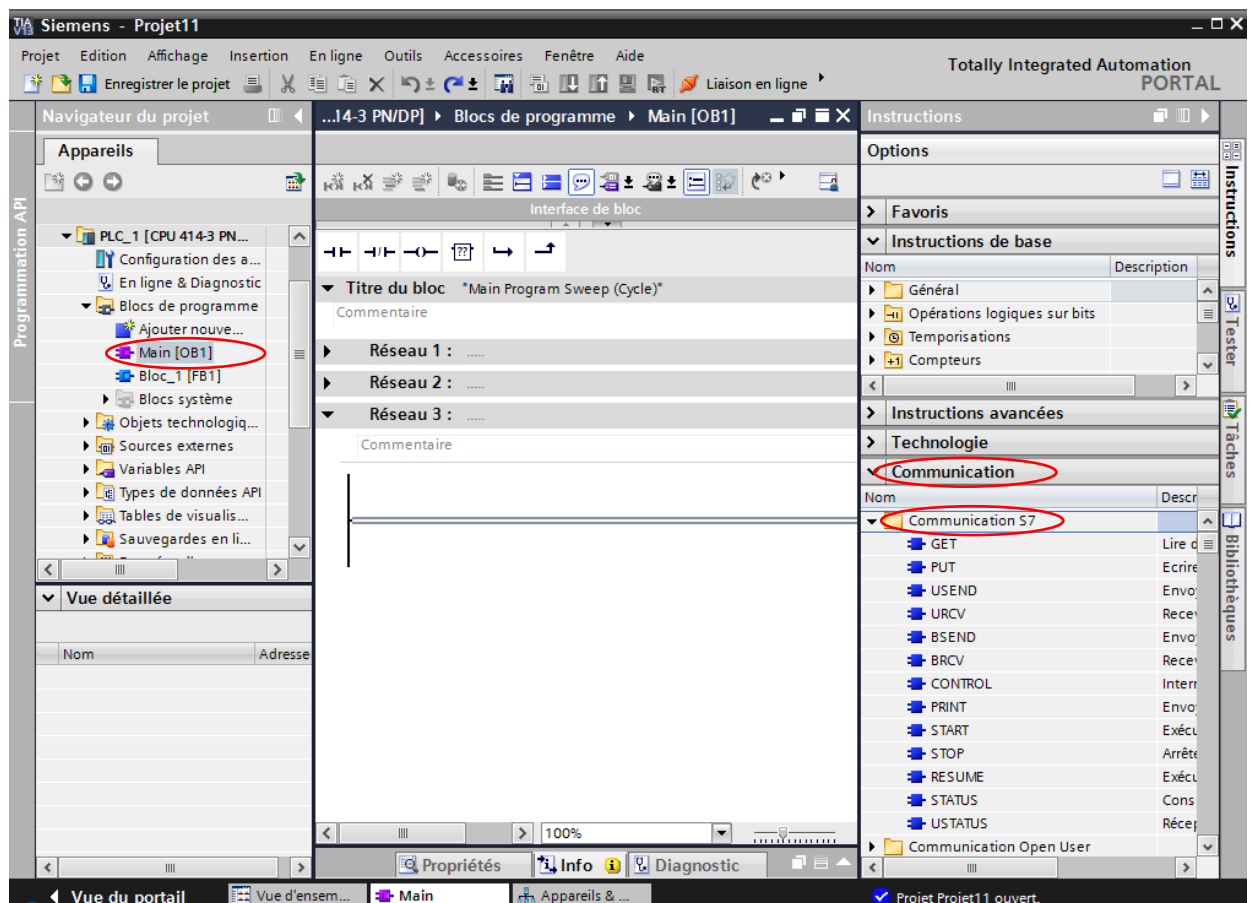


Figure 2_Communication S7

2. Instruction GET (Lire des données dans une CPU distante)

Cette instruction vous permet de lire des données dans une CPU distante lorsque la liaison ne s'effectue pas via un CP.

L'instruction est lancée en cas de front montant à l'entrée de commande REQ :

- Les pointeurs requis désignant les zones où lire les données (ADDR_i) sont envoyés à la CPU partenaire. La CPU partenaire peut être à l'état de fonctionnement Marche ou Arrêt.
- La CPU partenaire renvoie le contenu des données :
- Le dépassement de la taille maximale des données utiles par une réponse est signalé par le code d'erreur "2" au paramètre STATUS.

Lors de l'appel d'instruction suivant, les données reçues sont copiées dans les zones de réception configurées (RD_i).

Conditions requises pour l'utilisation de l'instruction

- Dans les propriétés de la CPU partenaire, la fonction "Autoriser accès via communication PUT/GET par le partenaire à distance a été activée sous "Protection".
- Les blocs auxquels vous accédez avec l'instruction "GET", ont été créés avec le mode d'accès "standard".
- Veillez à ce que les zones définies par les paramètres ADDR_i et SD_i concordent en nombre, en longueur et en type de données.
- La plage à lire (paramètre ADDR_i) ne doit pas être plus grande que la plage destinée au stockage des données (paramètre RD_i).

2.1. Paramètres

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, D, L ou constante	Paramètre de commande request ; son front montant active l'échange de données.
ID	Input	WORD	I, Q, M, D, L ou constante	Paramètre d'adressage pour indiquer la liaison avec la CPU partenaire.
NDR	Output	BOOL	I, Q, M, D, L	Paramètre d'état NDR : <ul style="list-style-type: none"> • 0: La tâche n'a pas encore été démarrée ou elle est encore active. 1: La tâche a été exécutée sans erreur.
ERROR	Output	BOOL	I, Q, M, D, L	Paramètres d'état ERROR et STATUS, signalisation d'erreur : <ul style="list-style-type: none"> • ERROR=0 STATUS a la valeur : <ul style="list-style-type: none"> ○ 0000H : ni avertissement ni erreur

STATUS	Output	WORD	I, Q, M, D, L	<ul style="list-style-type: none"> ○ <> 0000H : avertissement ; STATUS fournit des renseignements détaillés. • ERROR=1 <p>Une erreur s'est produite. STATUS fournit des renseignements détaillés sur le type d'erreur.</p>
ADDR_1	InOut	REMOTE	I, Q, M, D	Pointeur désignant les zones à lire dans la CPU partenaire
ADDR_2	InOut	REMOTE		Quand le pointeur REMOTE accède à un DB, il faut toujours spécifier ce DB.
ADDR_3	InOut	REMOTE		Exemple : P#DB10.DBX5.0 octet 10.
ADDR_4	InOut	REMOTE		
RD_1	InOut	VARIANT	I, Q, M, D, L	Pointeur désignant les zones de la propre CPU où stocker les données lues.
RD_2	InOut	VARIANT		
RD_3	InOut	VARIANT		
RD_4	InOut	VARIANT		

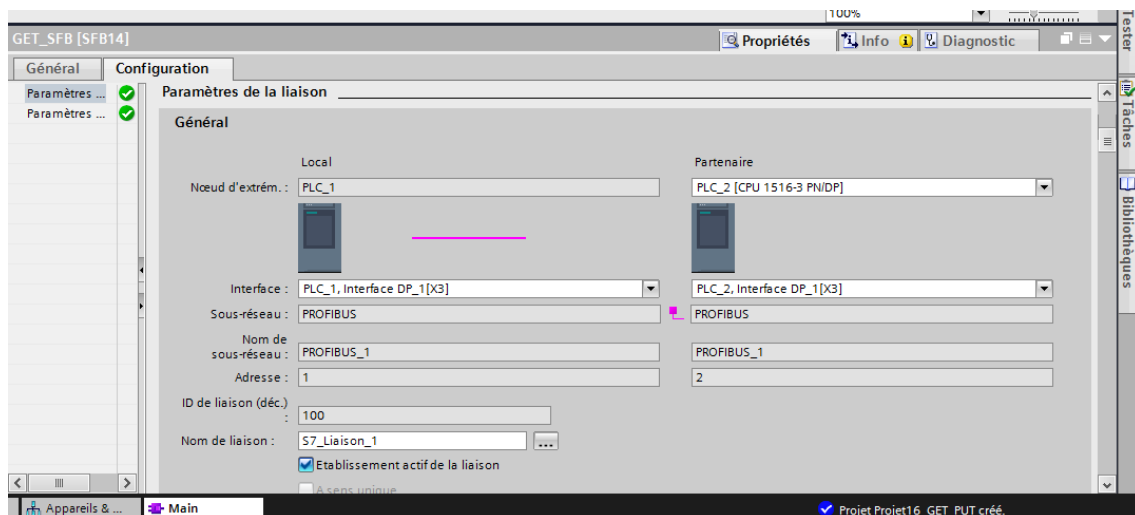


Figure 3_Liaison GET

On demande d'insérer le block GET et de faire le paramétrage selon la figure 4.

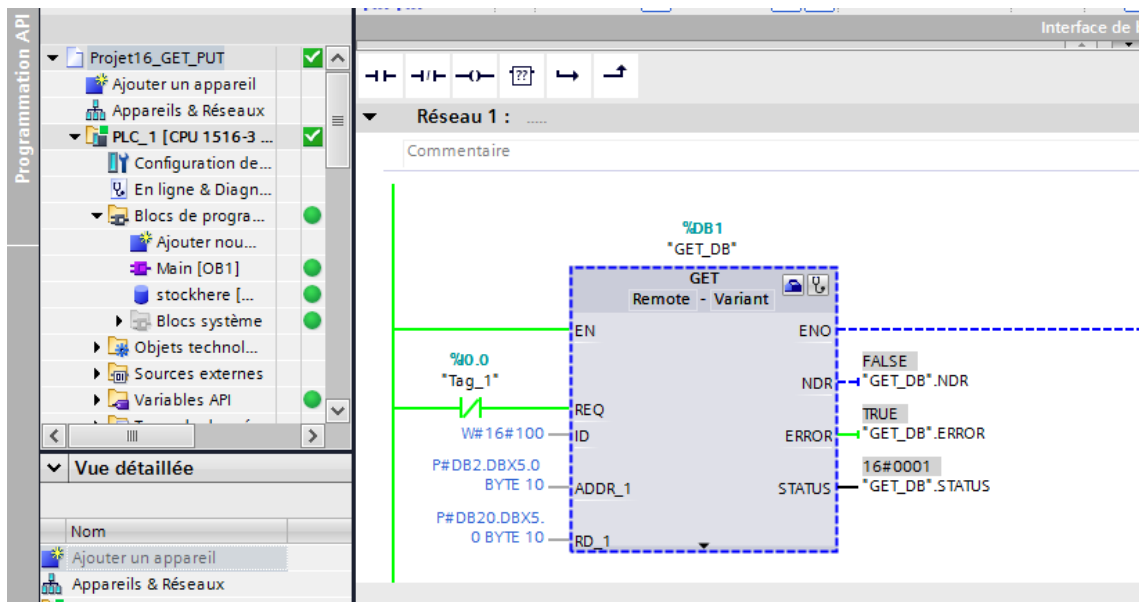


Figure 4_DB GET

3. Instruction PUT (Écrire des données dans une CPU distante s7-300/400)

S7-300 : L'envoi de données a lieu après un front montant à l'entrée REQ. Les paramètres ID, ADDR_1 et SD_1 sont pris en compte à chaque front montant de REQ. A la fin d'une tâche, vous pouvez affecter de nouvelles valeurs aux paramètres ID, ADDR_1 et SD_1.

S7-400 : L'instruction est lancée en cas de front montant à l'entrée de commande REQ. Les pointeurs désignant les zones où écrire les données (ADDR_i) et les données (SD_i) sont envoyés à la CPU partenaire.

Le partenaire distant stocke les données envoyées aux adresses indiquées et émet un message d'acquiescement en retour. Veillez à ce que les zones définies par les paramètres ADDR_i et SD_i concordent en nombre, en longueur et en type de données.

3.1. Paramètres

Paramètre	Déclaration	Type de données	Zone de mémoire	Description
REQ	Input	BOOL	I, Q, M, D, L	Paramètre de commande request ; son front montant active l'échange de données.
ID	Input	WORD	M, D ou constante	Paramètre d'adressage ID
DONE	Output	BOOL	I, Q, M, D, L	Paramètre d'état DONE : <ul style="list-style-type: none"> 0: La tâche n'a pas encore été démarrée ou elle est encore active. 1: La tâche a été exécutée sans erreur.
ERROR	Output	BOOL	I, Q, M, D, L	Paramètres d'état ERROR et STATUS, signalisation d'erreur : <ul style="list-style-type: none"> ERROR=0 STATUS a la valeur : 0000H : ni avertissement ni erreur <> 0000H : avertissement ; STATUS fournit des renseignements détaillés.
STATUS	Output	WORD	I, Q, M, D, L	

				<ul style="list-style-type: none"> • ERROR=1 Une erreur s'est produite. STATUS fournit des renseignements détaillés sur le type d'erreur.
S7-300: ADDR_1 S7-400: ADDR_i (1 ≤ i ≤ 4)	InOut	ANY	S7-300 : M, D S7-400 : I, Q, M, T, C, D	Pointeur désignant les zones où écrire dans la CPU partenaire Remarque : Quand le pointeur ANY accède à un DB, il faut toujours spécifier ce DB (par exemple, P#DB10.DBX5.0 Byte 10). Pour la transmission des structures de données (par ex. Struct, Array), il faut utiliser le type de données CHAR, BYTE, WORD ou DWORD au paramètre ADDR.

On prend l'exemple d'un sous-réseau Profinet PN/IE_1 entre CPU 414-3 PN/DP et CPU 416-3 PN/DP:

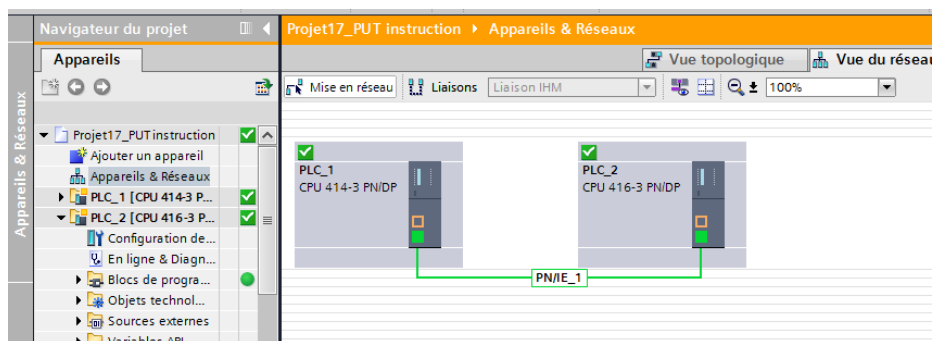


Figure 5_Sous-réseau PN/IE_1 entre CPU 414-3 PN/DP et CPU 416-3 PN/DP

On demande d'insérer le block PUT et de faire le paramétrage selon la figure 6.

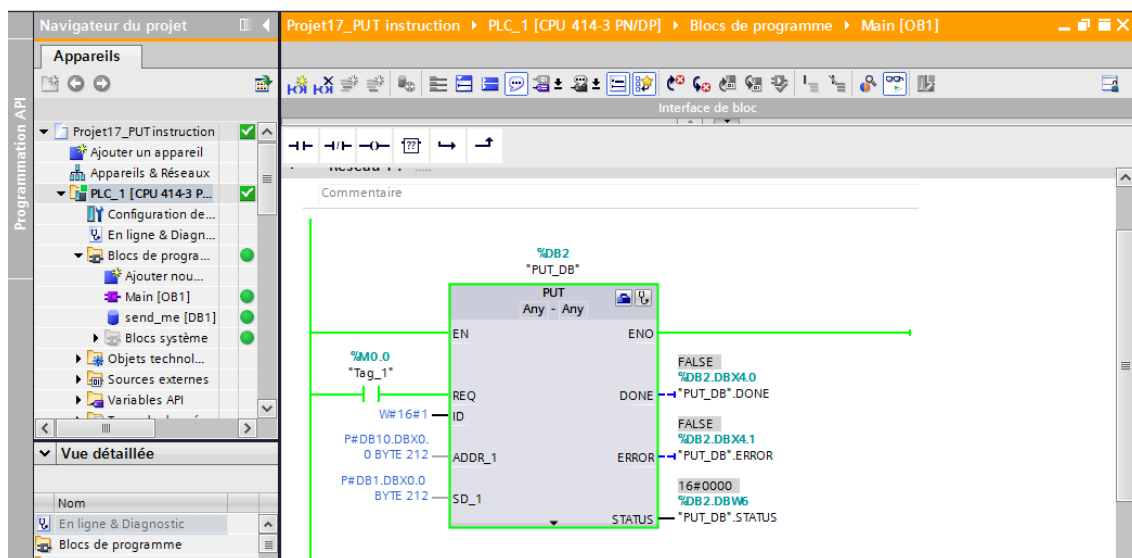


Figure 6_DB PUT

4. PROFINET IO

4.1. Créer un réseau PROFINET IO

Un réseau PROFINET IO est constitué d'un contrôleur PROFINET IO (**IO Controller**) et de ses périphériques PROFINET IO affectés (**IO Device**).

Pour créer un réseau PROFINET IO, vous avez besoin d'un contrôleur IO (p. ex. CPU 1214C) et d'un ou plusieurs périphériques IO (p. ex. un module de tête de la famille de périphérie décentralisée ET 200S).

Dès que vous connectez un contrôleur IO à un périphérique IO, un couplage contrôleur-périphérique est effectué.

Pour désigner un périphérique IO, on clique sur l'interface Profinet puis mode de fonctionnement et on sélectionne **Périphérique IO** puis on affecte son contrôleur.

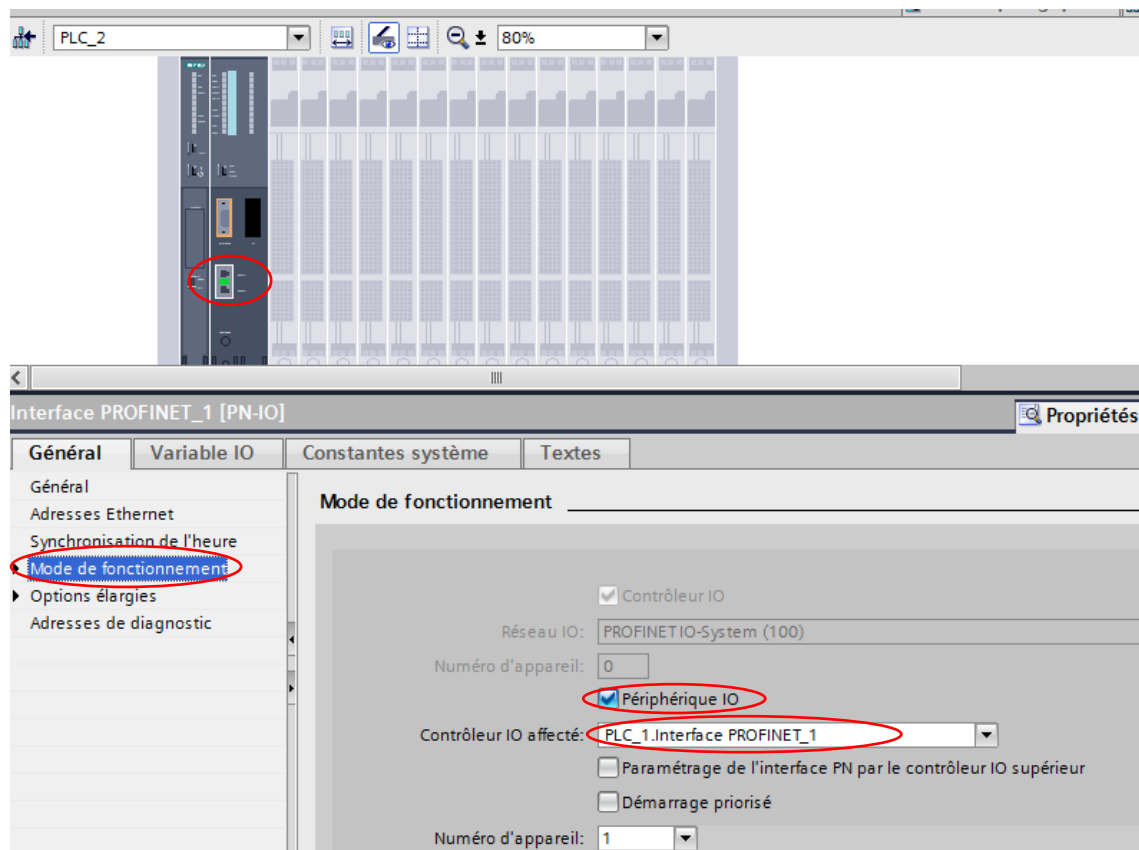


Figure 7_Configuration d'un périphérique IO

Sur la *vue réseau* le réseau IO est identifié par le nom du contrôleur (PLC1) (fig 8).

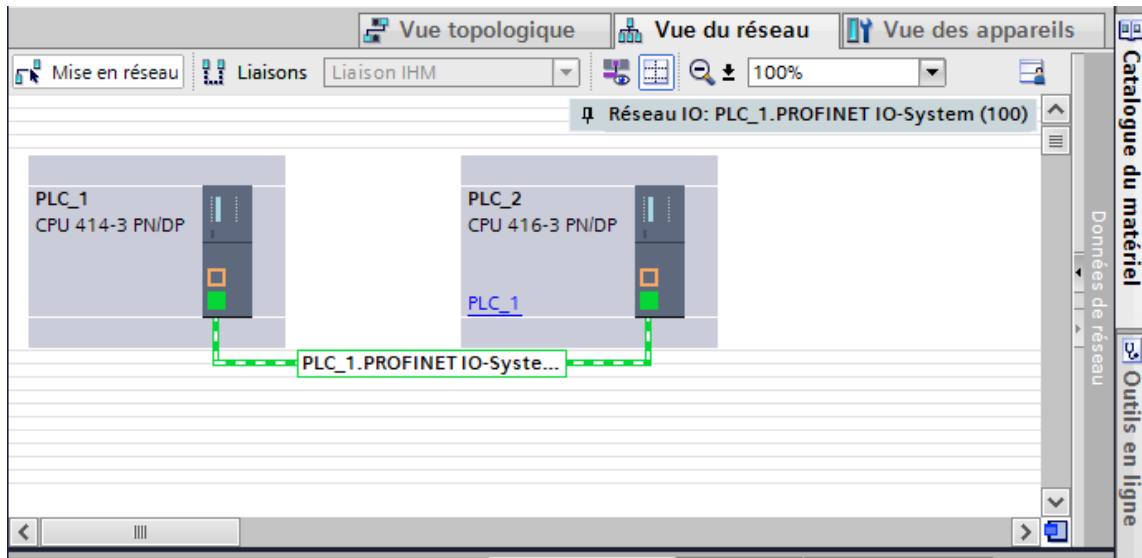


Figure 8_vu du réseau du Réseau IO

On veut envoyer la sortie du contrôleur **Q0.0 (PLC1)** vers l'entrée du périphérique **I0.0(PLC2)** (fig.9)

The screenshot shows the 'Interface PROFINET_1 [PN-IO]' configuration window. The 'Général' tab is selected, and the 'Communication périphérique 1' section is active. The 'Zones de transfert' table is displayed, showing a transfer zone for Q0 to I0 with a length of 1 octet.

...	Zone de transfert	Type	Adresse dans le contrôleur IO	↔ Adresse dans le pér..	Longueur
1	Zone de transfert_1	CD	Q 0	→ I 0	1 octets
2	<Ajouter nouveau>				

Figure 9_zone de transfert Profinet IO