

Compte Rendu

TP Interbus



26 février 2018

UCA

Mathurin CARTRON & Mickael DELAHAYE

Contenu

[Préface 0](#_Toc509407025)

[Travaille à effectuer 0](#_Toc509407026)

[Titre Tache 1 0](#_Toc509407027)

[Le code 0](#_Toc509407028)

[La simulation 0](#_Toc509407029)

# Préface

Dans le but de mieux comprend la finalité du cours sur les réseaux industriels. Nous avons trois TP sur trois réseaux différents. Celui-ci traite du réseau *Interbus* Avec un automate programmable de marque *RAVOUX* qui est une trémie.

# Travaille à effectuer

Nous allons donc devoir à l’aide du logiciel *PC Works* programmé la trémie et dialoguer avec. En gérant l’acheminement des pots et leur remplissage. Ensuite ils seront évacués.

## L’adresse IP de ma machine

Comme ce TP étais le même que l’année précédente je me suis permis d’utilisé le compte rendu de l’année dernière pour gagner du temps. Après tout c’est à ça qu’il sert. Je n’ai donc pas cherché comment trouver mon adresse IP mais je me suis contenter de taper la commande suivante dans un  *invite de command*.

P:\>ipconfig /all

Cf : annexe 1 réponse de l’invite de commande.

## Teste de la communication avec l’API

De même pour communiqué avec mon automate en réseau j’ai pris la peine de vérifié l’adresse de l’automate sur le réseau. J’ai effectué un test de Ping en pointant l’adresse de l’automate. Toujours dans un  *invite de commande*.

P:\>ping 172.31.228.10

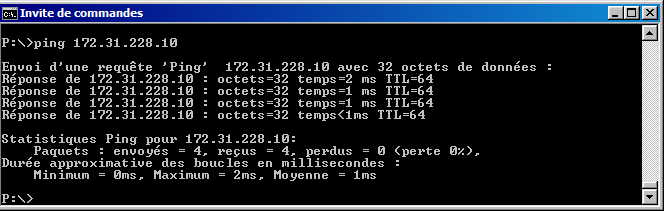


Figure 1 - Réponse de *l'invite de commande*

Ici on voit très bien qu’il y a un dialogue entre les deux machines. Il nous donne même le temps qu’elles mettent à échanger des données (moins 1ms)

Il nous fait même un petit paragraphe de conclusion avec le nombre de paquets envoyer ceux qu’il a perdu dans la discussion est ce qu’i est bien arrivé. On constate que la communication est bonne parce que je n’ai rien perdu et que le Ping est rapide (temps de réponse faible).

Pour voir ce qui se serai passé si ma machine avais été déconnecté ou défaillante j’ai fait une faute de frappe volontaire dans l’adresse de celle-ci. Histoire de voir ce que fais mon ordinateur quand il veut parler a un mur.

P:\>ping 172.31.228.18

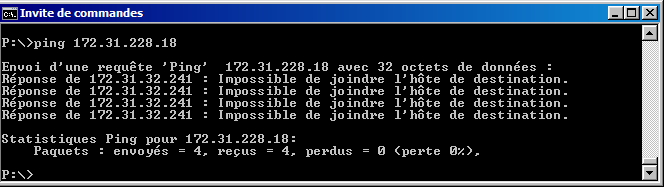


Figure 2 - réponse de *l'invite de commande*

## Développement du projet

J’ai commencé par ouvrir le logiciel pour piloter l’automate : PC WORX. Et j’ai créé mon projet. Mais à peine le contrôleur choisi qu’une fenêtre d’erreur pop déjà. Me disant que la compilation a échoué. Très rassurant. Lorsque je compile/enregistre. Le message re-pop ! Cette fois je les enregistrer. Mais le fait que le logiciel refuse d’obtempérer dès le début ne présage rien de bon pour la suite. Après rectification auprès de l’enseignant le message est en fait normale et du à l’installation logiciel.

Je paramètre les adresse dans le logiciel pour qu’il parle avec la bonne machine et une fois tous mes paramètres entrés, je teste et le logiciel me dit que tout est OK pour lui.

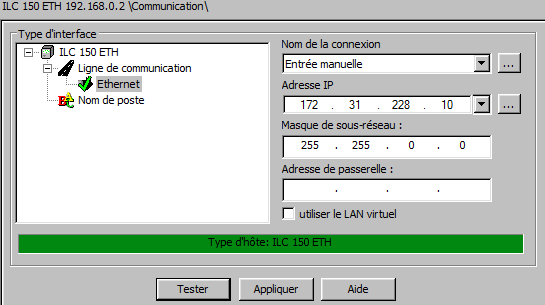


Figure 3 - Réponse du logiciel lors du paramétrage de l'adresse

Il me fait d’ailleurs par du type d’hôte : ILC 150 ETH

### Q.1- Ce réseau TCP/IP est-il de type terrain ou entreprise ?

On constate que nous somme sur un réseau Ethernet comme nous le dis l’abréviation ETH donc dans la pyramide CIM nous somme en haut. Soit un réseau de type entreprise.

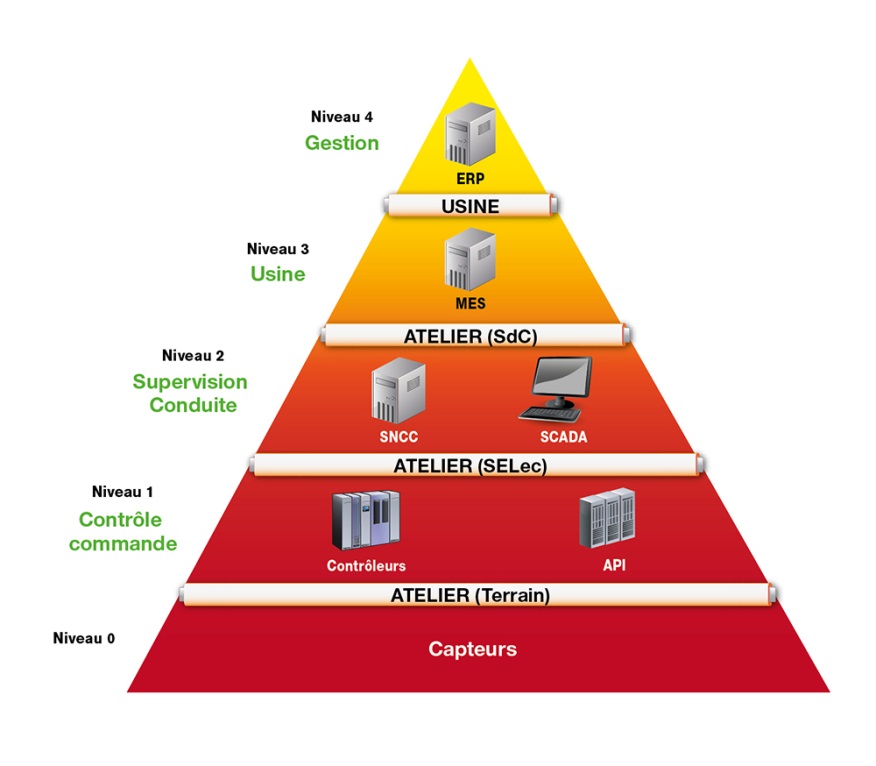


Figure 4 - Pyramide CIM

Je dois maintenant paramétrer ce qu’il y a sur mon réseau. Je demande à mon logiciel de cherche ce qu’il trouve sur le réseau que je viens de paramétrer. Je vais lui demander de me lister tous les INTERBUS connecté. Il va me sortir plein d’information et même si ça a l’aire incompréhensible au premier abord ça n’est pas que du chinois. Tout d’abord il fouille le réseau avec une trame de management (normal je lui ai demandé de le faire) Puis dès qu’il détecte quelque chose il me le liste. Avec son numéro (code d’identifiant) et la taille des infos qu’il peut me transmettre. Et il continue jusqu’à la fin du réseau. Et comme un petit dessin vaut mieux qu’un long discoure :

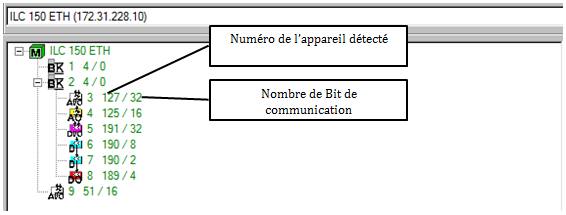


Figure 5 - Paramètre des éléments sur le réseau

Néanmoins qu’est-ce que mon appareil 127 ? Et bien parton du principe que l’ordinateur ne fait que ce que je lui demande. Donc si je ne sais pas il est fort probable que lui aussi. Je vais donc devoir lui dire ce qu’il a trouvé !

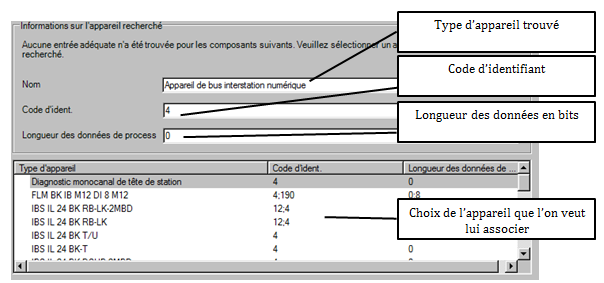


Figure 6 - Paramètre des éléments sur le réseau

Le logiciel à même l’amabilité de nous rappeler la configuration dans le panneau latéral.

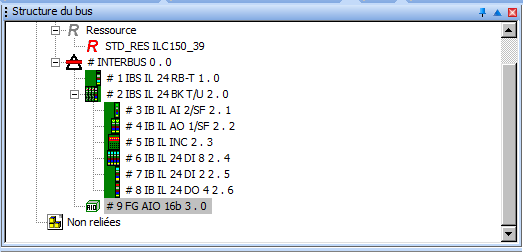


Figure 7 - Paramètre des élément sur le réseau *INTERBUS*

### Q.2- Comment se nomme ce réseau ? RLI ou RLE ?

Ici nous somme plus du côté des capteurs c’est donc un réseau de terrain. Soit un RLI.

### Q.3- Comment l’automate fait-il pour réaliser la détection du réseau ?

On commence par envoyer une trame de management. Tous les esclaves présents sur le réseau vont répondre par trois informations. Le numéro de l’appareil, son type, et le nombre de bit de communication. (Figure 5) le maitre reçoit donc une trame comportant dans l’ordre, toutes les informations nécessaire à la réalisation d’un réseau.

### Q.4- Quelle est la topologie réelle du réseau ?

En réalité ce n’est pas un bus CAN comme on pourrait le croire en voyant le câblage il s’agit d’un bus en anneau.

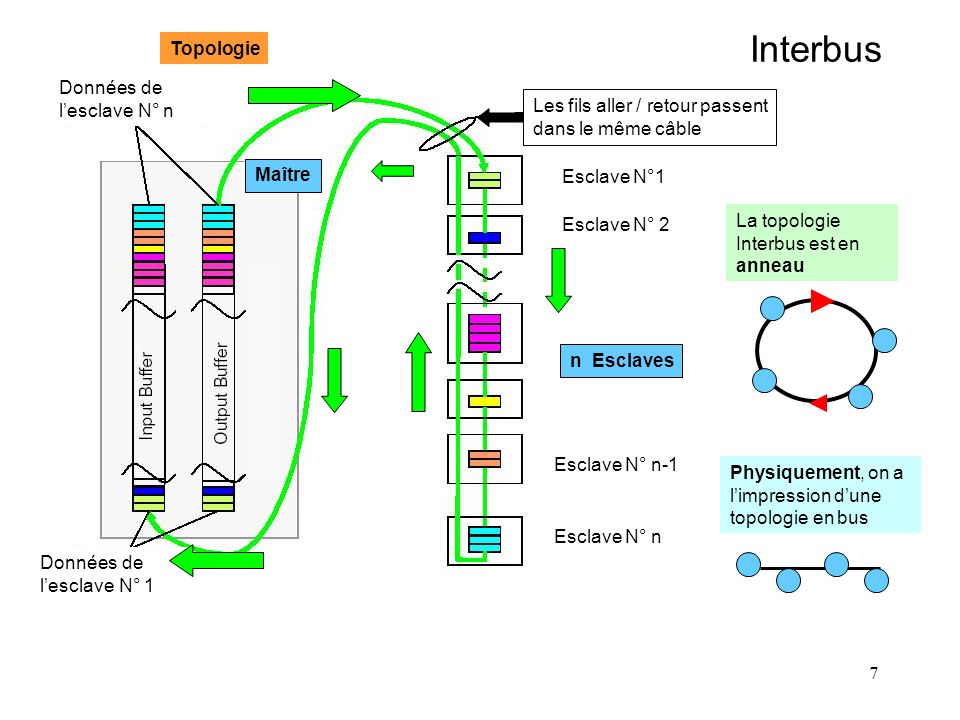


Figure 8 - Topologie d'un réseau *INTERBUS*

On peut aussi le représenter sous forme d’arbre. Se rapprochant plus de ce que l’on voit dans les faite mais il est moins aisé de comprendre le fonctionnement réelle de *l’INTERBUS*.

### Q.5- A quoi sert la tête de station ?

Une tête de station sert à « agrandir » le réseau. Je vais prendre ici un exemple concret. Il est courant qu’un automate ne possède pas qu’un seul esclave pour qu’il soit capable de travailler de manière plus complexe que de faire une seule action. Mais comment contrôler TOUT les composantes de TOUTES les machines ? C’est là que la « tête de station » entre en jeu. Elle permet d’étendre le réseau a un sous réseau que elle gère. Pour comprendre ça j’ai fait un parallèle avec quelque chose que je connaissais déjà : les fonctions en C. Plutôt que de tout coder dans le *Main* on va préférer déléguer à des fonctions. Ici c’est pareil. Plutôt que d’enquiller tous les automates à la suite on va les mettre dans des sous réseau piloté par des têtes de station.

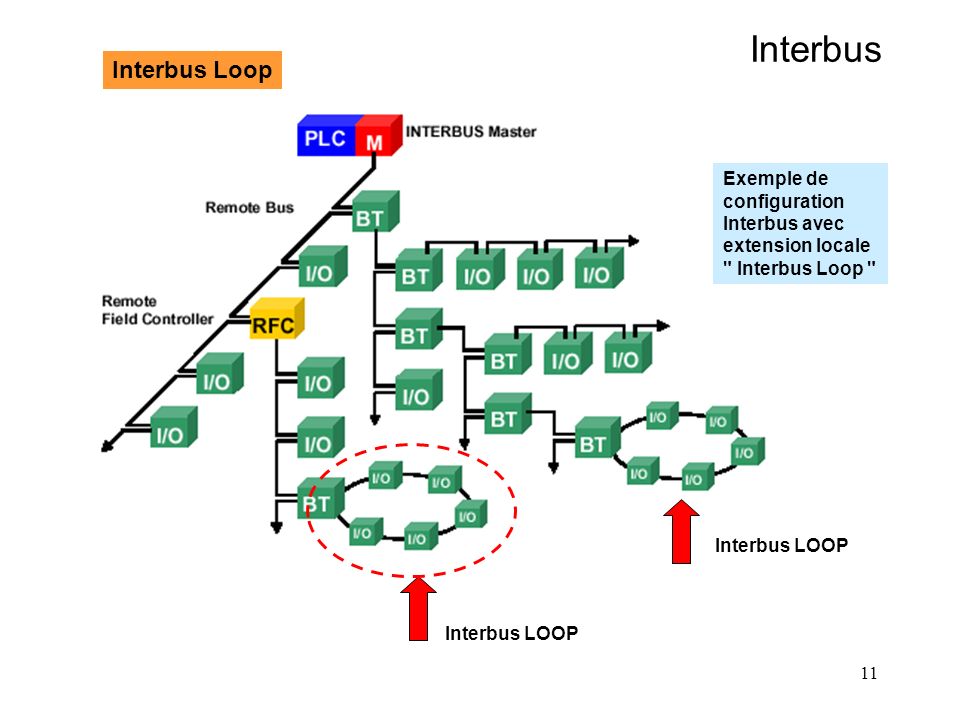


Figure 9 - Schéma du fonctionnement d'une tête de station

Nous avons donc un réseau en anneaux qui est un sous ensemble d’un plus grand réseau en annaux. Cela présente d’énorme avantage car on peut opérer sur un sous réseau sans perturber les autres. Car l’on ouvre l’anneau du sous ensemble et pas celui du réseau principale.

### Q.6- Aurait-on pu faire une configuration hors ligne du réseau ?

Tout à fait ! Comme expliquer plus haut. Il n’est pas rare que une machine soit trop loin, en maintenance ou tout simplement pas encore réalisé lors de la conception du programme qu’elle va habiter. On doit donc être en mesure de pouvoir réaliser un programme sans y être connecté. On va donc chercher dans une bibliothèque quelle esclaves sera ou dans le réseau. Puis on écrit notre programme. Que l’on pourra soumettre à l’automate par la suite. A condition que le réseau créé virtuellement corresponde au réseau réel. Mais il ne s’agit pas de la question ici. Mais d’une bonne gestion du projet en amont.

### Q.7- Observer le contenu de la table de variable, combien en a-t-on ?

Dans le *MainV* on voit que l’on possède plusieurs lignes.

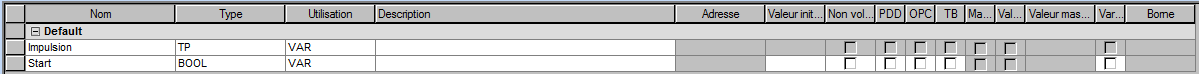


Figure 10 - Table de variable global

On voit que dans la table de variables global l’entrer de notre composant *Impulsion* et désormais déclaré comme une variable. Ce qui nous permettra de l’utiliser dans tout le programme.

### Q.8 Quel est le type de chaque variable ?

La variable Start a pour type *Bool*, il sera donc soit FALSE soit TRUE puis la variable Impulsion a pour type TP. Ce qui signifie *Time Pulse.*

### Q.9 Pour Start, vous avez trois choix, indiquer les différences ?

Nous avons donc trois possibilités

1. VAR : C’est une variable local. Elle ne peut être utilisée que dans le sous bloc fonctionnel  *Impulsion*.
2. VAR\_EXTERNAL : C’est un variable globale. Elle peut donc être modifiée par n’importe quel morceau du programme
3. VAR\_EXTERNAL\_PG : Elle a le même fonctionnement que celle du dessus mais en plus elle peut être reliée à une entrée/sortie physique.

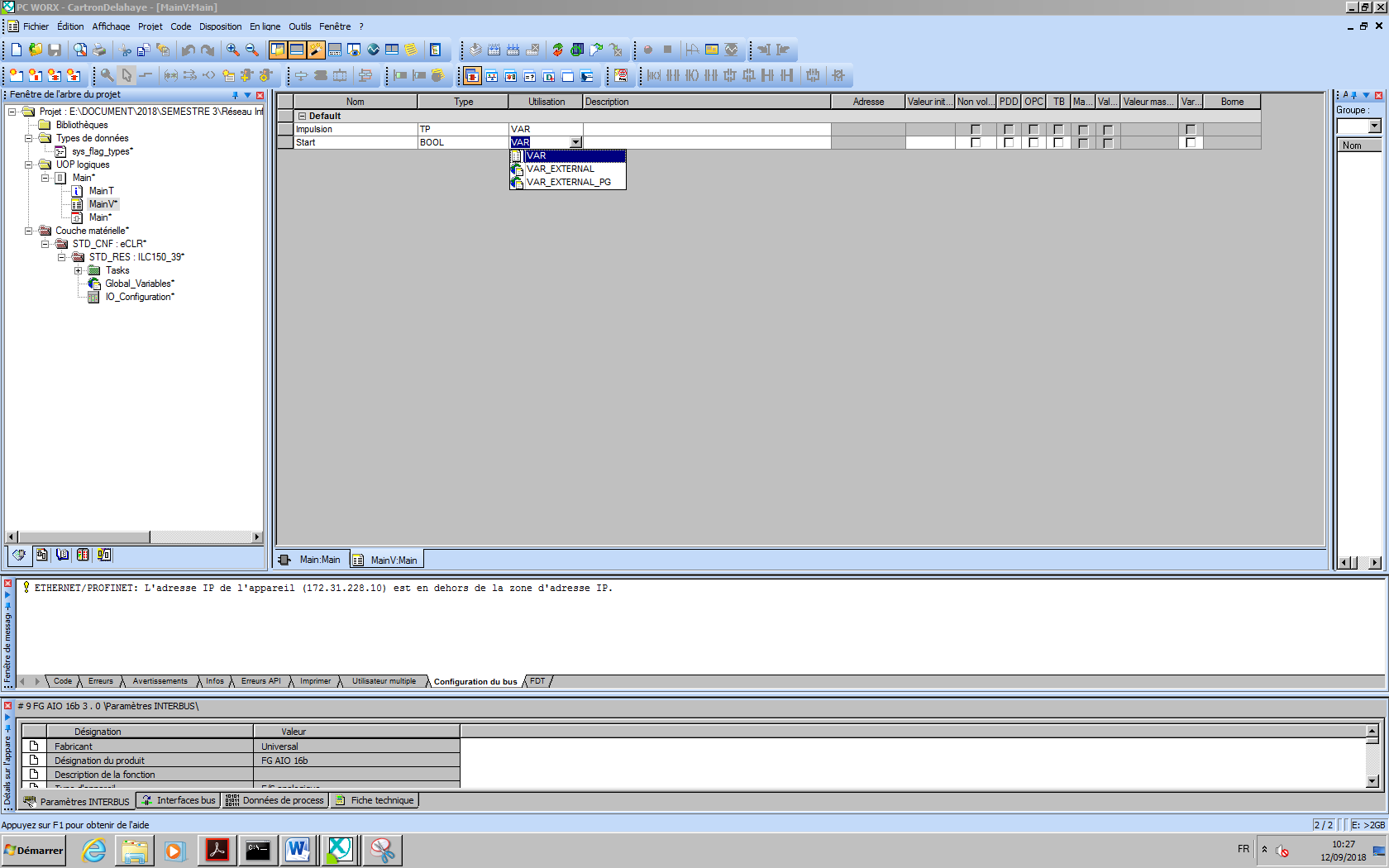


Figure 11 - Attribution du type au variable.



### Le code

Lorem ipsum dolor sit amet,

consectetur adipiscing elit.

In nulla mi, sodales sed eros et,

vehicula feugiat sem. Nulla eu hendrerit augue.

Curabitur orci mauris, convallis

a suscipit eu, vehicula et ante.

Orci varius natoque penatibus et

magnis dis parturient montes,

nascetur ridiculus mus. Sed lao

reet sagittis nibh, mollis portti

tor mauris. Donec lacinia a met

us quis rutrum. Vestibulum luctu

s quam eget ligula laoreet eleife

nd. Donec erat enim, lobortis vel

interdum a, tincidunt ac diam. Nul

la sed arcu ac nibh luctus finibu

s et ac ipsum.

### La simulation



Figure 12 - Simulation du compteur