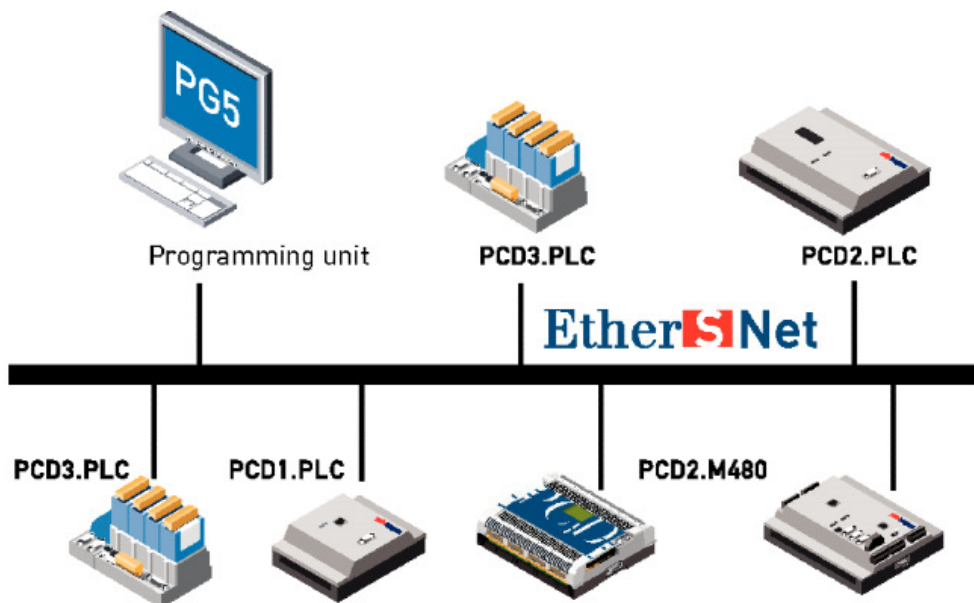


Getting Started: Ether-S-Bus communication



Historique du document

Date	Auteur	Modification
31.11.2009	S.Ki, TCS	- Création du document (Version 1) Exemple pour réaliser une communication Ether-S-Bus entre des PCD's - correction du texte par CD
10.12.2009	S.Ki, TCS	- Restructuration du document (Comment définir un MASTER, Comment définir un Slave)
11.01.2010	S.Ki, TCS	- Rajout du chapitre 6 et 7
18.01.2010	S.Ki, TCS	- Correction pour la publication du document

Contenus

1	Introduction.....	3
2	Condition d'utilisation.....	4
2.1	Matériels requis (Hardware).....	4
2.2	Logiciels requis (Software).....	4
3	Communication entre 2 PCD's.....	5
3.1	Plan de connexion réseau:.....	5
3.1.1	Avec seulement un câble Ethernet croisé	5
3.1.2	Avec un composant réseau tel un HUB ou un SWITCH	6
3.2	Configuration du "S-Bus address", l'"IP address" et l'IP Node	7
3.2.1	Sur la STATION 1: (dans cet exemple, un PCD3.M5340)	7
3.2.2	Sur la STATION 2: (dans cet exemple, un PCD3.M5540)	9
3.3	TCP/IP Setting Table	10
3.4	Comment définir une station SLAVE.....	12
3.5	Comment définir une station MASTER	12
4	Exemple	13
4.1	Programme d'exemple 1: MASTER to SLAVE communication.....	13
4.1.1	Diagramme d'échange de données (Master to Slave)	13
4.1.2	Configurez le SLAVE.....	14
4.1.3	Programmez le MASTER.....	14
4.1.4	Testez la communication MASTER to SLAVE	15
4.2	Programme d'exemple 2: MASTER to MASTER communication	16
4.2.1	Diagramme d'échange de données (Master to Master)	16
4.2.2	Configurez le MASTER 2	17
4.2.3	Configure the MASTER 1	18
4.2.4	Testez la communication MASTER to MASTER.....	18
5	Communication entre plus de 2 PCD's.....	19
5.1	Plan réseau	19
5.2	Programmation avec des stations additionnelles	19
6	Possibilités d'amélioration des performances	21
6.1	Échanger le plus grand nombre possible de données par FBox "SEND" ou "RCV"	21
6.2	FBox: SASI-Station IP	22
7	Appendix	24
7.1	Quelle est la différence entre un HUB et un SWITCH ?.....	24
7.2	Quel "IP Address" et "Masque (IP) de sous-réseau" sont à utiliser ???.....	24
7.3	Debugging: Signification de la couleur de la LED des Fbox.....	25
8	Contactez nous.....	27

1 Introduction

Ce document explique, comment établir une communication Ether-S-Bus entre 2 PCD's, 3 PCD's et plus encore... Ce document "Getting Started" est accompagné par des exemples que vous pouvez utiliser.

Tous les exemples ou explications de ce document seront programmés ou faites à l'aide de **PG5 2.0**. (Il est clair qu'il n'y a pas de problème pour programmer une communication Ether-S-Bus à l'aide de PG5 1.4.) La programmation se fait de la même manière, que ce soit avec PG5 2.0 ou PG5 1.4.

Il existe une seule différence importante, il s'agit de la configuration du **SBus station address** (= *adresse de Station Sbus*) et de l'**IP address** de vos PCD's.

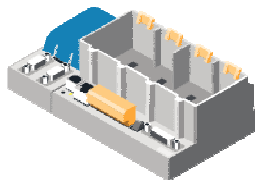
Pour les configurer:

- sur PG5 2.0, le **Device configurator** est utilisé (voir chapitre 3.2)
- sur PG5 1.4, la « fenêtre-outils » **Hardware Setting** est utilisé

2 Condition d'utilisation

2.1 Matériels requis (Hardware)

- 2 x **PCD3.M5540** optionnellement équipé d'un PCD3.S100.



PCD3.Mxxxx



PCD3.S100

(demo module interface):

Les exemples sont réalisés avec deux PCD3.M5540. Si ce matériel n'est pas à votre disposition, les mêmes fonctionnalités peuvent être réalisées avec :

- PCD3.Mxxxx (qui contient une interface/connecteur Ethernet)
- PCD2.M5xxx
- PCD2.M480 + PCD7.F65x (interface Ethernet additionnelle)
- PCD2.M170 + PCD7.F65x (interface Ethernet additionnelle)
- PCD2.M150 + PCD7.F65x (interface Ethernet additionnelle)
- PCD1.M13x + PCD7.F65x (interface Ethernet additionnelle)

- **soit** un câble Ethernet croisé
(seulement si le réseau est composé de 2 stations)
- **ou** si votre réseau contient plus de 2 stations,
plusieurs câbles Ethernet direct (CAT5) + un composant réseaux tel un
un HUB ou un SWITCH

- Un PC avec cable USB et/ou un câble PGU (PCD8.K111) pour programmer vos PCD's.

2.2 Logiciels requis (Software)

Dans ce chapitre aucun outils logiciel ou autres choses ne sont demandés pour programmer les différentes stations d'un réseau Ether-S-Bus, mise à part:

- **PG5 2.0.110**

- La **dernière version firmware** des PCD's correspondant
The **latest firmware version** of the corresponding PCD's is recommended

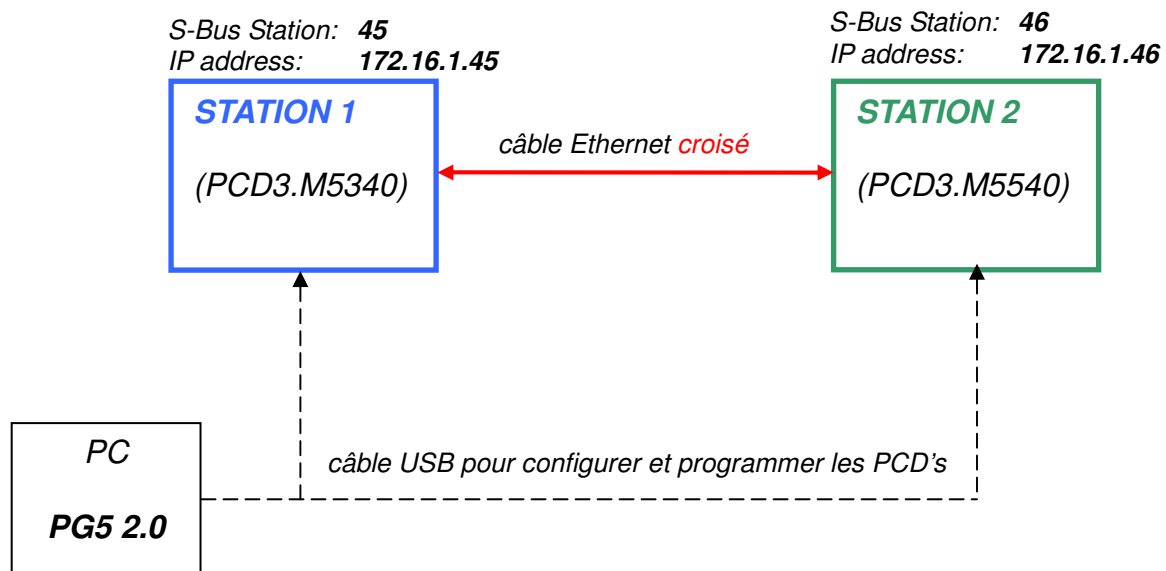
Ces éléments software et firmware peuvent être trouvés sur notre site support internet (www.sbc-support.com)

3 Communication entre 2 PCD's

3.1 Plan de connection réseau:

Il existe 2 façons de connecter 2 PCD's (ou stations) ensemble. Dans les "plan réseaux" ci-dessous, vous pouvez voir comment vous pourriez les réaliser.

3.1.1 Avec seulement un câble Ethernet croisé

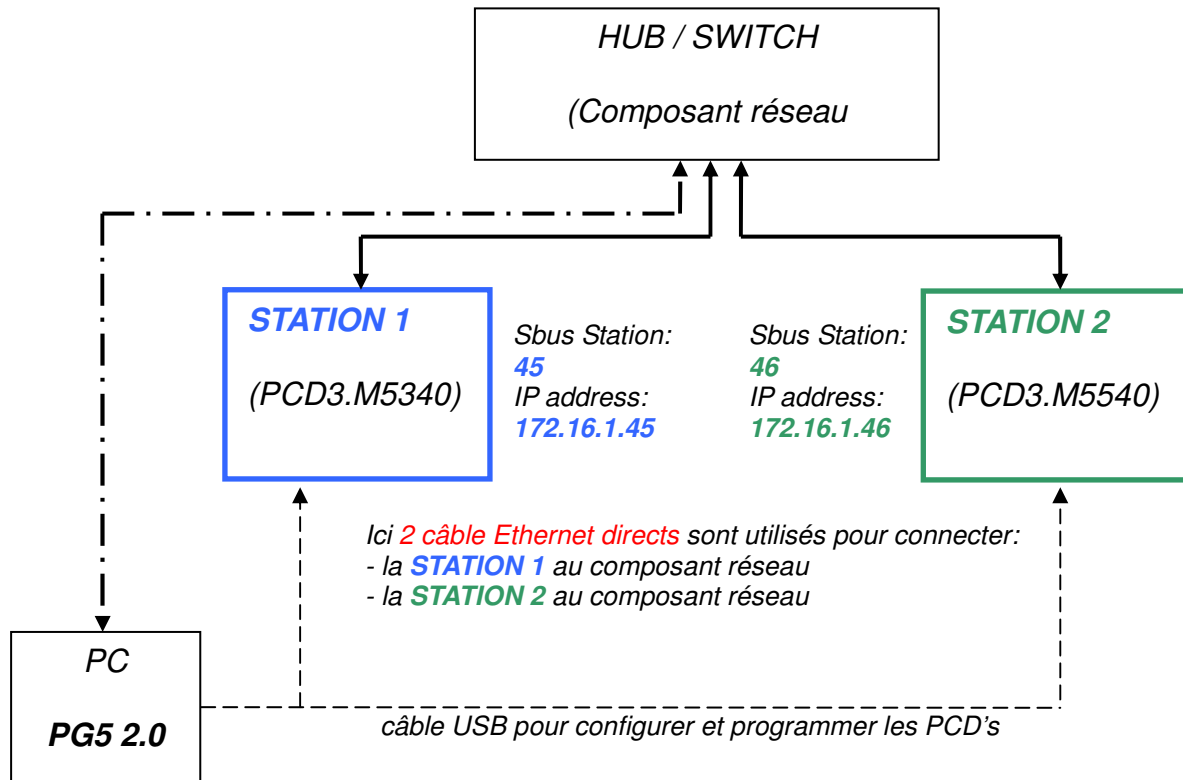


Dans le plan réseau, ci-dessus, il vous suffit d'utiliser un **câble Ethernet croisé** pour connecte les 2 PCD's ensemble.



La connexion entre deux appareils disposant d'une interface Ethernet (PC, PCD, ...), sans utiliser un HUB ou SWITCH, doit être réalisé à l'aide d'un câble Ethernet croisé

3.1.2 Avec un composant réseau tel un HUB ou un SWITCH



Si un composant réseau tel un HUB ou un SWITCH (voir chapitre: 7.1 Quelle est la différence entre un HUB et un SWITCH) se trouve à votre disposition, vous pouvez utiliser directement 2 câbles Ethernet direct pour connecter les 2 PCD's au composant réseau.



Un 3^{ème} câble Ethernet direct pourrait être utilisé pour connecter le PC au composant réseau. De cette manière vous pourriez programmer les PCD's par Ethernet et être en ligne avec le deux PCD's en même temps

Il y va de soi que ceci ne marche que si les PCD's contiennent déjà une « SBus address », une « IP address » et le **même** « masque de sous-réseau » (voir chapitre. 7.2).

3.2 Configuration du “S-Bus address”, l’“IP address” et l’IP Node

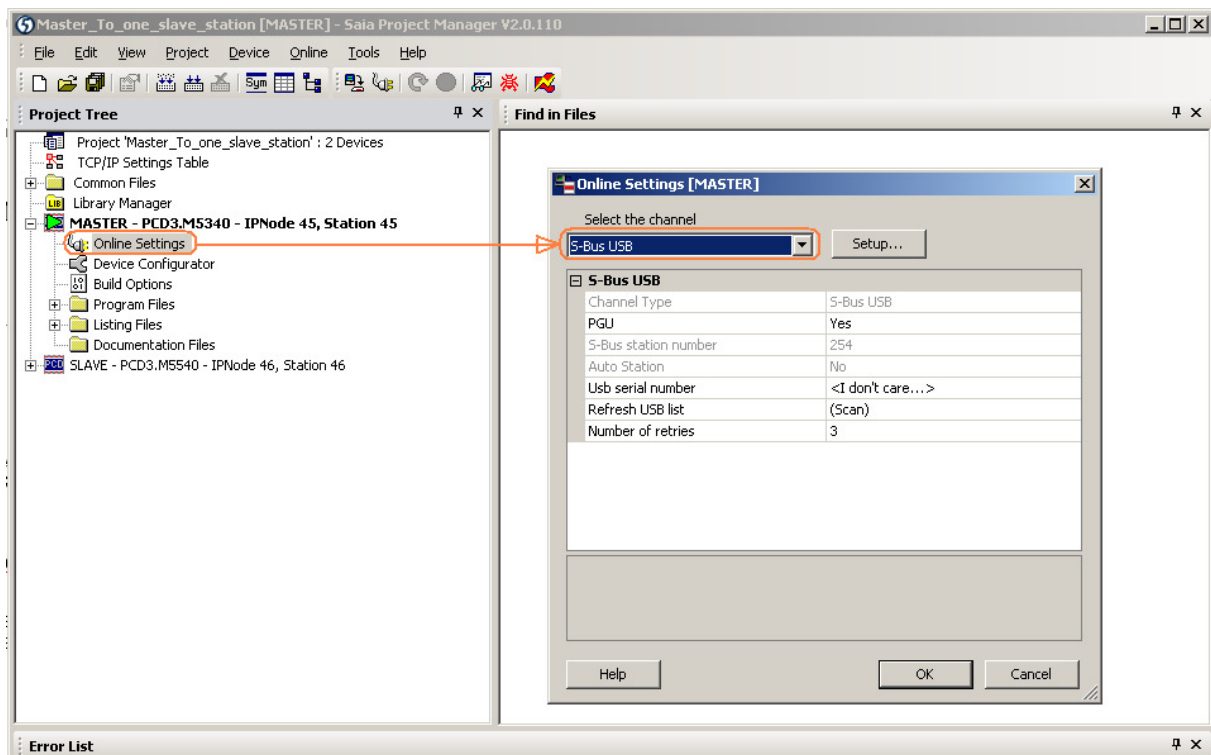
Comme vous pouvez le voir dans les deux plans réseau (chapitre : 3.1.1 et 3.1.2), la STATION 1 et 2 ont besoin d’une **S-Bus Address** (= S-Bus Station number) et d’une **IP address** (référencé à l’**IP Node**, décrit plus tard dans ce document)



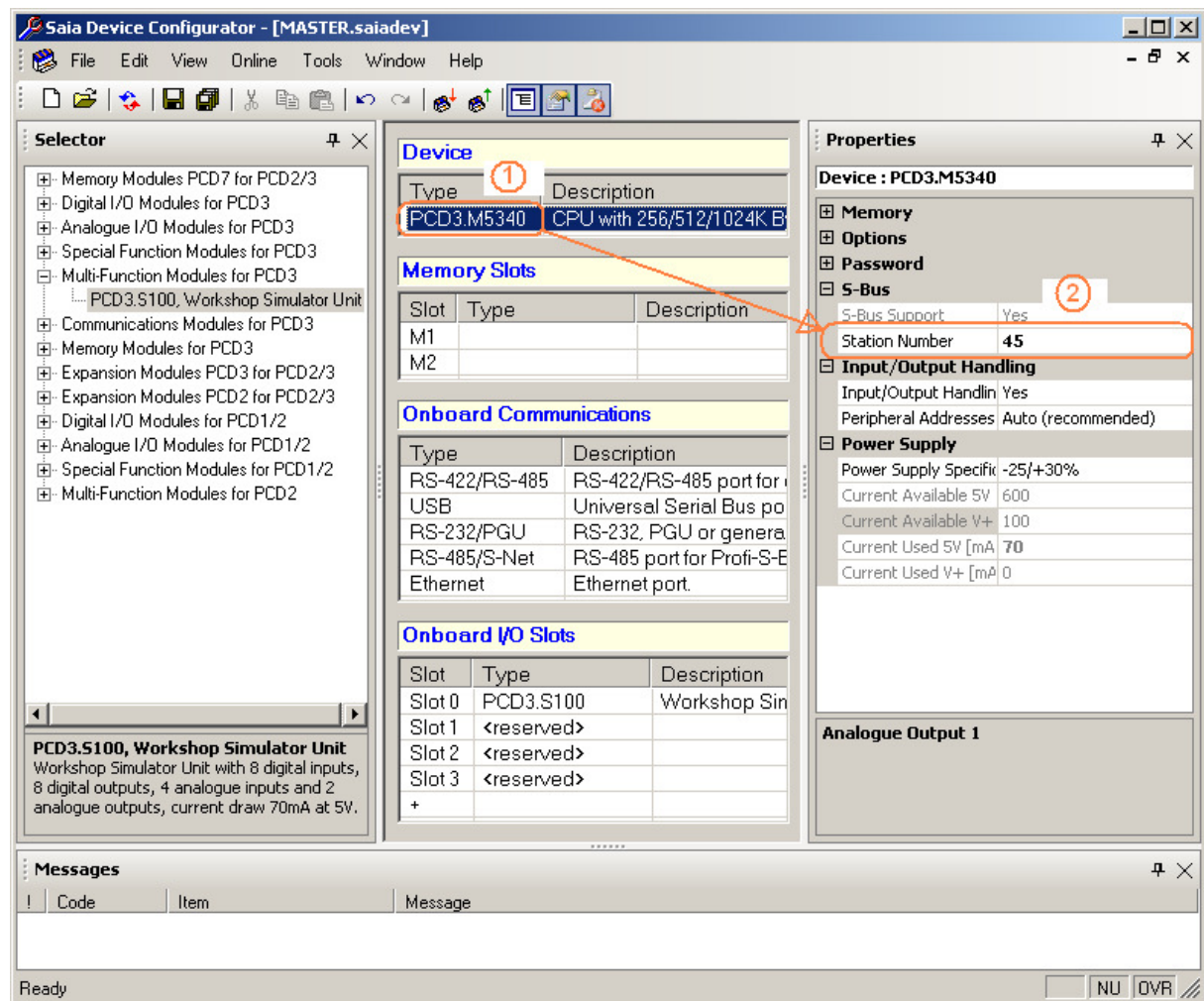
La configuration du S-Bus address et de l’IP address est toujours la première chose à faire, avant de programmer la communication entre 2 PCD’s.

3.2.1 Sur la STATION 1: (dans cet exemple, un PCD3.M5340)

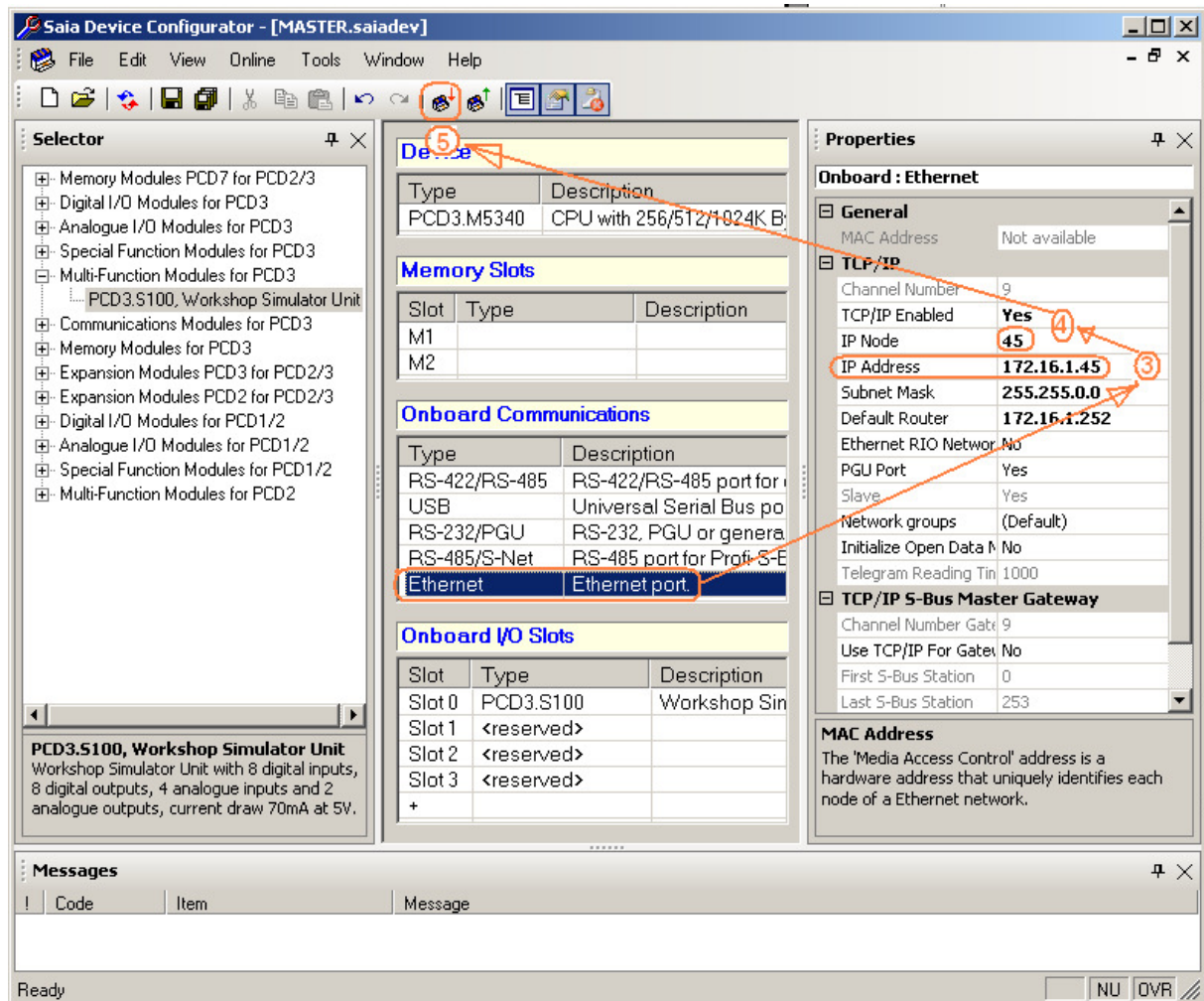
- Créez un nouveau projet (Project → New...).
- Créez un nouveau CPU (Device → New...dans PG5 pour la station master).
- Choisissez dans les **Online Setting** la connection **S-Bus USB** (voir l’image ci-dessous)
- Confirmez en cliquant “OK”



- Ouvrez le **Device Configurator**, choisissez le type de CPU de votre station
- Paramétrez le **SBus Station number** → **45** (voir image ci-dessous)



- Paramétrez l'**IP address** → **172.16.1.45** (avec le *Subnet Mask* et le *Default Router* adéquats)
- Paramétrez l'**IP node** → **45** (est utilisé par le *user-program* et fait référence à l'*IP address*)
- Branchez le câble USB sur la STATION 1.
- Chargez ensuite les *hardware settings* sur la STATION 1 en cliquant sur le bouton "Download configuration" (voir image ci-dessous)



➔ A présent le **S-Bus station number** , l'**IP address** et l'**IP Node** sont configurés dans la STATION 1.



Assurez-vous que chaque IP address soit unique et utilisée qu'une seule fois sur le réseau

3.2.2 Sur la STATION 2: (dans cet exemple, un PCD3.M5540)

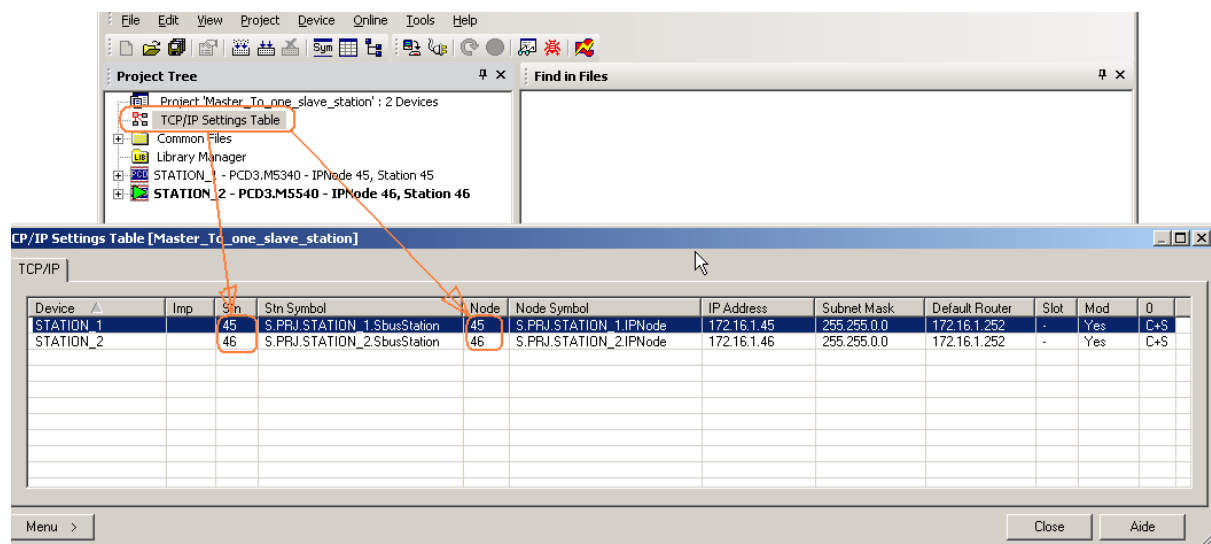
- Procédez aux mêmes étapes que le chapitre 3.2.1 mais adaptez les paramètres réseau pour cette station (slave)
- Changez:
 - S-Bus Station number → 46
 - IP Address → 172.16.1.46
 - IP Node → 46

3.3 TCP/IP Setting Table

Avant de commencer à programmer, vous devez comprendre ce que représente la **TCP/IP Setting Table**.

Après que vous ayez configuré le *S-Bus station number*, l'*IP address* et l'*IP Node* dans la STATION 1 et 2, vous pouvez observer que la **TCP/IP Settings table** est apparue.

Cette table (voir image ci-dessous) contient la **SBus address**, l'**IP address** et l'**IP Node** des stations qui sont configurées et qui doivent être présentes physiquement sur le réseau Ether-S-Bus.



Dans l'image ci-dessus:

- **Stn**: est identique au *S-Bus address*
- **(IP) NODE**: UTILISÉE PENDANT LA PROGRAMMATION
(fait référence à l'*IP address* de chaque station)
- **IP Address**: NON-UTILISÉE PENDANT LA PROGRAMMATION
(uniquement stocké dans la configuration de chaque PCD)

→ Dans ce cas précis, les deux STATION 1 et 2 sont présents dans ce réseau.

Ces 2 informations, **Stn** et **IP NODE** sont utilisés dans le *user program* (dans les Fbox's), afin d'établir la communication Ether-S-Bus.



Cette TCP/IP setting table est chargée automatiquement en même temps que le user program de chaque automate

Ainsi fait, chaque station connaît la *S-Bus address* et l'*IP address* de chaque autre station présente sur le réseau et peut communiquer facilement avec les autres

stations en utilisant ces 2 paramètres réseaux ([Stn](#) et [IP NODE](#)).

3.4 Comment définir une station SLAVE

Il n'y a pas de configurations supplémentaires à effectuer pour définir une station comme *SLAVE* (esclave)

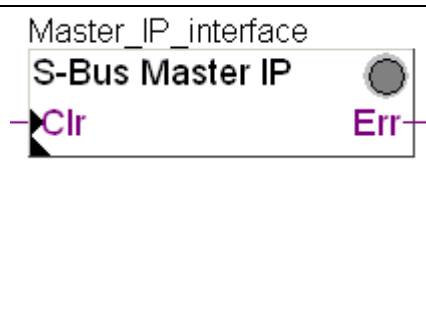
Aussitôt que vous avez configuré et chargé le **S-Bus address**, l'**IP address** et l'**IP Node** (voir chapitre: 3.2), la station fonctionne déjà comme station SLAVE sur le réseau Ether-S-Bus.

3.5 Comment définir une station MASTER

Vous devez configurer le **Sbus address**, l'**IP address** et l'**IP Node** dans une station (voir chapitre: 3.2), après quoi la station fonctionne comme SLAVE station.

Pour la transformer en station MASTER, vous devez effectuer les étapes suivantes:

- créez un fichier FUPLA
- ajoutez une FBox **SASI SBus Master IP**: elle permet d'initialiser l'interface Ethernet de votre automate en tant que MASTER (, la fonction SLAVE reste active)

	<p>Dans les propriétés ("Properties") de la Fbox:</p> <p>→ Choisissez le nom ("name") de cette FBox qui sera utilisé comme "reference name" pour les autres FBox's de communication</p> <p>→ Choisissez le numéro de port ("channel") correspondant à l'interface Ethernet de votre PCD</p>
---	---

→ Cette FBox (ci-dessus) **definit la STATION comme station MASTER**



Le protocole Ether-S-Bus a l'avantage qu'un PCD peut être simultanément MASTER et SLAVE. Ainsi un PCD peut être commandé par le PG5 par exemple et en même temps agir en tant que MASTER pour lire les media (flags, registres, ...) d'un autre PCD à travers l'Ether-S-Bus

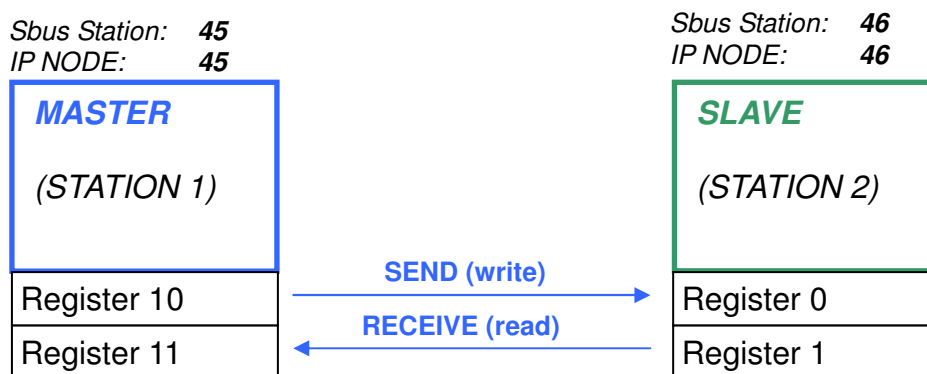
4 Exemple

Dans ce chapitre, vous pouvez utiliser n'importe quels PCD's (voir chapitre **Error! Reference source not found.**).

4.1 Programme d'exemple 1: MASTER to SLAVE communication

Dans cet exemple, nous utilisons 2 PCD3.Mxxxx et les paramètres réseau ont déjà été configurés (voir chapitre 3.2) sur les 2 stations. Nous devons maintenant (avec le *FUPLA editor*) programmer la communication entre la STATION 1 et 2. C'est la raison pour laquelle, il est nécessaire de créer un programme pour la **STATION 1** pour le transformer en station **MASTER**.

4.1.1 Diagramme d'échange de données (Master to Slave)



Le SLAVE répond automatiquement aux requêtes de toutes station MASTER pour lire (read)/écrire (write) ses ressources (registres, flags, ...)

Le MASTER envoie une requête **SEND (write)** : (programmé en FUPLA)
→ il va **écrire** la valeur du REGISTRE 10 dans le REGISTRE 0 du SLAVE

Le MASTER envoie une requête **RECEIVE (read)** : (programmé en FUPLA)
→ il va **lire** la valeur du REGISTRE 1 du SLAVE et la copié dans son REGISTRE 11



GARDER BIEN EN TETES QUELS VARIABLES DU SLAVE SONT UTILISEES, IDEM POUR LES VARIABLES DU MASTER ET CE QUI EST EXACTEMENT ECHANGE ENTRE EUX

4.1.2 Configurez le SLAVE

Dans PG5, du côté SLAVE (STATION 2), aucune programmation n'est exigée. La STATION 2 est configuré comme **SLAVE** sur ce réseau Ether-S-Bus aussitôt que les paramètres réseau (3.2) sont chargés sur le PCD.

Pour être une station SLAVE, a PCD a besoin de:

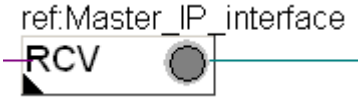
- 1) **SBus Address**
- 2) **IP address**
- 3) **IP Node** (utilisé durant la programmation au lieu de l'*IP address*)
- 4) Le **User-program** est optionnel

Il y va de soi que vous pouvez ajouter un programme comme vous le souhaitez pour votre application.

4.1.3 Programmez le MASTER

Dans PG5, du côté MASTER (= STATION 1):

- Voir **Chapitre 3.5**
 - “**name**” = “*Master_IP_interface*” (dans cette exemple)
 - “**channel**” = **9** (dans cet exemple)
 - pour transformer la Station 1 en station **MASTER**
- ajoutez une FBox **Receive Integer** : (lit (**read**) la valeur de la station distante/**SLAVE**)



Dans les propriétés (“**Properties**”) de la Fbox:

→ **reference name** = “*Master_IP_interface*” (c’est le nom de la FBox **SASI S-Bus Master IP**)

→ **IP-Node** = **46** (IP address du SLAVE)

→ **Source station** = **46** (S-Bus address du SLAVE que vous voulez accéder)

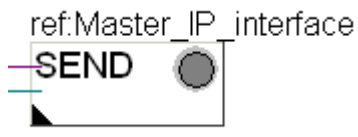
Ces informations sont aussi contenus dans la **TCP/IP setting table** (section 3.3)

→ **Source element** = **Register**

→ **Source address** = **0** (adresse du media sur le SLAVE, que vous voulez lire (**read**).)

→ La sortie de la Fbox doit être connectée au **Registre 10** de la station local/MASTER (pour cet exemple)

- ajoutez une Fbox **Transmit Integer**: (écrit (**write**) une valeur dans la station distante/**SLAVE**)



The diagram shows a rectangular box labeled 'SEND' with a circular port on its right side. A line connects this port to the text 'ref.Master_IP_interface' located above it.

Dans les propriétés (“**Properties**”) de la Fbox:

- **reference name** = “Master_IP_interface”
- **IP-Node** = **46**
- **Source station** = **46** (S-Bus address du SLAVE que vous voulez accéder)
- **Source element** = **Register**
- **Source address** = **1** (adresse du media sur le SLAVE, que vous voulez écrire (**write**).)
- L’entrée de la Fbox doit être connectée au **Registre 11** de la station local/MASTER (pour cet exemple)

- Faîtes un REBUILD ALL
- Charger le *User-program* dans le PCD.

4.1.4 Testez la communication MASTER to SLAVE

Assurez-vous que la première entrée (**enable** input) de la FBox “SEND” et “RCV” soient à l’état haut. Si cela est fait, la communication marche dans le cas où la LED des FBox’s est verte.

Contrôlez aussi quelles ressources sont écrites ou lues avec le **diagramme d’échange des données**. (voir chapitre : 4.1.1).

Autrement vous avez à disposition un projet prêt à être utilisé à propos de cette exemple.

Nom : **2_PCD_with_MasterToSlave_com.zip**

Vous pouvez le trouvez dans le même dossier compressé où vous avez trouvé ce **Getting Started**.

4.2 Programme d'exemple 2: MASTER to MASTER communication

Dans cet exemple, nous utilisons 2 PCD3.Mxxxx et les paramètres réseau ont déjà été configurés (voir chapitre 3.2) sur les 2 stations.

Le procédé essentiel pour programmer cette communication est expliqué dans le chapitre 4.1. La seule différence est qu'à présent les 2 stations fonctionnent comme MASTER (parallèlement ils seront aussi SLAVE)

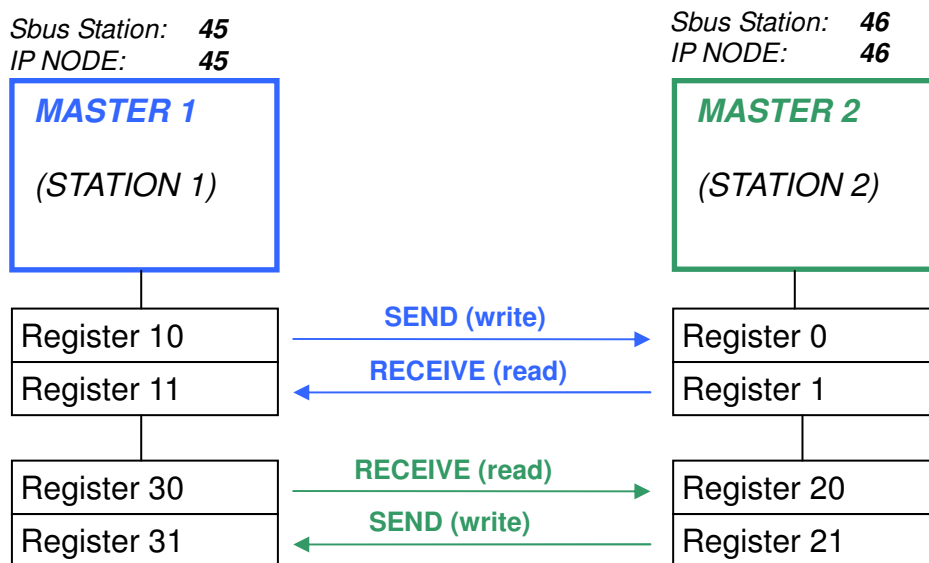
Maintenant nous allons programmer les:

- **STATION 1** comme **MASTER 1**
- **STATION 2** comme **MASTER 2**

Comme il a été expliqué, la FBox **SASI SBus Master IP** ajoute à une station la capacité de lire/écrire les données d'une autre station.

Nous allons donc faire ceci pour la STATION 2 et la transformer en une station **MASTER 2**. C'est pourquoi nous avons besoin de créer un programme pour la **STATION 2**.

4.2.1 Diagramme d'échange de données (Master to Master)



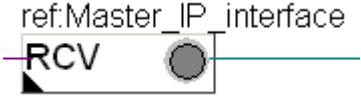
GARDER BIEN EN TÊTES QUELS VARIABLES DU MASTER 1 SONT UTILISEES, IDEM POUR LES VARIABLES DU MASTER 2 ET CE QUI EST EXACTEMENT ECHANGE ENTRE EUX

4.2.2 Configurez le MASTER 2

Similaire au chapitre 4.1.3, mais adaptez les paramètres réseaux pour la station MASTER 2.

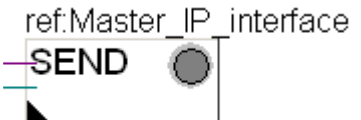
Dans PG5, du côté MASTER 2 (= STATION 2),

- See **Section 3.5** :
 “**name**” = “*Master_IP_interface*” (dans cet exemple)
 “**channel**” = **9** (dans cet exemple)
 pour transformer la Station 2 en station **MASTER 2**
- ajoutez une FBox **Receive Integer** : (lit (**read**) la valeur de la station distante / **MASTER 1**)



- **reference name** = “*Master_IP_interface*”
- **IP-Node** = **45** (IP address du **MASTER 1**)
- **Source station** = **45** (S-Bus address du **MASTER 1**)
- **Source element** = **Register**
- **Source address** = **30** (adresse du média du **MASTER 1**, que vous voulez lire (**read**).)
- La sortie de la Fbox doit être connectée au **Registre 20** de la station local/ **MASTER 2** (pour cet exemple)

- ajoutez une Fbox **Tranmit Integer** : (écrit (**write**) une valeur dans la station distante / **MASTER 1**)



- **reference name** = “*Master_IP_interface*”
- **IP-Node** = **45** (IP address du **MASTER 1**)
- **Source station** = **45** (Sbus address du **MASTER 1**)
- **Source element** = **Register**
- **Source address** = **31** (adresse du media du **MASTER 1**, que vous voulez écrire (**write**).)
- L'entrée de la Fbox doit être connectée au **Registre 21** de la station local/ **MASTER 2** (pour cet exemple)

Comme d'habitude, faite d'abord un *REBUILD ALL* et ensuite un *Download*.

4.2.3 Configure the MASTER 1

LES MÊMES CONFIGURATIONS SONT EFFECTUÉES DANS LE CHAPITRE 4.1.3

4.2.4 Testez la communication MASTER to MASTER.

Assurez-vous que la première entrée (**enable** input) de la FBox “SEND” et “RCV” soient à l’état haut. Si cela est fait, la communication marche dans le cas où la LED des FBox’s est verte.

Contrôlez aussi quelles ressources sont écrites ou lues avec le **diagramme d’échange des données**. (see section 4.2.1)

Autrement vous avez à disposition un projet prêt à être utilisé à propos de cette exemple.

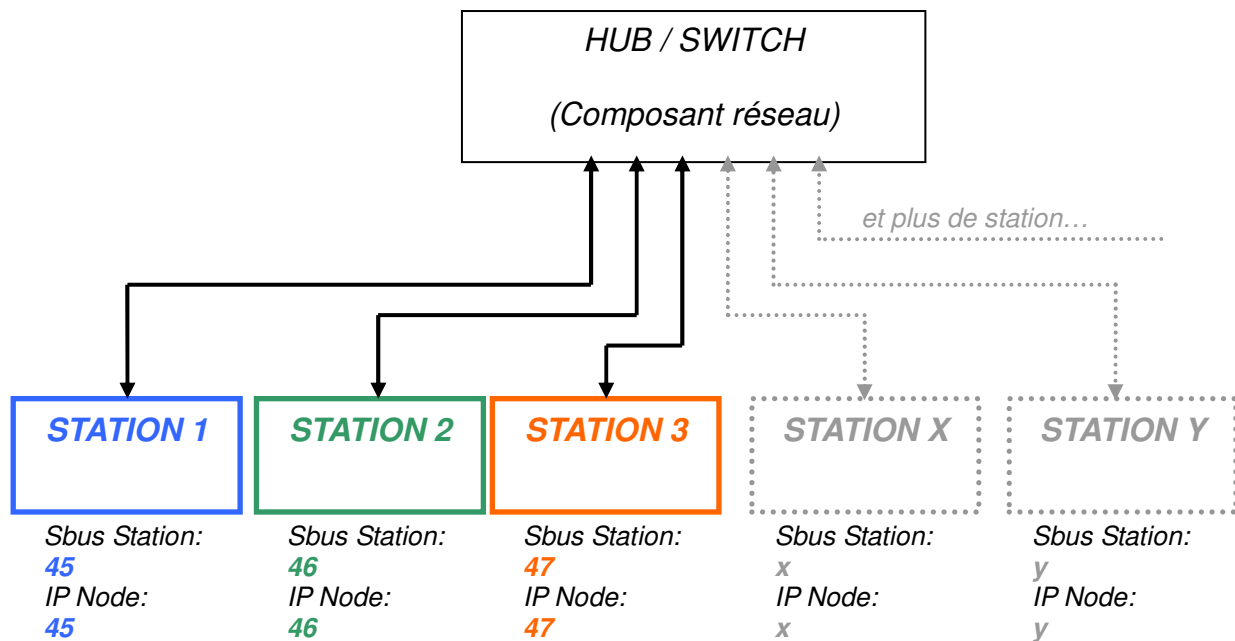
Nom : ***2_PCD_with_MasterToMaster_com.zip***

Vous pouvez le trouvez dans le même dossier compressé où vous avez trouvé ce **Getting Started**.

5 Communication entre plus de 2 PCD's

5.1 Plan réseau

Ci-dessous, vous pouvez voir un exemple de réseau avec plusieurs stations. Toutes ces stations ont leurs propres paramètres réseau. Vous pouvez modifier leurs paramètres comme vous le souhaitez.



5.2 Programmation avec des stations additionnelles

- Comme vous pouvez le voir dans le plan réseau du dessus, la mise en réseaux de plus de 2 stations est seulement possible avec un **HUB** ou un **SWITCH** (un composant réseau)
- Le câble Ethernet croisé est seulement utilisé quand la communication doit être réalisée uniquement entre 2 PCD's.
- Dans votre projet PG5, créez le même nombre de CPU que le nombre de stations qui seront présentes dans votre réseau Ether-S-Bus.
- Comme il a déjà été expliqué (**chapitre 3.2**), la première chose à faire pour intégrer une STATION (PCD) dans un réseau, c'est la configuration du a) **S-Bus address**, b) de l'**IP Address** et c) de l'**IP node** de chaque STATION.

- Aussitôt que ces 3 paramètres ont été introduites, chaque station est à considérer en tant que station SLAVE (**section 3.4**)
- Maintenant choisissez quelle(s) station(s) doit(doivent) être station MASTER et programmez le(les). (**chapitre 3.5**)
- Programmez l'échange de données entre les différentes station MASTER et SLAVE (**chapitre 4.1.3**)

6 Possibilités d'amélioration des performances

6.1 Échanger le plus grand nombre possible de données par FBox "SEND" ou "RCV"

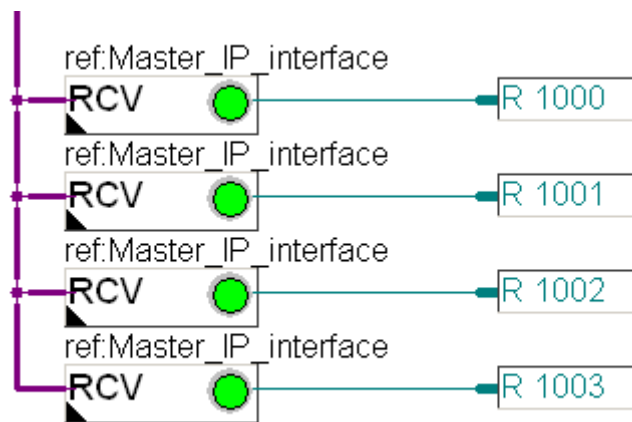
Pour la programmation de la communication en Ether-S-Bus, vous utilisez soit la FBox **SEND** soit la FBox **RCV** pour écrire/lire des ressources d'une station distante.

1ère possibilité: (pas la meilleure)

Par exemple, si vous devez lire 4 registres d'une station distante, il serait possible d'introduire dans le fichier Fupla 4 FBox's **RCV Integer** correspondantes.

Cela signifie que le PCD enverra 4 télégrammes de requête sur le réseau Ether-S-Bus.

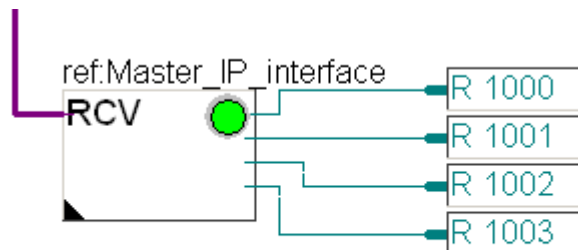
L'avantage dans cette méthode est qu'aucune préparation pour l'organisation des "blocs" de ressources n'est nécessaire.



2ème possibilité: (meilleur)

Pour améliorer la lecture de 4 registres d'une station distante, utilisez seulement une seule FBox **RCV Integer** (au lieu de 4 FBox **RCV Integer**).

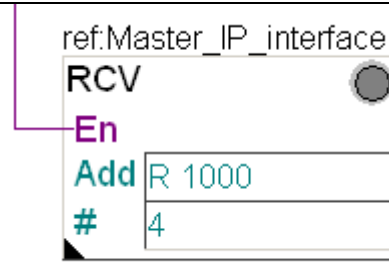
Ceci permet de décharger le réseau Ether-S-Bus car dans ce cas un seul télégramme de requête sera envoyé pour lire simultanément les 4 registres sur la station distante.



Astuce:

Avec la Fbox **“Read Integer quick”**, jusqu’à 32 registres (= 128 flags aussi) peuvent être lus à la fois.

Il s’agit du plus grand nombre de données possible par télégramme (Ether-)S-Bus et ainsi le plus efficace.

Remarque:

Dans un réseau, où peu de télégrammes circulent, la façon de lire ou écrire les ressources de la station distante n’a pas réellement d’importance, si vous utilisez 4 FBox’s pour lire/écrire 4 registres ou si vous utilisez 1 seule Fbox pour lire/écrire 4 registres.

Mais si le réseau est vraiment (sur-)chargé, la 2^{ème} façon de lire/écrire est meilleure que la 1^{ère}. Vous le remarquerez.

6.2 FBox: SASI-Station IP

La Fbox **SASI-Station IP** est très utile dans la situation suivante:

Dans un réseau qui est composé de plusieurs stations SLAVE et d’au moins 1 station MASTER, il peut se produire que plusieurs SLAVE’s perdent leur connexion avec le MASTER pour diverses raisons (mauvais paramètres réseau, câble cassé, etc...).

Comme résultat, si les entrées “enable” de chaque Fbox RCV/SEND sont toujours à 1, ces FBox’s enverrons toujours des requêtes de lecture/écriture vers les SLAVE’s (aussi à ceux qui sont déconnectés) and ceci augmentera fortement le temps d’accès de chaque station connectée.

Exemple d’un long temps d’accès

- Vous avez programmez la communication pour 10 stations slave.
- 9 slaves sont défectueux (à cause d’un mauvais adressage, câble coupé, etc...)
- Seul 1 seul slave communiqué bien
- Response timeout: 250 [ms] (appliqué pour chaque station slave)
- Number of retry: 3 fois (appliqué pour chaque station slave)

Le master a besoin d’environ **6.75 [s]** pour accéder la seule station qui fonctionne.

Pourquoi ?

Chaque fois que le master n’arrive pas à atteindre un slave en lui envoyant une requête, le master essaiera d’envoyer 2 autres requête avec un *Response Timeout* de 250[ms] par défaut. Ensuite il essaie d’atteindre la prochaine station (qui ne

marche n'ont plus pas) et ainsi de suite...

Cela signifie que le temps pour atteindre une station slave connectée est égale au temps résultant à la somme de nombre de *Retry* par station slave non-connectée, multiplié par le *Response Timeout*.

→ **6.75 [s] = 3 (retry) x 0.25 (timeout) x 9 (stations défectueuses)**

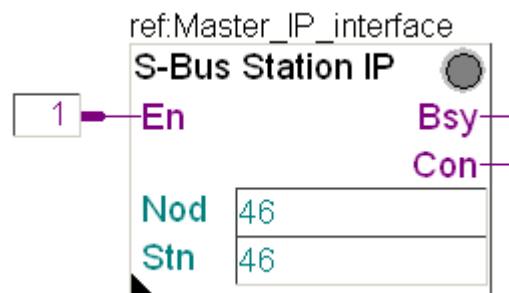
Ceci est problématique si la ressource exigée (du slave fonctionnant) change chaque seconde parce que vous ne pourrez l'accéder qu'approximativement chaque 6.75[s] Et l'effet sera d'autant plus grand avec 100 stations déconnectées ou plus...

Solution:

Pour éviter ce long temps d'échange de données, ajoutez dans le programme du master une Fbox **S-Bus Station IP** pour chaque communication slave.



La Fbox doit être place avant la Fbox **SEND/RCV** (et bien sûr après la Fbox **SASI S-Bus Master IP**)



Pour plus d'information, veuillez regarder dans le Help de la Fbox.

7 Appendix

7.1 Quelle est la difference entre un HUB et un SWITCH ?







	HUB	SWITCH
Connectivité	<ul style="list-style-type: none"> - Possède plusieurs Port Ethernet - Connecte ensemble différents appareils contenant une interface Ethernet, pour construire un réseau. 	
Comportement		<ul style="list-style-type: none"> - Apprend l'adresse MAC d'un station distante connecté à son Port Ethernet - Enregistre son adresse MAC et le référence/lie avec le Port Ethernet correspondant où la station distante est connecté
Processus d'échange de données	Une trame reçue sur 1 port est toujours dupliquée et transmis sur tous les autres ports.	<p>Chaque trame reçue contient une adresse MAC de destination.</p> <p>Cette adresse de destination MAC est lue et la trame reçue est redirigée vers le Port Ethernet correct.</p>

7.2 Quel "IP Address" et "Masque (IP) de sous-réseau" sont à utiliser ???

Pour ces informations, nous vous recommandons de lire le document "word" qui est contenu dans le même dossier où vous avez ouvert ce "Getting Started".

Nom: ***IP_in_general_application_V2.0.doc***

7.3 Debugging: Signification de la couleur de la LED des Fbox

LED of the FBox:			Description of the probable LED error
SASI Master IP	Station S-Bus IP	SEND / RCV	
	peut être rouge ou vert	peut être rouge ou vert	1) La LED concerné est rouge, mais les PCD's communique correctement → Faites un "clear" de la SASI Master IP Fbox avec l'entré correspondante et la LED devient verte → Pourquoi? Le(s) Slave(s) est/sont eteint(s). Votre station master a été mise sous tension, avant qu'une ou plusieurs slave(s) ne soit/soient mise(s) sous tension, donc le master ne pouvait pas communiquer avec le(s) slave(s), mais maintenant il peut
	peut être rouge ou vert	peut être rouge ou vert	2) Le faux port est configure dans la SASI S-BUS master IP → En étant Online: " SASI error ! " est mentionné dans la FBox → Configurez le port number correct de l'interface TCP/IP disponible sur votre PCD
	peut être rouge ou vert	peut être rouge ou vert	Autrement la LED peut devenir rouge consécutivement dans les cas suivant: 3) , 4) , 5)
peut être rouge ou vert		peut être rouge ou vert	3) La station slave correspondante n'est pas connecté → Controllez si la station slave contient les configurations IP correctes (qui sont utilisées par le master) → Controlez si le slave est physiquement connecté (au même réseau que le master)
peut être rouge ou vert	peut être rouge ou vert		4) Un IP Node (3.3) faux est configuré dans la FBox SEND / RCV → L'IP Node n'existe pas dans le TCP/IP setting table (3.3). → Le slave contenant cette IP Node n'est pas présent sur le réseau.
peut être rouge ou vert	peut être rouge ou vert		5) Un faux numéro de Source Station (3.3) est configuré dans la FBox SEND or RCV ou le slave avec ce numéro de Source Station (= S-Bus Address) n'est pas présent sur le réseau.

8 Contactez nous

Si vous constatez des erreurs dans ce document, faites le nous savoir.

Pour de plus amples informations, vous pouvez nous joindre par mail ou téléphone.

- **Saia-Burgess Controls AG**

Rue de la Gare 18
CH-3280 Morat
Switzerland

- **Technical Support**

Phone: 026 / 672 71 11
Fax: 026 / 672 74 99
Website: www.sbc-support.ch
E-mail : pcdsupport@saia-burgess.com