#### MODULE : AUTOMATES RÉSEAUX SPÉCIALITÉ : GÉNIE INDUSTRIEL ET MAINTENANCE / S6

## 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 :
---	------------------	-----------------	----------	-------------

-	Processeur => Intel® Celeron® Dual Core 2,2 GHz (Ivy/Sandy Bridge)							
-	$RAM \Rightarrow 2 GB$							
-	Espace disque disponible => 20 GB							
-	Systèmes d'exploitation => Windows 7 (64 bits)							
-	Résolution de l'écran => 1024 x 768							
2 Automate	2 Automates programmables S7-1200_ CPU 1215C DC/DC/DC							
2 câbles ré	seau Ethernet avec connecteurs rj45 croisés							
En	cas de Profibus DP : 2 modules de communication SIMATIC S7-1200							
1 c	âble 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<sup>er</sup> CPU en tant que maître et le second en esclave (figure 1).



Figure 1\_création d'un réseau Profibus

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

VA	🕼 Siemens - Projet11 _ 🗆 X								
Pr	ojet Edition Affichage Insertion	En ligne Outils Accessoires Fenêtre Aide	Totally Integrated Automatio	n					
	🕴 🎦 Enregistrer le projet 🛛 🔒	🗎 🗎 🗙 🏷 ± (4 ± 🌆 🖥 🖳 🌆 🖳 🕼 🖉 🎝 Liaison en ligne 📩	POR	TAL					
	Navigateur du projet 🛛 🔳 🖣	14-3 PN/DP] > Blocs de programme > Main [OB1] 🛛 🗕 🖬 🗮 🗙	Instructions 📑 🛽						
	Appareils		Options						
		a a 🔹 🛸 🖦 🚍 🚍 💬 🕾 ± 🖂 🔝 😢 ' 🖂	·	E Ins					
P		Interface de bloc	> Favoris	- E					
le le	▼ 🛅 PLC_1 [CPU 414-3 PN 📃 🔨		<ul> <li>Instructions de base</li> </ul>						
lati	Configuration des a	⊣⊢⊣/⊢⊸─  □	Nom Descriptio	n 5					
	足 En ligne & Diagnostic	▼ Titre du bloc "Main Program Sweep (Cycle)"	Général	^					
gra	<ul> <li>Blocs de programme</li> </ul>	Commentaire	Opérations logiques sur bits	E 2					
Pr	Ajouter nouve	Résourt :	Temporisations	est					
		Neseau I	Compteurs	♥ ₽					
	Blocs système	Réseau 2 :	< III	>					
	Objets technologig	▼ Réseau 3 :	Instructions avancées						
	Sources externes	Commentaire	> Technologie	ac					
	🕨 🔁 Variables API		Communication	les					
	Types de données API		Nom Des	scr					
	Tables de visualis		Communication S7	^ L					
	Sauvegardes en li		= GET Lin	e d 🔳 🛱					
	Image: Second		= PUT Eci	rire j					
	✓ Vue détaillée		USEND En	vo: hé					
	· · · · · · · · · · · · · · · · · · ·		- URCV Red	e le					
	Nom Adresse		BSEND En	vo: vo					
	Non-		BRCV Red	e					
			CONTROL Int	err					
			PRINT En	vo:					
			STARI EXE						
			PECUNE AT	ett					
				er					
		< III > 100%	Communication Open User						
	<	🔍 Propriétés 🚽 🚺 Info 👔 📱 Diagnostic 🚽 🖃 🔺	< III	>					
	✓ Vue du portail	sem 🚁 Main 🚓 Appareils &	Projet Projet11 ouvert.						

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).

Paramètre	Déclaration	Type de données	Zone de mémoire	Description	
REQ	Input	nput BOOL I, Q, M, D, L ou constante Paramètre de commande request ; s			
ID	Input WORD I, Q, M, D, L ou constante liaison avec la CPU partenaire.				
NDR	Output	BOOL	I, Q, M, D, L	<ul> <li>Paramètre d'état NDR :</li> <li>0: La tâche n'a pas encore été démarrée ou elle est encore active.</li> <li>1: La tâche a été exécutée sans erreur.</li> </ul>	
ERROR	ERROR Output BOOL		I, Q, M, D, L	<ul> <li>Paramètres d'état ERROR et STATUS, signalisation d'erreur :</li> <li>ERROR=0 <ul> <li>STATUS a la valeur :</li> <li>0000H : ni avertissement ni erreur</li> </ul> </li> </ul>	

#### 2.1. Paramètres

STATUS	Output	WORD	I, Q, M, D, L	<ul> <li>&lt;&gt; 0000H : avertissement ; STATUS fournit des renseignements détaillés.</li> <li>ERROR=1</li> <li>Une erreur s'est produite. STATUS fournit des renseignements détaillés sur le type d'erreur.</li> </ul>
ADDR_1	InOut	REMOTE		Pointeur désignant les zones à lire dans la
ADDR_2	InOut	REMOTE		or o partenaire
ADDR_3	InOut	REMOTE	1, Q, M, D	Quand le pointeur REMOTE accède à un DB, il faut toujours spécifier ce DB.
ADDR_4	InOut	REMOTE		Exemple : P#DB10.DBX5.0 octet 10.
RD_1	InOut	VARIANT		
RD_2	InOut	VARIANT	I, Q, M, D,	Pointeur désignant les zones de la propre
RD_3	InOut	VARIANT	] L	CPU où stocker les données lues.
RD_4	InOut	VARIANT	1	

		100%	
GET_SFB [SFB14]		🖳 Propriétés 🚺 Info 🚺 📱 Diagnostic	
Général Configuration			
Paramètres 📀 Paramètres de l	a liaison		<u> </u>
Paramètres 🤡 Général			aches
	Local	Partenaire	
Nœud d'extré	n.: PLC_1	PLC_2 [CPU 1516-3 PN/DP]	
	<b>—</b>		Sibliotheque
Interfa	ce : PLC_1, Interface DP_1[X3]	PLC_2, Interface DP_1[X3]	S
Sous-rése	au : PROFIBUS	PROFIBUS	
Nom sous-rése	de PROFIBUS_1	PROFIBUS_1	
ID de liaison (d	ćc.) : 100		
Nom de liaison	: S7_Liaison_1		
	🖬 Etablissement actif de la liaison		
<	A sens unique		~
🚓 Appareils & 🔁 Main		✓ Projet Projet16_GET_PUT créé.	

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'acquittement 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.

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
				Paramètre d'état DONE :
DONE	Output	BOOL	I, Q, M, D, L	<ul> <li>0: La tâche n'a pas encore été démarrée ou elle est encore active.</li> <li>1: La tâche a été exécutée sans erreur.</li> </ul>
				Paramètres d'état ERROR et STATUS,
ERROR	Output	BOOL	L, Q, WI, D,	ERROR=0     STATUS a la valour :
STATUS	Output	WORD I, Q, M, D, L		0000H : ni avertissement ni erreur

#### 3.1. <u>Paramètres</u>

				• ERROR=1 Une erreur s'est produite. STATUS fournit des renseignements détaillés sur le type d'erreur.
<b>S7-300:</b> ADDR_1			S7-300 : M, D	Pointeur désignant les zones où écrire dans la CPU partenaire <b>Remarque :</b> Quand le pointeur ANY accède à un DB, il faut toujours spécifier ce DB (par
<b>S7-400:</b> ADDR_i (1≤ i ≤4)	InOut	ANY	S7-400 : I, Q, M, T, C, D	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 contrôleurpériphérique 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.





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

F Vue t	topologique 🛛 🛔 Vue du rése	au 🛿 Vue des appareils	
Mise en réseau Liaison IHM	🔽 🗮 🔜 🔍 ± 100	0% 🔽 📑	Cata
	4 Réseau IO: PLC_1.I	PROFINET IO-System (100)	logu
			le di
PLC_1 CPU 414-3 PN/DP	PLC_2 CPU 416-3 PN/DP	, and the second s	ma
			térie
	PLC_1	de la cela	1
	IQ-Syste		2
		Ľ	Out
			ls e
			nlig
< IIII		> 📃	ne

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)

1	Interface PROFINET_1 [PN-IO]						Rropriétés	1	Info 🚺 🎚 Diagn	ostic	
ſ	Général	Variable IO		Cons	tantes système	Textes					
Г	Général				Numéro d'appare	il: 1	<b>T</b>				*
	Adresses Eth	ernet									
	Synchronisati	ion de l'h	<b>C</b> -								
P	Mode de fonc	tionnement	Co	mmur	nication peripherique	·					
Options élargies     Zones de transfert											
	Adresses de diagnostic										
					Zone de transfert	Туре	Adresse dans le contrôleur IO	↔	Adresse dans le pér	Longueur	
				1	Zone de transfert_1	CD	Q 0	+	10	1 octets	
		<b>^</b>		2	<ajouter nouveau=""></ajouter>						

Figure 9\_zone de transfert Profinet IO