



TUTORIEL Modbus IP



Retrouvez-nous sur www.pl-systems.fr



Sommaire

Introduction.....	3
Configuration de l'adresse IP	4
Programmation du modbus IP Maitre	5
Programmation du modbus IP Esclave.....	9



INTRODUCTION

Produits concernés

USP-070-XX/USP-104-B10/USP-156-B10/USX-XX-XX /USC-XX-XX

Le **Modbus IP** est un protocole de communication qui permet à plusieurs équipements de communiquer via le réseau Ethernet.

On va retrouver un maître qui exécute les requêtes Modbus et un ou plusieurs esclaves qui les reçoit.

Dans notre exemple, le maître aura pour adresse IP : 192.168.1.10 et l'esclave sera en : 192.168.1.20

Exemple de configuration :



USP-104-B10 (Maitre)



USC (Esclave)

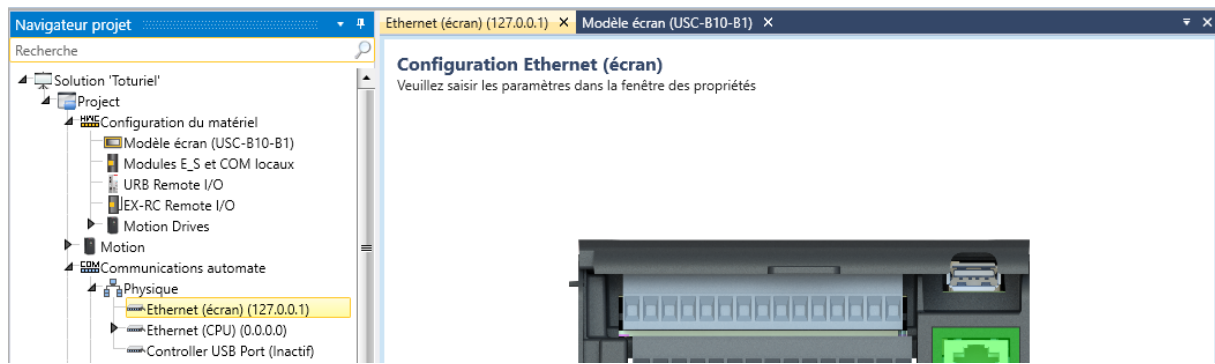
Retrouvez-nous sur www.pl-systems.fr

CONFIGURATION DE L'ADRESSE IP

On va commencer par configurer l'adresse IP du maître.

Un UniStream dispose de 2 adresses IP : « **IP écran** » et « **IP CPU** ». Le protocole Modbus IP utilise l' « **IP écran** », nous allons donc configurer celle-ci.

Dans la partie « **Navigateur projet** », aller dans « **Ethernet (écran)** » :



Project > Communications Automate > Physique > Ethernet (écran) (127.0.0.1)

Puis dans la fenêtre « **Propriétés** », renseigner les paramètres IP comme ci-dessous :



Faire la même chose pour l'esclave avec ces paramètres :

Adresse IP : 192.168.1.20

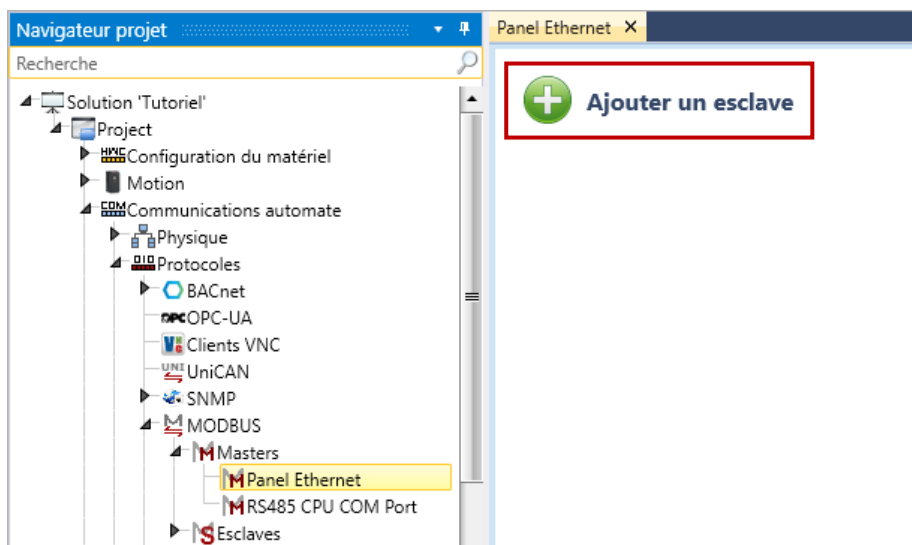
Masque sous-réseau : 255.255.255.0

Passerelle par défaut : 192.168.1.1

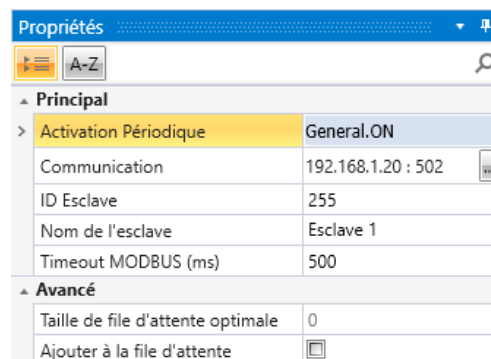


PROGRAMMATION DU MODBUS IP MAITRE

Pour créer les requêtes Modbus à exécuter, il faut aller dans la partie « **Navigateur projet** », « Communication Automate > Protocoles > MODBUS > Masters > Panel Ethernet », et cliquer sur « **Ajouter un esclave** ».



Une fois l'esclave créé, renseigner les propriétés de ce dernier dans la fenêtre des « Propriétés » à droite comme ci-dessous :



A présent l'esclave est configuré, il faut lui associer des requêtes Modbus. Il y a différents types de commandes Modbus.

Retrouvez-nous sur www.pl-systems.fr



Tout d'abord il existe 4 types d'exécution des requêtes :

- **Périodique** : Les commandes de lecture ou d'écriture s'exécutent automatiquement selon une fréquence (exemple : à chaque 500ms, on effectue la requête)
- **Apériodique** : Les commandes de lecture ou d'écriture s'exécutent depuis la programmation ladder.
- **Multiple Périodique** : La commande combine la lecture et l'écriture et s'exécutent automatiquement selon une fréquence.
- **Multiple Apériodique** : La commande combine la lecture et l'écriture et s'exécutent depuis la programmation ladder.

Ensuite il y a les types de commande en fonction des données à lire/écrire :

- **Read Coils (1)** : Lecture d'une variable de type « Bit »
- **Read Discrete Inputs (2)** : Lecture d'une entrée TOR
- **Write Single Coil (5)** : Ecriture d'une variable de type « Bit »
- **Write Multiple Coils (15)** : Ecriture de plusieurs variables de type « Bit »
- **Read Holding Registers (3)** : Lecture d'une variable de type « INTEGER »
- **Read Input Registers (4)** : Lecture d'une entrée analogique
- **Write Single Register (6)** : Ecriture d'une variable de type « INTEGER »
- **Write Multiple Registers (16)** : Ecriture de plusieurs variables de type « INTEGER »


A présent nous allons créer 2 requêtes de manière périodique et apériodique :

- Une pour lire une variable de type « INTEGER »
- Une pour écrire une variable de type « BIT ».



1- Les requêtes périodiques

Dans l'onglet « **Bits (Périodique)** », cliquer sur « **Ajouter nouvelle opération** » :


Ajouter nouvelle opération

Bits (Périodique)		Registres (Périodique)		Bits (Apériodique)		Registres (Apériodique)		Write/Read Multiple Registers Periodic		Write/Read Multiple Registers Aperiodic	
Name	Tag	Adresse	Action	Opération	Fréquence Envoi	Unité	Activer	Status			
Coil1	Bit écrit vers esclave	0	Write	Write Multiple Coils (15)	100	Milliseconds	<Empty>	<Empty>			

On va écrire dans notre esclave un BIT à l'adresse 0 toute les 100ms. La valeur écrite sera celle de la variable « Bit écrit vers esclave ».

Dans l'onglet « **Registres (Périodique)** », cliquer sur « **Ajouter nouvelle opération** » pour créer une commande Modbus :


Ajouter nouvelle opération


Bits (Périodique)		Registres (Périodique)		Bits (Apériodique)		Registres (Apériodique)		Write/Read Multiple Registers Periodic		Write/Read Multiple Registers Aperiodic	
Name	Tag	Adresse	Action	Opération	Fréquence Envoi	Unité	Activer	Status			
Register1	Mot lu depuis esclave	0	Read	Read Holding Registers (3)	100	Milliseconds	<Empty>	<Empty>			

On va lire dans notre esclave un INTEGER à l'adresse 0 toute les 100ms. La valeur lue sera stockée dans notre variable « Mot lu depuis esclave ».



2- Les requêtes apériodiques


Dans l'onglet « **Registres (Apériodique)** », cliquer sur « **Ajouter nouvelle opération** » pour créer une commande Modbus :


Ajouter nouvelle opération

Bits (Périodique)	Registres (Périodique)	Bits (Apériodique)	Registres (Apériodique)	Write/Read Multiple Registers Periodic	Write/Read Multiple Registers Aperiodic		
Name	ID	Group ID	Tag	Adresse	Action	Opération	Status
Register1	0	Sans	Mot lu depuis esclave	0	Read	Read Holding Registers (3)	<Empty>

Lorsque la requête sera exécutée, on va lire dans notre esclave un INTEGER à l'adresse 0. La valeur lue sera stockée dans notre variable « Mot lu depuis esclave ».

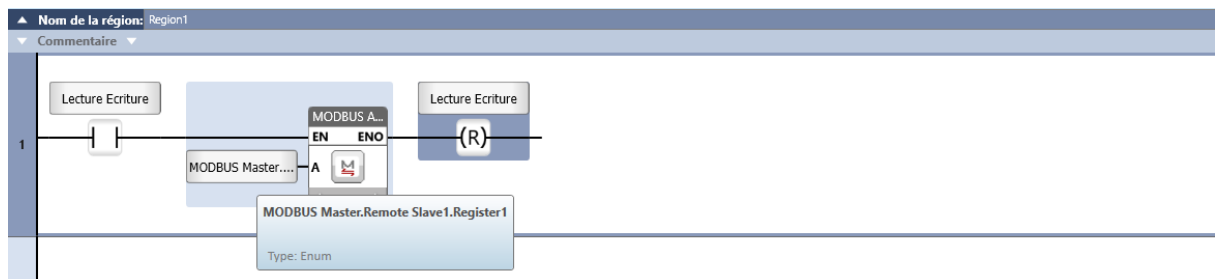
Dans l'onglet « **Bits (Apériodique)** », cliquer sur « **Ajouter nouvelle opération** » :


Ajouter nouvelle opération

Bits (Périodique)	Registres (Périodique)	Bits (Apériodique)	Registres (Apériodique)	Write/Read Multiple Registers Periodic	Write/Read Multiple Registers Aperiodic		
Name	ID	Group ID	Tag	Adresse	Action	Opération	Status
Coil1	1	Sans	Bit écrit vers esclave	0	Write	Write Multiple Coils (15)	<Empty>

Lorsque la requête sera exécutée, on va écrire dans notre esclave un BIT à l'adresse 0. La valeur écrite sera celle de la variable « Bit écrit vers esclave ».

Et pour exécuter la requête « Register1 » ou « Coil1 », il faut le programmer dans le ladder en utilisant le bloc « **Modbus Aperiodic Trigger** » comme ci-dessous :



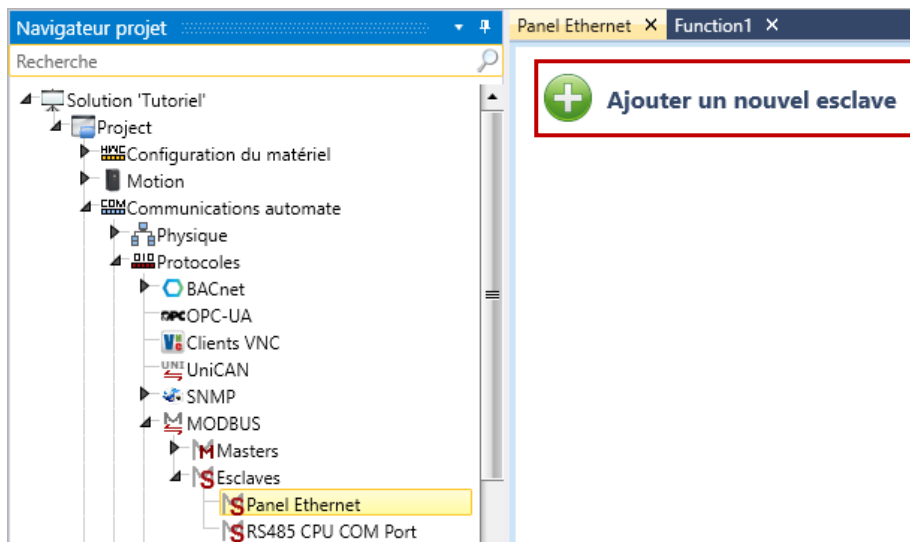
PROGRAMMATION DU MODBUS IP ESCLAVE

Lorsque l'UniStream est esclave Modbus, il faut configurer les variables à échanger avec le maître. Ainsi le maître pourra effectuer ses requêtes si :

- le type de la requête est correcte
- le type de la variable correspond
- l'adresse modbus correspond
- la permission le permet

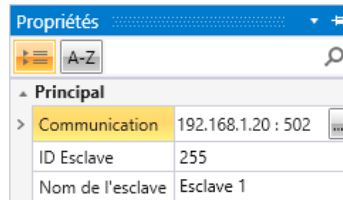
Pour la partie maître, on a créé 2 requêtes Modbus (lecture d'un INTEGER et écriture d'un BIT), on va donc configurer la communication côté esclave pour que le maître puisse effectuer ces 2 requêtes.

Du coup dans la partie « **Navigateur projet** », on va aller dans la partie esclave de la communication Modbus : « Communication automate > Protocoles > MODBUS > Esclaves > Panel Ethernet », et cliquer sur « **Ajouter un esclave** ».





Une fois l'esclave ajouté, renseigner les propriétés dans la fenêtre des « Propriétés » à droite comme ci-dessous :



A présent, dans l'onglet « **Registres** », cliquer sur « **Ajouter nouvelle opération** » et renseigner cette dernière comme ci-dessous :



Bits		Registres			
Name	Tag	Adresse	Permissions		
Register1	Mot lu depuis maître	0	Lecture seule		

Le maître pourra lire un INTEGER à l'adresse 0.

Dans l'onglet « **Bits** », cliquer sur « **Ajouter nouvelle opération** » et la renseigner comme ci-dessous :



Bits		Registres			
Name	Tag	Adresse	Permissions		
Coil1	Bit reçu depuis maître	0	Lecture/Ecriture		

Le maître pourra écrire un BIT à l'adresse 0.