



## Ethernet pour la série Saia PCD®

**0 Table des matières**

0.1	Historique du document .....	0-4
0.2	Marques déposées .....	0-4

**1 Mise en service**

1.1	Consignes importantes .....	1-1
1.2	Configuration d'un port Ethernet du Saia PCD® avec Saia PG5® .....	1-1
1.2.1	Créer un nouveau projet qui englobe toutes les unités centrales dans Ethernet 1-1	
1.2.2	Effectuer les paramétrages matériels .....	1-2
1.2.3	Téléchargement de la configuration.....	1-3
1.3	Connexion en ligne via Ethernet.....	1-4
1.3.1	Sélection des Online Settings (paramétrages en ligne).....	1-4
1.3.2	Etablissement d'une connexion en ligne via Ethernet .....	1-4
1.4	Création d'un programme utilisateur dans Fupla .....	1-6

**2 Caractéristiques matérielles**

2.1	Systèmes Saia PCD® avec interface Ethernet embarquée.....	2-1
2.1.1	PCD3.M3xx0 et PCD3.M5xx0.....	2-1
2.1.2	PCD2.M5540 .....	2-2
2.1.3	PCD1.M2120 .....	2-2
2.1.4	PCD3.T665/T666 Ethernet RIO .....	2-2
2.2	Module Ethernet TCP/IP PCD7.F65x .....	2-3
2.2.1	Schéma synoptique .....	2-3
2.2.2	Structure du PCD7.F650/F655 .....	2-4
2.2.3	Structure du PCD7.F651/F652 .....	2-4
2.2.4	Raccordement PGND .....	2-4
2.2.5	Fonctions des DEL.....	2-5
2.2.6	Affectation des broches de la prise RJ45 .....	2-5
2.2.7	Câblage.....	2-6
2.2.8	Câbles.....	2-7
2.3	Systèmes configurés.....	2-8
2.3.1	PCD1.M135F65x .....	2-8
2.3.2	PCD2.M150F65x .....	2-9
2.3.3	PCD2.M480F65x-2 .....	2-10
2.3.4	PCD4.M170Fx9 .....	2-11
2.3.5	PCD7.F65x sur xx7.....	2-11
2.4	Systèmes compatibles Ethernet non configurés.....	2-12
2.4.1	PCD2.M170 avec PCD7.F65x .....	2-12
2.4.2	PCD7.F65x sur xx7.....	2-12

**3 Caractéristiques et fonctions**

3.1	Connexions possibles et topologies de réseau.....	3-1
3.2	Ether-S-Bus .....	3-3
3.2.1	Topologie et adressage du réseau.....	3-4
3.2.2	Programmation, dépiage des pannes et dépannage via Ethernet .....	3-6
3.2.3	Communication multi-maître .....	3-7
3.2.4	Passerelles et sous-réseaux S-Bus.....	3-8
3.2.5	Règles de communication d'une passerelle S-Bus.....	3-9
3.2.6	Règles de communication d'un serveur OPC .....	3-10
3.2.7	Télégrammes de diffusion.....	3-12

3.3	Open Data Mode via TCP/IP ou UDP/IP.....	3-14
3.4	Conception et structure du réseau.....	3-16
3.4.1	Concentrateurs (réseau en étoile).....	3-16
3.4.2	Les commutateurs.....	3-17
3.4.3	Routeur.....	3-18
3.4.4	Composants de réseau.....	3-19
3.4.5	Amélioration des performances.....	3-19
<b>4</b>	<b>Configuration et programmation</b>	
4.1	Configuration et programmation.....	4-1
4.1.1	Configuration du port IP S-Bus (serveur).....	4-1
4.1.2	Adressage de la station serveur IP.....	4-2
4.1.3	Logements et numéros de canal.....	4-3
4.2	Programmation de S-Bus via Ethernet.....	4-4
4.2.1	Description des instructions du Saia PCD®.....	4-4
4.2.2	Programmation de Ethernet-S-Bus avec des boîtes de fonctions (Saia PG5® FBox).....	4-11
4.2.3	Multi-maître S-Bus IP.....	4-14
4.2.4	Télégrammes de diffusion via S-Bus IP.....	4-14
4.3	Programmation de l'Open Data Mode via Ethernet.....	4-15
4.3.1	Description du mode Open Data.....	4-15
4.3.2	Configuration.....	4-15
4.3.3	Remarques importantes dans UDP et TCP en mode Open Data.....	4-16
4.3.4	Programmation avec une liste d'instructions (IL).....	4-17
4.3.5	InitODM (Open Data Mode).....	4-21
4.3.6	Diagnostic.....	4-25
4.3.7	Nombre de canaux ODM.....	4-30
4.3.8	OpenUDP.....	4-31
4.3.9	OpenClientTCP.....	4-32
4.3.10	OpenServerTCP.....	4-33
4.3.11	Close.....	4-34
4.3.12	ConnectTCP.....	4-35
4.3.13	DisconnectTCP.....	4-36
4.3.14	« GetConnectionTCP ».....	4-37
4.3.15	AcceptConnectionTCP.....	4-38
4.3.16	SendData.....	4-39
4.3.17	SendDataRev.....	4-40
4.3.18	ReceiveData.....	4-41
4.3.19	ReceiveDataRev.....	4-42
4.3.20	Permutation d'octets.....	4-43
4.3.21	Décodage des adresses IP.....	4-44
4.3.22	Flux de connexion TCP type.....	4-45
4.4	CSF supplémentaires.....	4-46
4.4.1	CSF NA-Reset.....	4-46
4.4.2	CSF SetLocalIPNode.....	4-46
4.4.3	CSF IPPhyConfig.....	4-47
4.4.4	CSF SendEtherSBUS.....	4-49
4.4.5	CSF RecvEtherSBUS.....	4-51
4.5	Messages d'erreur Ethernet TCP/IP.....	4-52

<b>5</b>	<b>Diagnostic, dépistage des pannes et dépannage</b>	
5.1	Résumé.....	5-1
5.2	Communication TCP/IP.....	5-1
5.3	Analyse descendante.....	5-1
5.3.1	PING .....	5-2
5.3.2	ARP.....	5-3
5.3.3	Autres instructions utiles pour l'ordinateur hôte .....	5-3
5.4	Analyseur de protocole Ethernet - Wireshark .....	5-4
5.5	Débogage de pile IP du PCD7.F655 via RS-232 du Saia PCD® .....	5-9
<b>6</b>	<b>Exemples de programmation</b>	
6.1	Référence à des liens Ethernet.....	6-1
6.2	Mesures de performances .....	6-1
6.3	Exemple de programmation : Mode Open Data TCP/IP.....	6-2
6.3.1	Serveur .....	6-3
6.3.2	Client.....	6-7
<b>A</b>	<b>Annexe</b>	
A.1	Icônes .....	A-1
A.2	Adresses.....	A-2

## 0.1 Historique du document

0

Version	Date	Publié	Remarques
FR01	-	-	Première édition validée
FR02	2002-06-02	2002-06-02	Adressage 5.7 RxBroadcast_error (bit14) et autres corrections
FR03	2003-08-03	2003-08-03	Nombre de canaux autorisés 5.17 Transfer de données 6.3.12-17
FR05	2010-01-15	2010-02-17	Traduction de l' Edition révisée DE05
FR06	2011-08-02	2011-08-02	Chapitre 2: Full-Duplex et MDIX ajouté
FR07	2012-04-02	2012-04-02	Chapitre 4.3.9: « <b>OpenUDP</b> » remplacé par « <b>OpenClientTCP</b> »
FR08	2013-11-05	2013-11-05	Nouveau logo et nouveau nom de la société
FR09	2014-02-18 2014-04-03	2014-04-03 2014-04-03	Ch4 - organigramme remplacé Ch4 - longueur des télégrammes UDP
FRA10	2016-02-26	2016-02-26	Ch4.2.1 - erreur dans „Exemple d'application“
FRA11	2017-09-01	2017-09-01	Remplacé tous les « <b>IPD</b> » par « <b>IP</b> »
FRA12	2018-06-05 2018-06-21	2018-06-05 2018-06-21	Ch2.2.5 - modifiées les commentaires des tables Ch3.4.5 - ajouté les modes semi-duplex
FRA13	2019-08-08	2019-08-08	Typo corrigé « <b>CTP</b> » → « <b>TCP</b> »

## 0.2 Marques déposées

Saia PCD® et Saia PG5® sont des marques déposées de Saia-Burgess Controls AG.

Les modifications techniques dépendent de l'état de la technologie.

Saia-Burgess Controls AG, 2002. © Tous droits réservés.

Publié en Suisse.

# 1 Mise en service

## 1.1 Consignes importantes

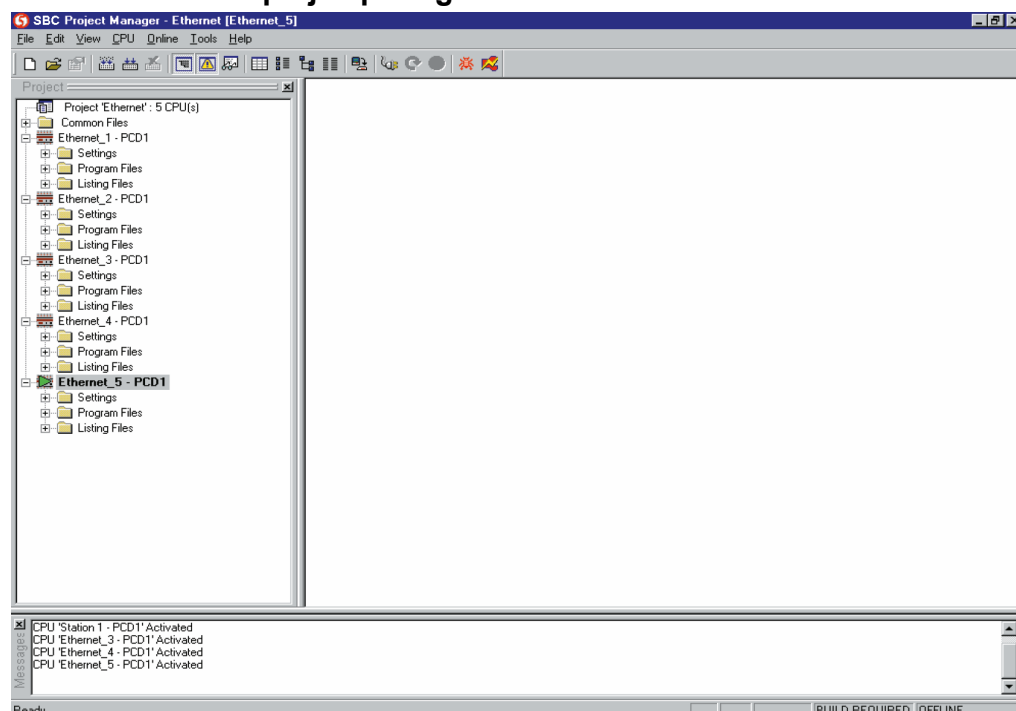
1

- Un seul module Ethernet peut être utilisé par système PCS, hormis sur le PCD2.M480
- Un seul port PGU S-Bus est autorisé par système. Lorsque le module Ethernet est configuré comme PGU S-Bus, aucun autre port que le port PGU S-Bus ne peut être utilisé
- Le module Ethernet peut également être configuré pour le protocole S-Bus réduit (voir instructions suivantes). Dans ce cas, n'importe quel autre port peut être utilisé comme PGU S-Bus
- Une station Ethernet nécessite une table d'affectation avec une adresse IP et un nœud IP pour toutes les stations dans le réseau. Cette table d'affectation est générée automatiquement par la configurateur matériel et téléchargé dans un DBX dans le Saia PCD®. La configuration matérielle de tous les systèmes PCD dans Ethernet doit, en outre, être configurée dans le gestionnaire de projets dans le projet lui-même
- Si plusieurs personnes programment différentes stations Saia PCD® dans le même projet, il est recommandé de définir dans un premier temps l'ensemble de la configuration réseau sur un PC de programmation. Cette configuration doit alors être copiée sur les autres PC de programmation ou importée par les PC de programmation
- La communication via passerelle est uniquement possible dans le sens Ethernet - sous-réseau S-Bus, mais pas dans le sens inverse. Un sous-réseau S-Bus est autorisé par passerelle.

## 1.2 Configuration d'un port Ethernet du Saia PCD® avec Saia PG5®

Pour de plus amples informations, voir l'aide Saia PG5®.

### 1.2.1 Créer un nouveau projet qui englobe toutes les unités centrales dans Ethernet



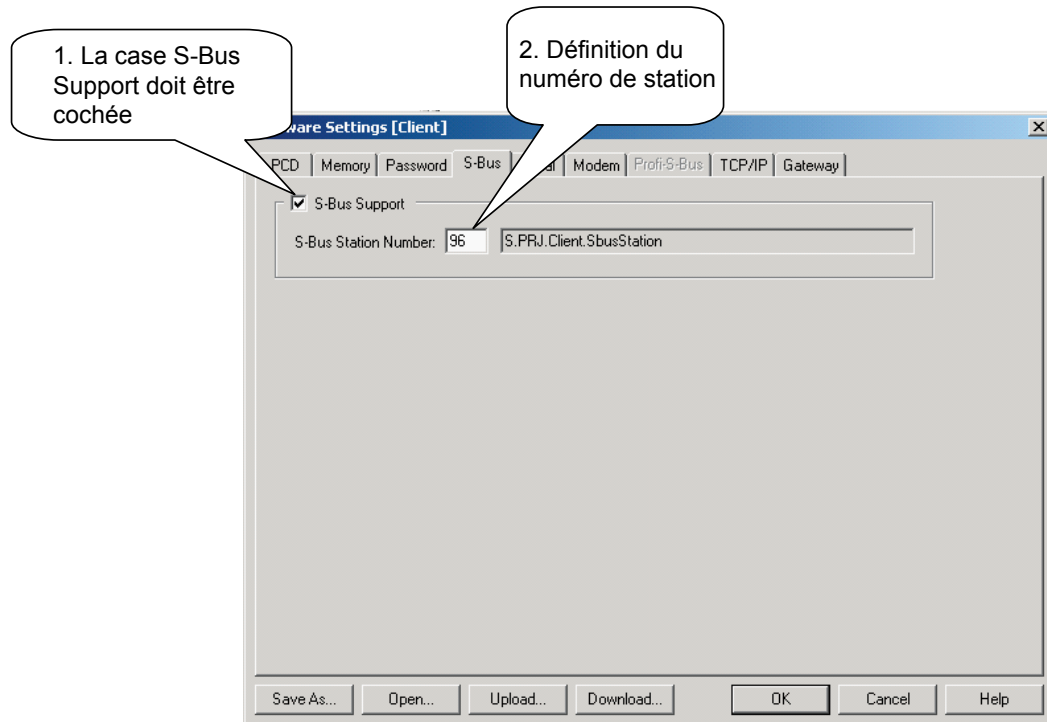
Les paramètres matériels doivent être saisis pour toutes les stations Saia PCD®.

## 1.2.2 Effectuer les paramètres matériels

1

Les fenêtres suivantes sont disponibles pour la configuration d'Ethernet :

- **Paramétrages SBC S-Bus**



- **Paramétrage de passerelle**

La définition est uniquement nécessaire pour une station-passerelle.

Les paramètres sont les mêmes que pour la communication S-Bus standard.

- **Paramétrages du module Ethernet TCP/IP**

Saisie des valeurs suivantes : IP Node , IP Address , Subnet Mask et Default Router.  
 Cette case doit être cochée

Dans le cas d'un seul sous-réseau, la saisie de la valeur Default Router n'est pas obligatoire.

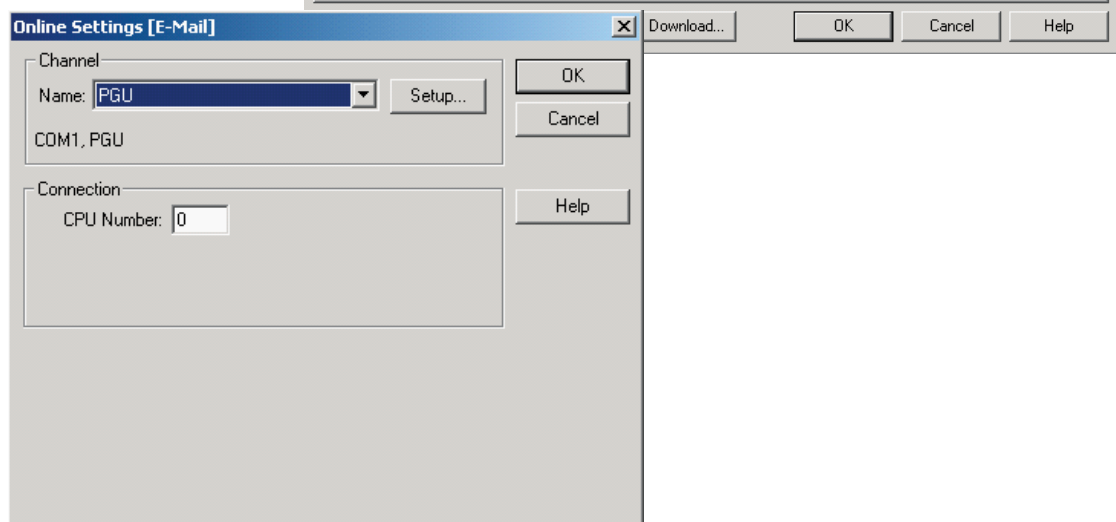
Les paramétrages Channel and Slot (Canal et logement) dépendent si le module Ethernet a été installé.

Pour les applications simples, il n'est pas nécessaire d'adapter le paramétrage pour les Network Groups (groupes de réseaux).

Pour de plus amples informations, voir aussi l'aide Saia PG5®.

### 1.2.3 Téléchargement de la configuration

La première fois, la configuration doit être téléchargée via le port PGU, en utilisant le canal PGU.



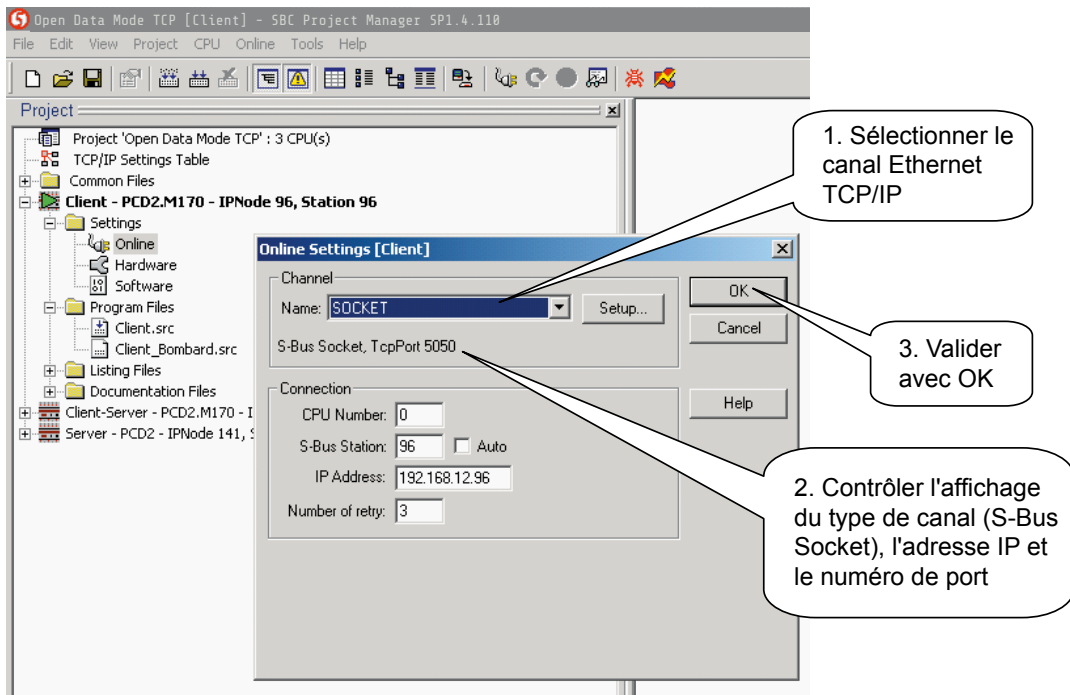
Après avoir sélectionné le PGU Channel (canal PGU), les paramètres de configuration peuvent être téléchargés à partir du menu Hardware Settings (Paramétrages matériels).



### 1.3 Connexion en ligne via Ethernet

#### 1.3.1 Sélection des Online Settings (paramétrages en ligne)

1

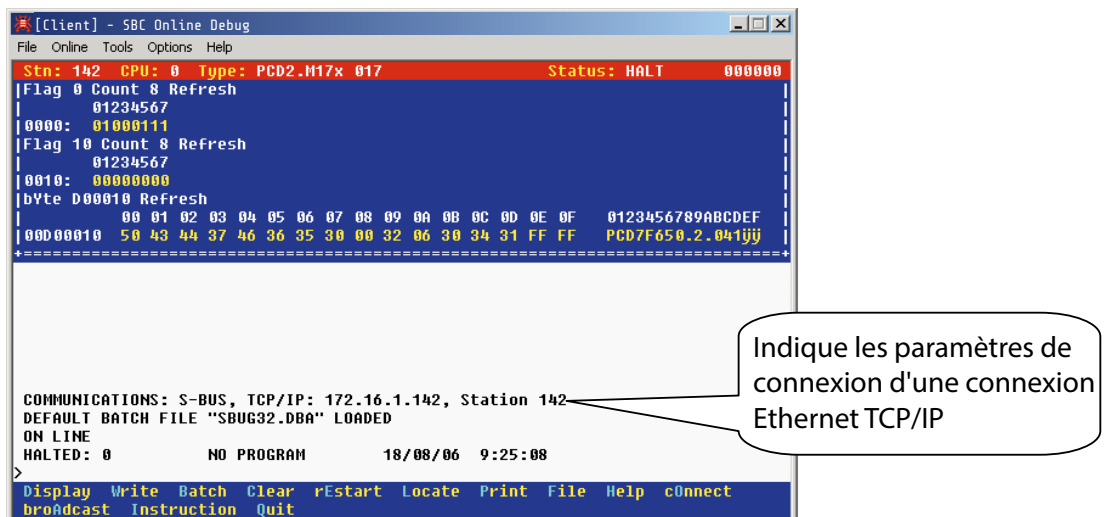


Les valeurs de CPU Number (numéro CPU), S-Bus Station (station S-Bus) et IP Address (adresse IP) sont automatiquement définies en fonction des paramètres matériels de la station sélectionnée ; elles peuvent être modifiées, si nécessaire.

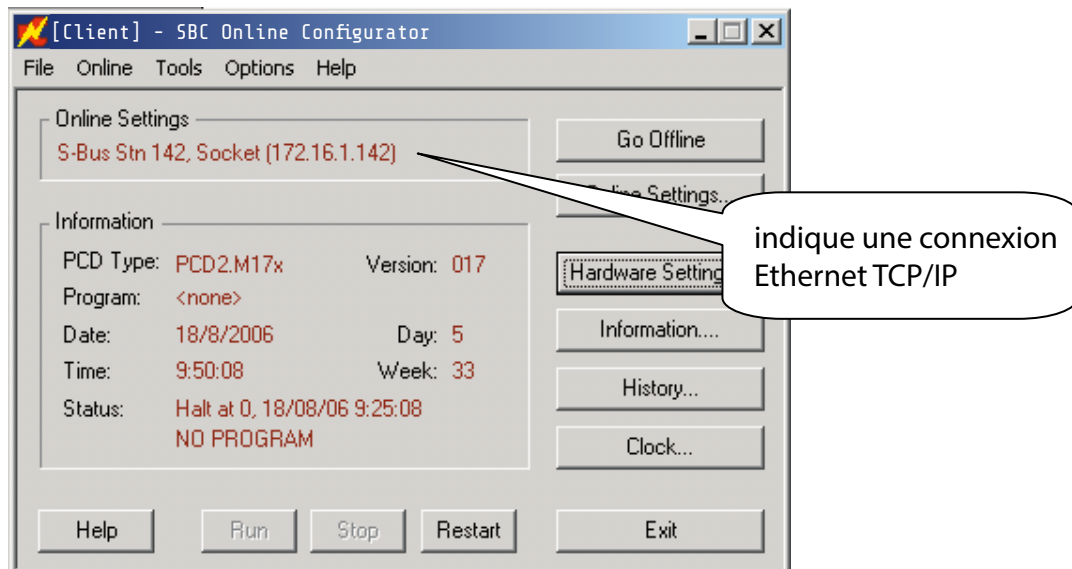
#### 1.3.2 Etablissement d'une connexion en ligne via Ethernet

La connexion avec les Saia PCD® peut être établie à l'aide d'un des outils en ligne suivants.

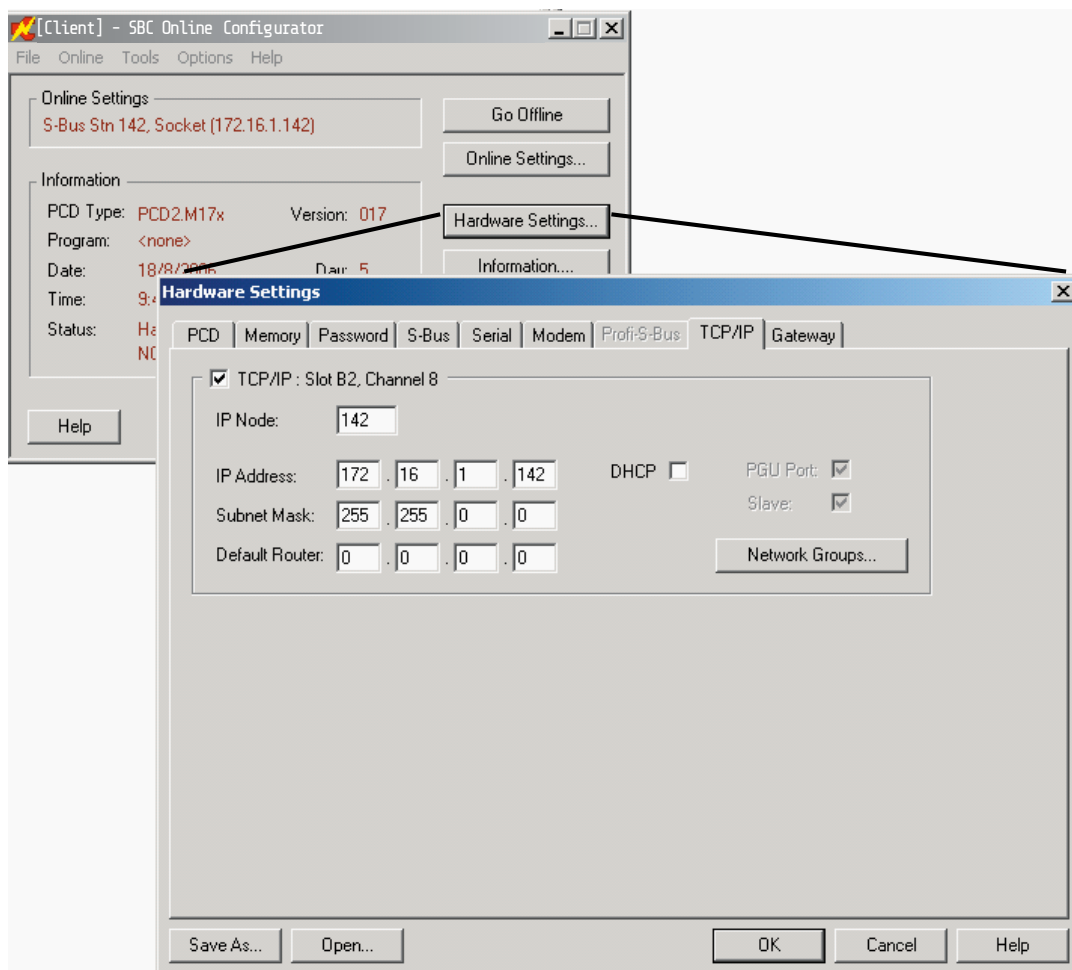
Débugueur :



Configurateur en ligne :



Vous pouvez vérifier ou modifier en ligne les paramètres pour Ethernet TCP/IP du Saia PCD® à l'aide des instructions de commande suivantes :



Si la connexion en ligne ne fonctionne pas malgré un Saia PCD® correctement configuré, voir chapitre 5 : Diagnostic, dépiage et dépannage.

## 1.4 Création d'un programme utilisateur dans Fupla

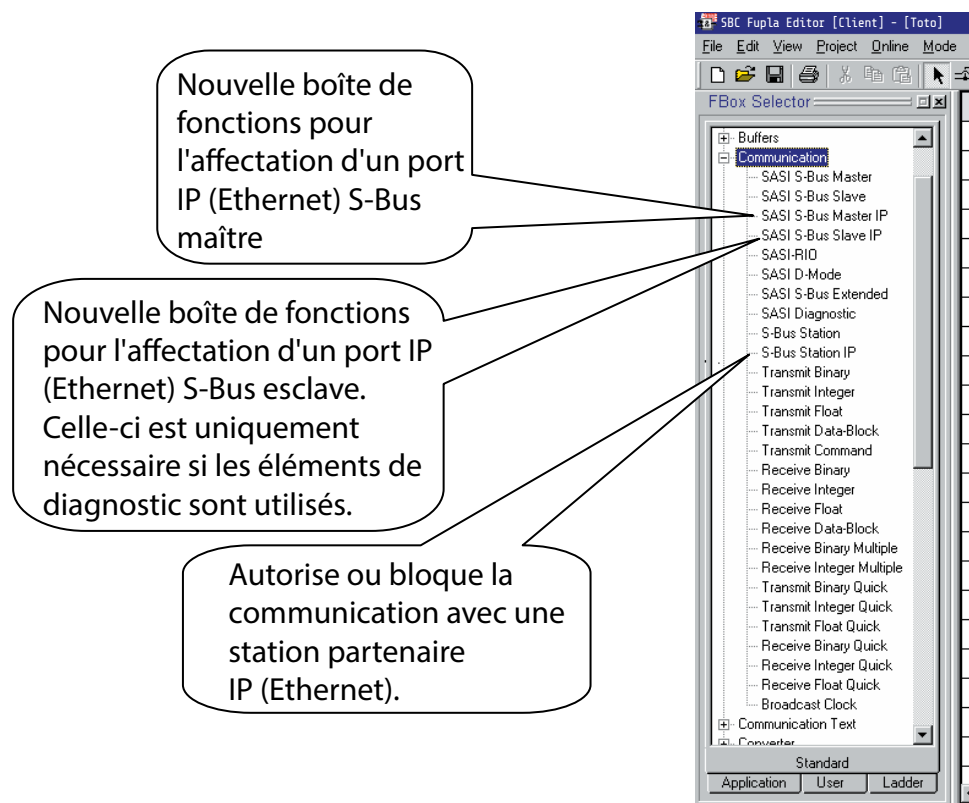
Pour programmer une communication via Ethernet, une bibliothèque de communication à jour est nécessaire.

1

Le protocole Ethernet TCP/IP est pris en charge à partir de la version \$2.2.003.

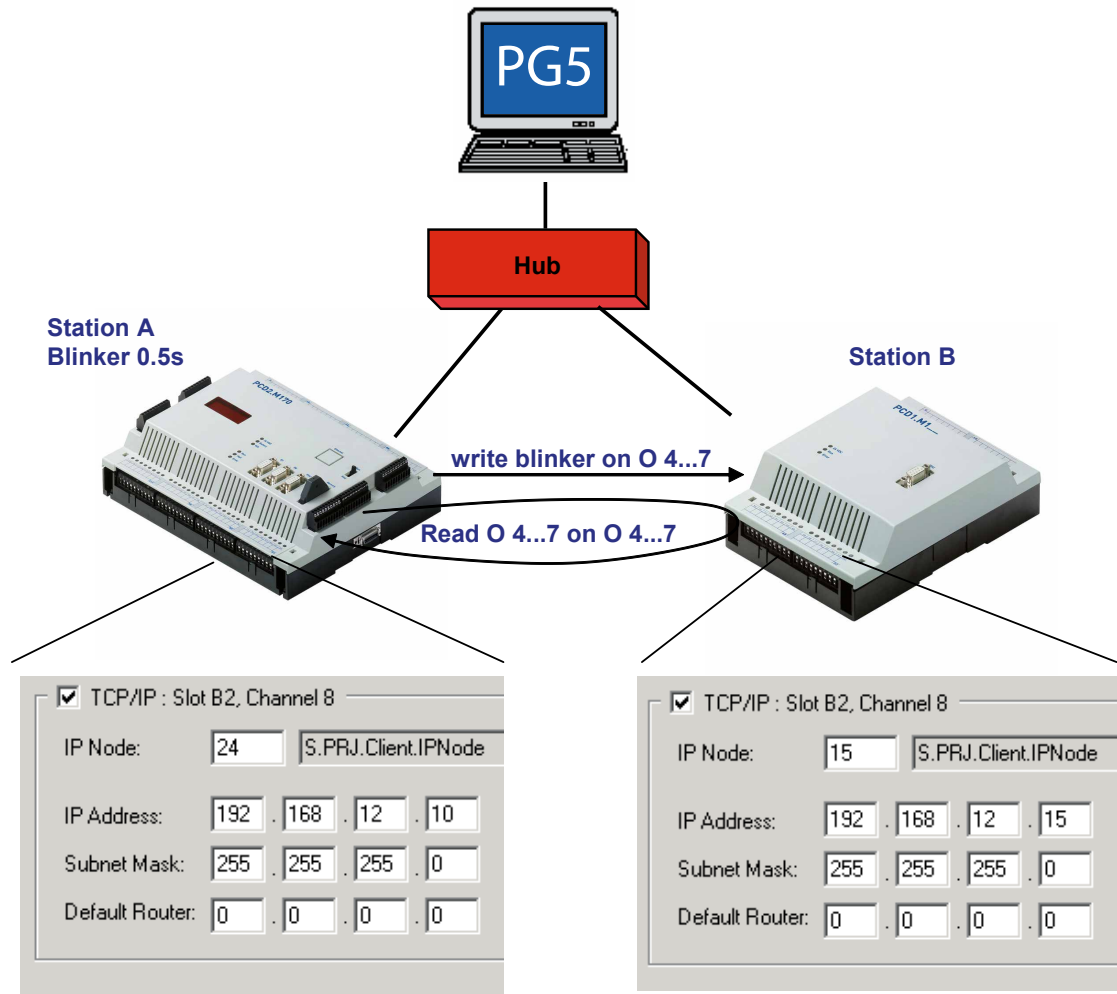
Vérifiez l'installation correcte de la nouvelle bibliothèque.

Pour toutes les boîtes de fonctions d'émission/de réception (Saia PG5® FBox) (ex. : Transmit/Receive Binary), un nouveau paramètre supplémentaire doit être défini : le numéro de nœud IP (IP Node Number).



La communication Ethernet TCP/IP se programme comme la communication S-Bus standard.

Exemple de programmation dans Fupla :



1

**Tâche :**

Un signal clignotant est généré sur la station A. Le signal clignotant est copié via Ethernet TCP/IP dans la station B sur les sorties 4...7. Ces sorties sont alors relues et copiées sur les sorties 4...7 de la station A.

Programme utilisateur dans Fupla :

The screenshot shows the SBC Fupla Editor interface. On the left is the 'FBox Selector' with a tree view of communication functions. The main workspace displays a ladder logic diagram with a 'SEND' block connected to a 'Blink' block. A configuration dialog box titled 'Adjust: Senden Binär' is open, showing settings for a binary send operation. Below the editor is a 'Symbols' table listing outputs out4 through out7.

Group/Symbol	Type	Address/Value	Comment
out4	Output	4	
out5	Output	5	
out6	Output	6	
out7	Output	7	

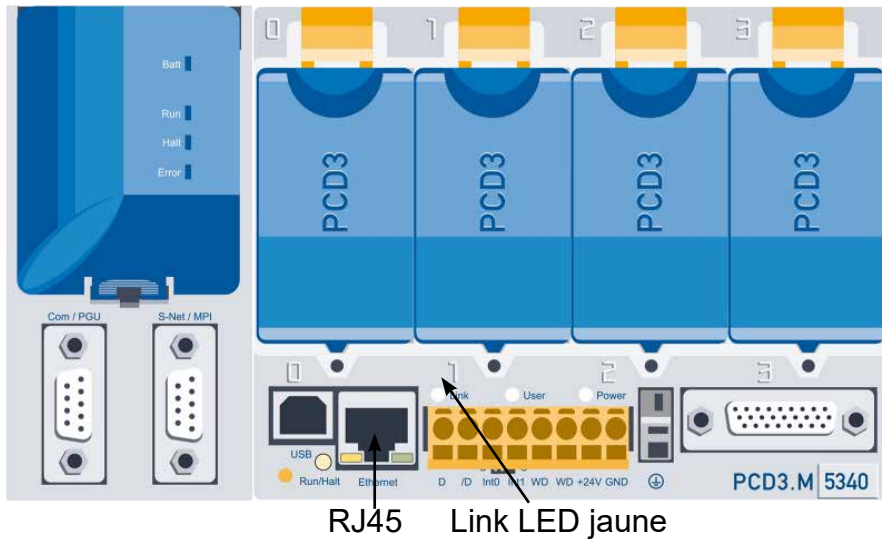
1

## 2 Caractéristiques matérielles

### 2.1 Systèmes Saia PCD® avec interface Ethernet embarquée

Quelques modèles de la famille PCD3.Mxx ont une interface Ethernet embarquée. Le trafic Ethernet utilise le canal 9 sur ces appareils.

2



#### 2.1.1 PCD3.M3xx0 et PCD3.M5xx0

Les types PCD3.M3xx0 et PCD3.M5xx0 avec la version matérielle F ou plus récente supportent le mode full-duplex et Auto-MDIX (croisement automatique des signaux Rx et Tx). Si un PCD3.Mxxx0 supporte les modes full-duplex et Auto-MDIX peut être vu en contrôlant la présence de DEL sur le connecteur RJ45. Si ce dernier est équipé de DEL, les modes full-duplex et Auto-MDIX sont supportés.



**Numéro de commande :**

PCD3.M5540	PCD3 avec Ethernet embarqué	
PCD3.M3120	PCD3 avec Ethernet embarqué	(sans batterie, non-extensible)
PCD3.M3330	PCD3 avec Ethernet embarqué	(sans batterie, extensible E/S)
PCD3.M6340	PCD3 avec Ethernet embarqué	(avec interface LAN)
PCD3.M6540	PCD3 avec Ethernet embarqué	(avec interface maître Profibus DP)
PCD3.M2330A4Tx	PCD3 Wide Area Controller	(WAC)
PCD3.M2130V6	PCD3 Compact	
PCD3.M5560	PCD3 CPU rapide	

PCD3.M6360	PCD3 CPU rapide	(avec interface LAN)
PCD3.M5560	PCD3 CPU rapide	(avec interface maître Profibus DP)

### 2.1.2 PCD2.M5540

Le PCD2.M5540 est équipé d'un commutateur Ethernet intégré avec deux connecteurs. La communication Ethernet utilise le canal 9 sur le PCD3.M5540. Toutes les versions matérielles du PCD3.M5540 supportent le mode full-duplex et Auto-MDIX (croisement automatique des signaux Rx et Tx).

### 2.1.3 PCD1.M2120

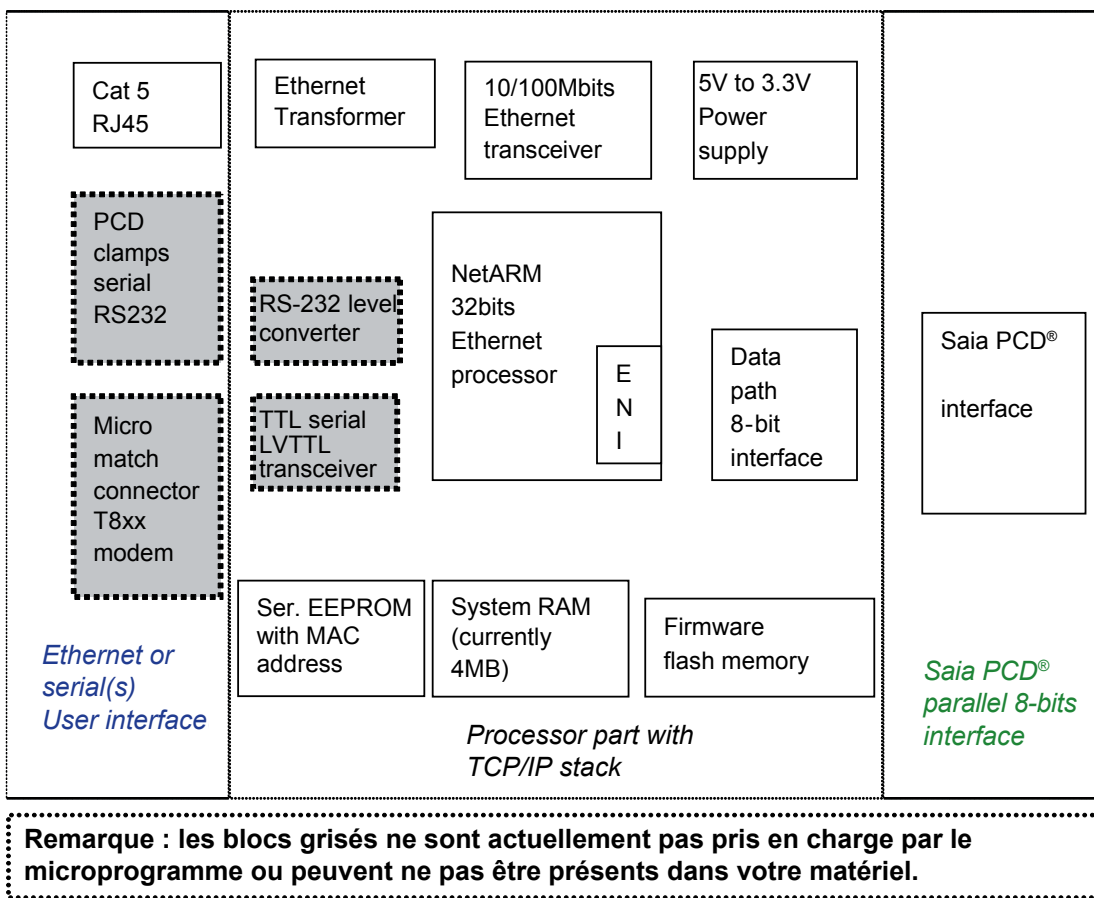
Le PCD1.M2120 est équipé d'un commutateur Ethernet intégré avec deux connecteurs. La communication Ethernet utilise le canal 9 sur le PCD1.M2120. Toutes les versions matérielles du Saia PCD®1.M2120 supportent le mode full-duplex et Auto-MDIX (croisement automatique des signaux Rx et Tx).

### 2.1.4 PCD3.T665/T666 Ethernet RIO

Le PCD3.T665/T666 Ethernet RIO est équipé d'un commutateur Ethernet intégré avec deux connecteurs. La communication Ethernet utilise le canal 9 sur le PCD3.T665/T666. Toutes les versions matérielles du PCD3.T665/T666 supportent le mode full-duplex et Auto-MDIX (croisement automatique des signaux Rx et Tx).

## 2.2 Module Ethernet TCP/IP PCD7.F65x

### 2.2.1 Schéma synoptique



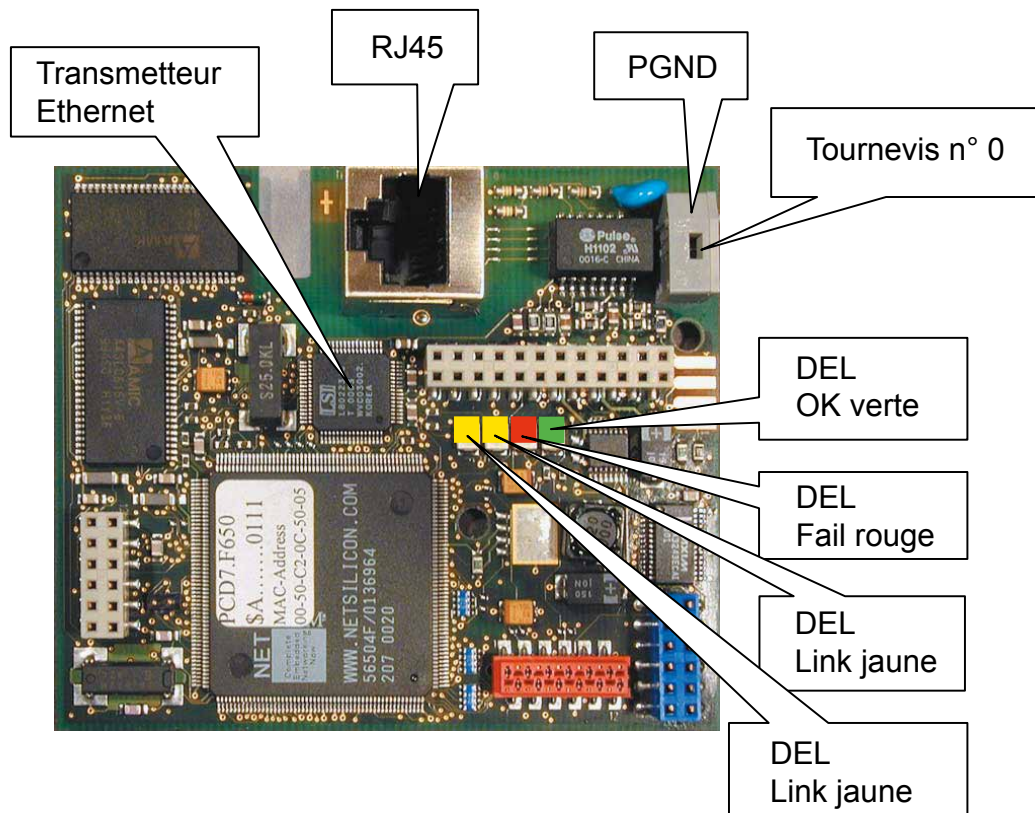
2



Les interfaces série sont gérées par le processeur Ethernet et ne peuvent pas servir d'interfaces Saia PCD® régulières. Ces deux interfaces sont destinées à la connexion IP (PPP, SLIP, etc.) via modem ou à des fonctions de dépiage des pannes et de dépannage.



## 2.2.2 Structure du PCD7.F650/F655



## 2.2.3 Structure du PCD7.F651/F652

La structure correspond à celle du PCD7.F650/F655 à l'exception des différentes variantes de prises RJ45.

## 2.2.4 Raccordement PGND

Afin de garantir le respect des valeurs de compatibilité électromagnétique (CEM), la terre de protection (PGND) doit toujours être reliée à la terre du châssis.

### Raccordement de la terre de protection :

- Ouvrir le bornier à ressort en insérant un tournevis (n°0) dans la fente
- Insérer le fil de mise à la terre dans l'ouverture PGND
- Retirer le tournevis
- Tirer sur le conducteur de mise à la terre de protection pour vérifier s'il est correctement branché

## 2.2.5 Fonctions des DEL

- Fast (jaune) S'allume si une connexion rapide 100 Mbit/s est détectée, s'éteint en cas de connexion 10 Mbit/s
- Link (jaune) S'allume si une connexion est détectée et clignote en cas de transmission Ethernet (arrêt = activité, marche = inactivité)
- Fail (rouge) Clignote si une erreur de matériel ou de microprogramme a été détectée
- OK (vert) Indique l'absence d'erreur (clignote à une fréquence de 2 Hz)



Les deux DEL jaunes ont une autre signification qu'en mode normal, pendant la séquence d'initialisation du processeur. Juste après une réinitialisation (lors de la mise sous tension), la DEL Fast indique une connexion à 10 Mbit/s, tandis que la deuxième DEL Link indique une connexion à 100 Mbit/s.

DEL Link jaune	DEL Fast jaune	Commentaire
LINK 100	LINK10	Après la réinitialisation
LINK + ACT	10/100	Pendant le fonctionnement

La DEL Fail (rouge) indique qu'une erreur a été détectée. La fréquence de clignotement fournit des informations sur l'erreur détectée. La DEL OK peut également clignoter

Selon l'erreur détectée, les DEL verte et rouge peuvent clignoter. Si seule la DEL verte clignote, aucune erreur n'est présente.



Risque de blessures ! Le transmetteur Ethernet est assez chaud. En l'absence de communication, sa température dépasse d'environ 30° la température ambiante. Ceci est normal.

## 2.2.6 Affectation des broches de la prise RJ45

Broche	Nom	Paire de fils torsadés	Couleur de fil	Ethernet	Fast Ethernet	Remarque
1	TPO+	Paire 2	blanc/orange	TPO+	TPO+	
2	TPO-	Paire 2	orange	TPO-	TPO-	
3	TPI+	Paire 3	blanc/vert	TPI+	TPI+	
4	Term. 1a	Paire 1	bleu	-	Paire 1	terminaison à 75 Ω*
5	Term. 1b	Paire 1	blanc/bleu	-	Paire 1	avec broche 4*
6	TPI-	Paire 3	vert	TPI-	TPI-	
7	Term. 4a	Paire 4	blanc/marron	-	Paire 4	terminaison à 75 Ω*
8	Term. 4b	Paire 4	marron	-	Paire 4	avec broche 8*

\* terminaison Bob Smith

Pour Ethernet (10 Mbit/s), les paires 1 et 4 peuvent être absentes.

L'affectation des câbles de Fast Ethernet (câble CAT5) est compatible avec Ethernet 10 Mbit/s.

## 2.2.7 Câblage

### Type de câble :

Ce module est conçu pour être utilisé avec des câbles 100 ohms blindés et non blindés (UTP ou STP). En cas d'interférences électromagnétiques, un câble blindé est fortement recommandé pour un meilleur débit. Si une liaison 100 Mbit/s est utilisée, le câble doit être au minimum de catégorie CAT 5. Pour une liaison 10 Mbit/s, le câble doit être au minimum de catégorie CAT 3.



Ce module utilise la méthode d'auto-négociation pour définir la vitesse de transmission et le mode de fonctionnement. Pour établir la liaison avec 100 Mbit/s et duplex intégral, les deux extrémités doivent prendre en charge l'auto-négociation. Si tel n'est pas le cas, le module établit la liaison avec 10 Mbit/s et semi-duplex. La méthode d'auto-négociation ne détecte cependant pas le type de câble utilisé. Cela signifie que vous pouvez établir la liaison avec un câble CAT 3 à 100 Mbit/s, mais qu'il est possible que celle-ci ne fonctionne pas (ceci peut être le cas, par exemple, avec un câble croisé).

Une fonction CSF (Call System Function) spéciale permet de commuter le PCD7. F65x en mode duplex intégral ou semi-duplex et de communiquer avec 10 Mbit/s ou 100 Mbit/s. Pour de plus amples détails, voir le chapitre relatif aux fonctions CSF spéciales.

### Longueur de câble :

Maximum 100 m

Rayon de courbure :

Certaines limitations s'appliquent au rayon de courbure.

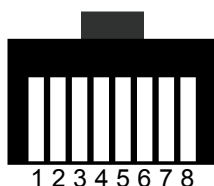
Selon l'EIA SP-2840A, le rayon de courbure minimal est égal à 10 fois le diamètre extérieur du câble. Selon l'ISO DIS 11801, il est égal à 8 fois le diamètre extérieur.

### Câblage :

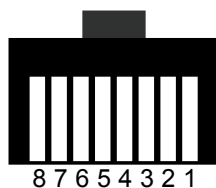
Le câblage peut s'effectuer soit de paire à paire avec un câble croisé, soit au moyen d'un câble de raccordement normal via un concentrateur ou un commutateur.

### Affectation des broches :

**Prise mâle** (vue de l'extrémité de la fiche mâle, de l'avant, câble fuyant)



**Prise femelle** (vue de la prise murale, de l'avant)



## 2.2.8 Câbles

**Affectation des câbles pour câbles croisés (paire à paire) :**

Broche	Couleur		Broche
1	orange/blanc	↔	3
2	orange	↔	6
3	vert/blanc	↔	1
4	bleu	↔	4
5	blanc/bleu	↔	5
6	vert	↔	2
7	blanc/marron	↔	7
8	marron	↔	8

**Affectation des câbles pour câbles de raccordement (via un concentrateur) :**

Broche	Couleur		Broche
1	orange/blanc	↔	1
2	orange	↔	2
3	vert/blanc	↔	3
4	bleu	↔	4
5	blanc/bleu	↔	5
6	vert	↔	6
7	blanc/marron	↔	7
8	marron	↔	8

Couleurs définies selon l'EIA/TIA T568B.

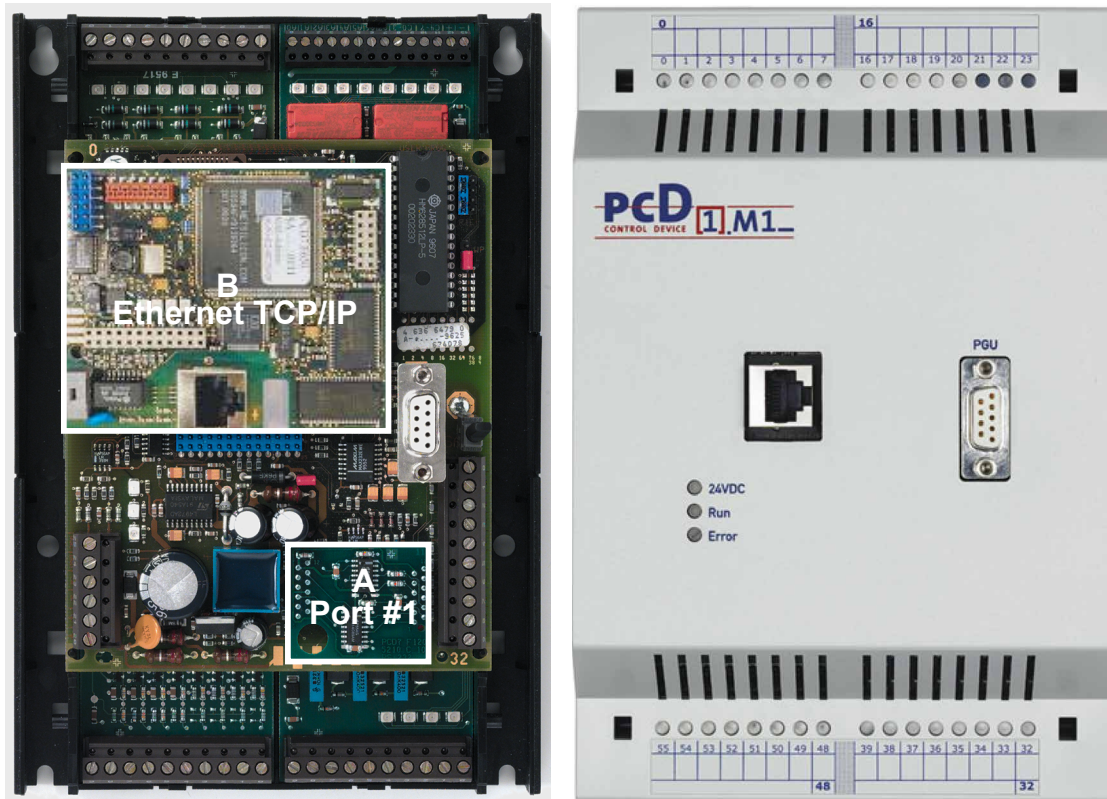
## 2.3 Systèmes configurés

### 2.3.1 PCD1.M135F65x

PCD7.F65x sur logement B / canal 9

PCD1.M135F65x

2



Le fil PGND du PCD7.F65x doit être relié à la borne 23 du PCD1.

#### Pour commander :

- Système configuré :  
PCD1.M135F655      Système configuré PCD1 avec module Ethernet
- Compléments :  
PCD7.F655            Module Ethernet pour PCD1/PCD2  
4'104'7409'0        Capot pour PCD1.M135 avec découpe pour prise RJ45

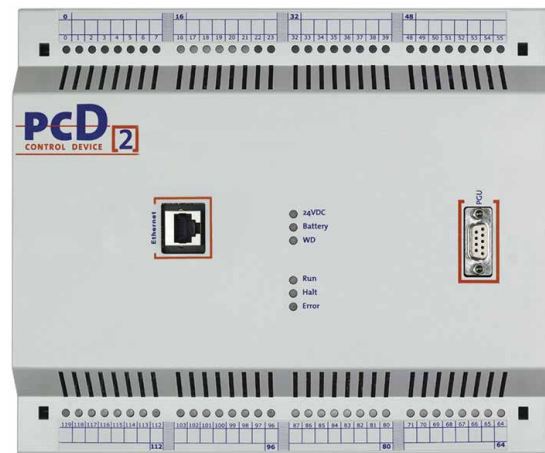
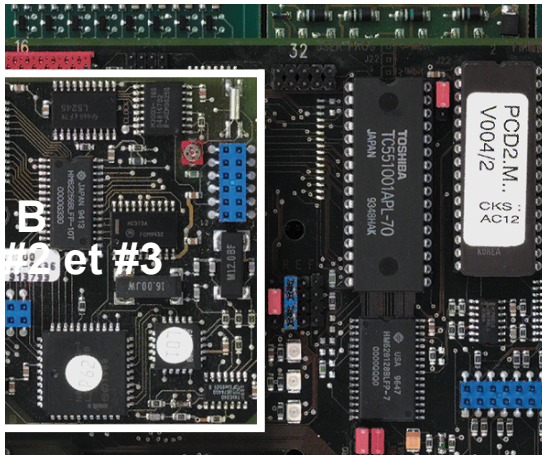


Certaines limitations peuvent s'appliquer aux anciennes versions matérielles du PCD1.

### 2.3.2 PCD2.M150F65x

PCD7.F65x sur logement B / canal 9

PCD2.M150F65x



Le fil PGND du PCD7.F65x doit être relié à la borne 27 du PCD2. Les bornes Saia PCD® 30...39 doivent être réservées pour RS-232 (RS-232 en cours).

#### Pour commander :

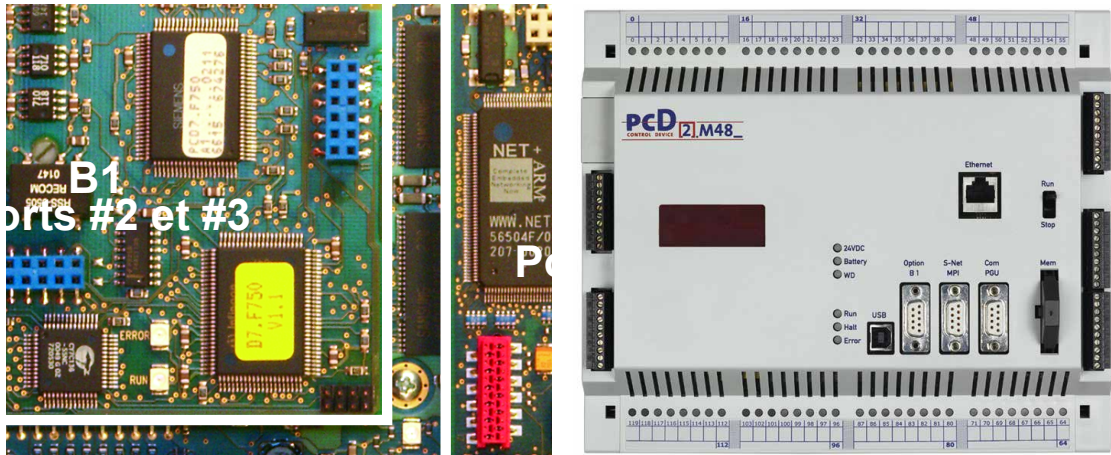
- Système configuré :  

PCD2.M150F655	Système configuré PCD2 avec module Ethernet
---------------	---
- Compléments :  

PCD7.F655	Module Ethernet pour PCD1/PCD2
4'104'7410'0	Capot pour PCD2.M150 avec découpe pour prise RJ45

### 2.3.3 PCD2.M480F65x-2

PCD7.F65x sur logement B2 / canal 8 et également sur logement B1 / canal 9



2



Le fil PGND du PCD7.F65x doit être relié à la borne PGND à côté de B1 DB9. Les bornes Saia PCD® 40...49 doivent être réservées pour RS-232 (RS-232 en cours).

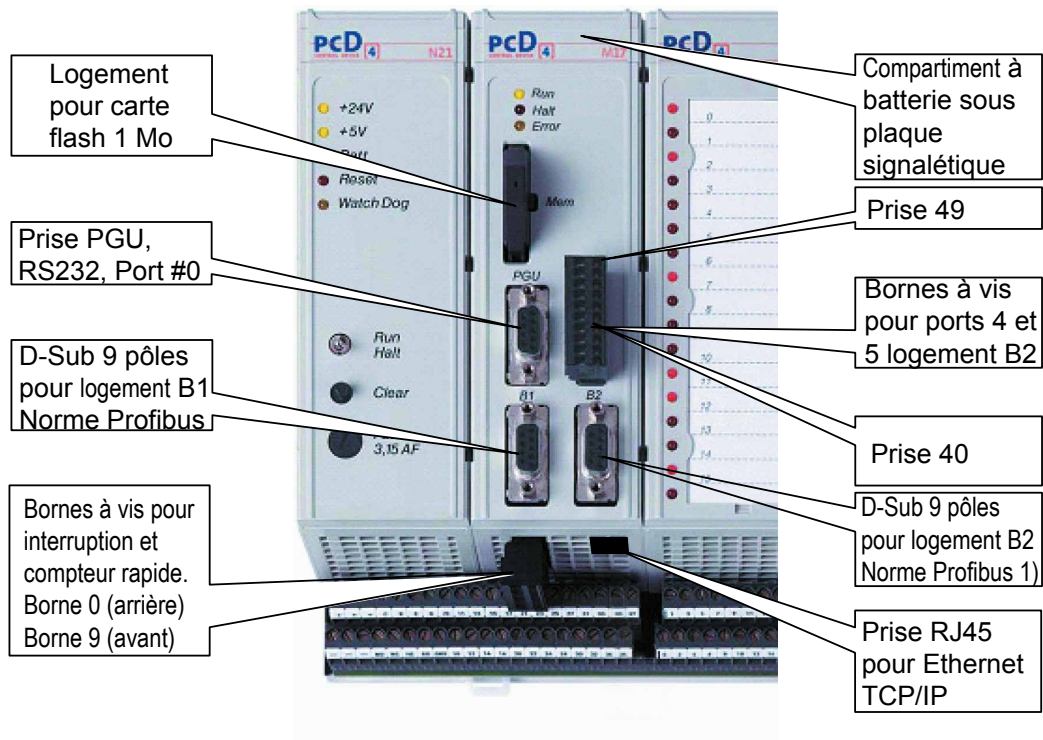
#### Pour commander :

- Système configuré :  
PCD2.M480F655-2    Système configuré PCD2 avec deux modules Ethernet
- Compléments :  
PCD7.F655            Module Ethernet pour PCD1/PCD2  
4'104'7503'0        Capot pour PCD2.M480F655-2 avec découpe pour deux prises RJ45

### 2.3.4 PCD4.M170Fx9

La connexion Ethernet s'effectue par la prise RJ 45 située à la base du PCD4.M17x. Seul le PCD4.M170Fx9 (où x est le code de l'interface raccordée à B1) est disponible comme système configuré.

PCD7.F65x sur logement B1 / canal 9



Le fil PGND du PCD7.F65x doit être raccordé à PGND. Les bornes Saia PCD® 40...049 doivent être réservées pour RS-232 (RS-232 en cours).

#### Pour commander :

- Système configuré :  
PCD4.M170Fx9      Système configuré PCD4 avec module Ethernet

### 2.3.5 PCD7.F65x sur xx7

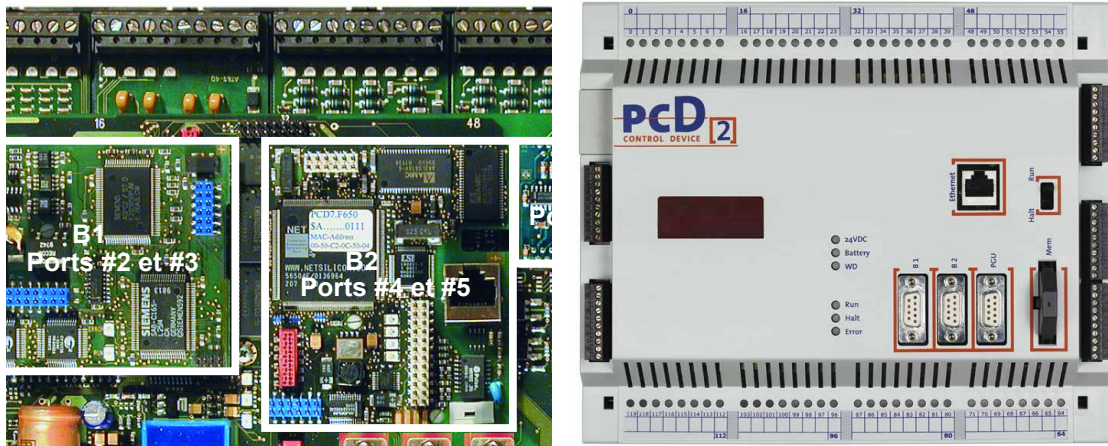
Il existe des systèmes configurés et non configurés pour xx7 compatible Ethernet avec PCD7.F65x. A cet effet, voir le manuel Ethernet xx7 26-791.



## 2.4 Systèmes compatibles Ethernet non configurés

### 2.4.1 PCD2.M170 avec PCD7.F65x

PCD7.F65x sur logement B2 / canal 8



Le fil PGND du PCD7.F65x doit être relié à la borne PGND à côté de B1 DB9. Les bornes Saia PCD® 40...049 doivent être réservées pour RS-232 (RS-232 en cours).

**Pour commander :**

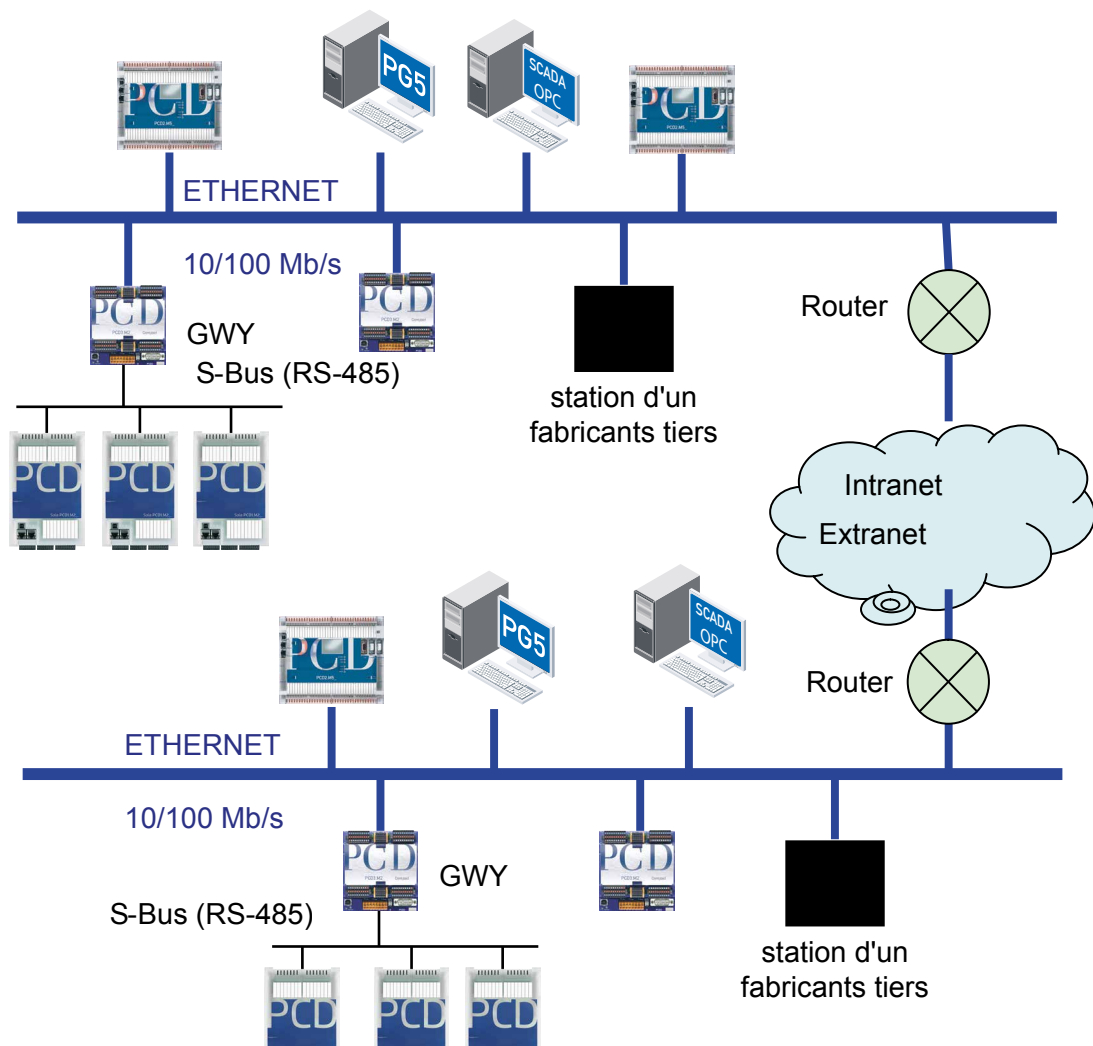
PCD7.F655                    Module Ethernet pour PCD1/PCD2

### 2.4.2 PCD7.F65x sur xx7

Il existe des systèmes configurés et non configurés pour xx7 compatible Ethernet avec PCD7.F65x. A cet effet, voir le manuel Ethernet xx7 26-791.

### 3 Caractéristiques et fonctions

#### 3.1 Connexions possibles et topologies de réseau



Ethernet permet normalement toutes sortes de connexions entre les stations reliées au réseau. L'outil de programmation Saia PG5® et un système SCADA peuvent accéder directement à un PCD et à une station d'un fabricant tiers via Ethernet. L'outil de programmation Saia PG5® et le système SCADA qui est basé sur SCOM.dll, sont des clients.

Les modules Ethernet TCP/IP de Saia Burgess Controls ne nécessitent pas de réseau propriétaire et peuvent être utilisés dans des réseaux standard avec des composants standard comme des concentrateurs, commutateurs, routeurs, etc. Les modules Ethernet TCP/IP de Saia Burgess Controls SA prennent en charge toutes les topologies de réseau actuelles.

Il est possible d'établir un réseau S-Bus et de le positionner sous une station passerelle reliée à Ethernet. L'accès au PCD dans ce sous-réseau S-Bus a lieu indirectement via un PCD qui a été configuré comme station passerelle Ethernet. Celui-ci envoie alors les messages reçus d'Ethernet au sous-réseau S-Bus. Plusieurs réseaux S-Bus peuvent ainsi facilement s'intégrer à Ethernet.

Dans un réseau Ethernet TCP/IP, il existe deux protocoles :

- Protocole S-Bus (UDP/IP, port 5050)
- Protocole Open Data-Mode (UDP/IP ou TCP/IP et ports définis par l'utilisateur), qui utilise une interface de communication (socket) pour l'application d'un programme utilisateur.

### Présentation des fonctions et de leur moyen d'accès

Fonctions	Accessibles avec		
	Ethernet S-Bus UDP/IP Port 5050	Open Data Mode Ethernet UDP/IP	Open Data Mode Ethernet TCP/IP
Programmation du PCD et dépiage des pannes/dépannage avec Saia PG5®	✓		
Communication S-Bus multi-maître PCD	✓		
Raccordement de stations PCD à un système SCADA	✓	✓	✓
Communication entre stations PCD et un système tiers			
Implémentation de protocoles utilisateurs		✓	✓
Fonctionnalité de passerelle S-Bus (d'Ethernet à S-Bus standard)	✓		

S-Bus UDP, Open Data Mode UDP et Open Data Mode TCP peuvent être utilisés simultanément avec le même PCD.

### 3.2 Ether-S-Bus

Le protocole Ether-S-Bus est utilisé pour la communication entre :

- deux PCD
- un PCD et l'outil de programmation Saia PG5®
- un PCD et d'autres stations (systèmes SCADA (GTB), serveurs OPC ou un autre PCD qui prend en charge le protocole S-Bus).

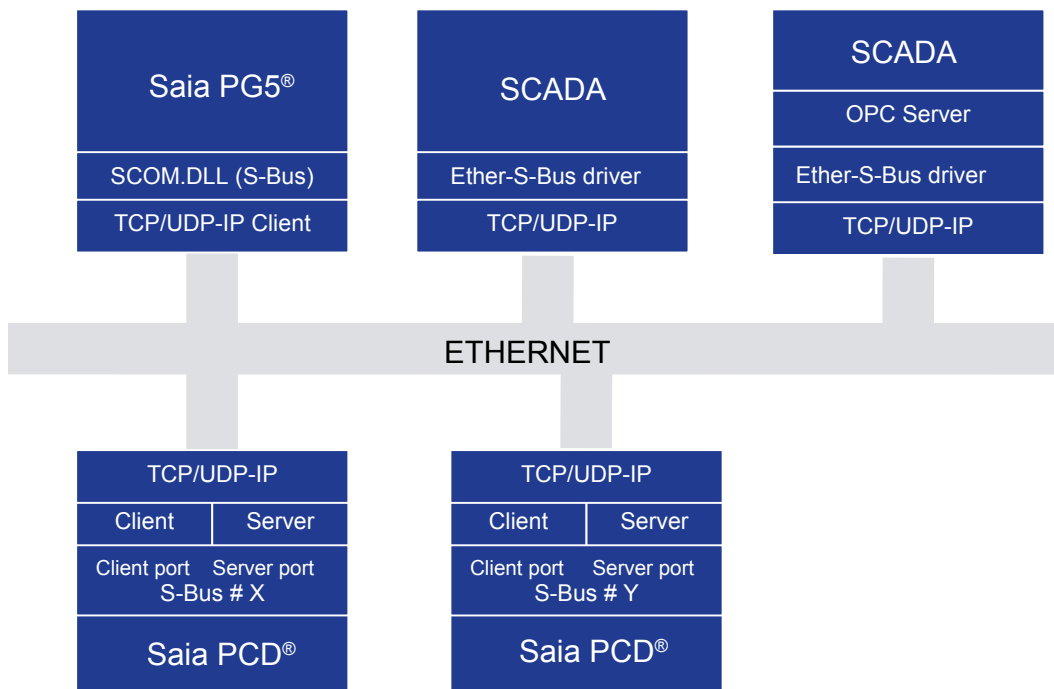
Les données sont échangées à l'aide des traditionnelles instructions STXM/SRXM en IL (AWL) ou des boîtes de fonctions FUPLA appropriées. La syntaxe est très similaire aux télégrammes S-Bus existants.

Le S-Bus via IP est implémenté en utilisant des sockets UDP par le biais du port d'entrée fixe 5050. Si la gestion de connexion se fait à travers un pare-feu, il faut s'assurer que ce port est activé dans la configuration du pare-feu.

La communication multi-maître entre PCD est prise en charge dans Ether-S-Bus. C'est pourquoi, chaque PCD dispose de ports serveur et client et peut travailler simultanément en tant que client ou rester passif en tant que station serveur.

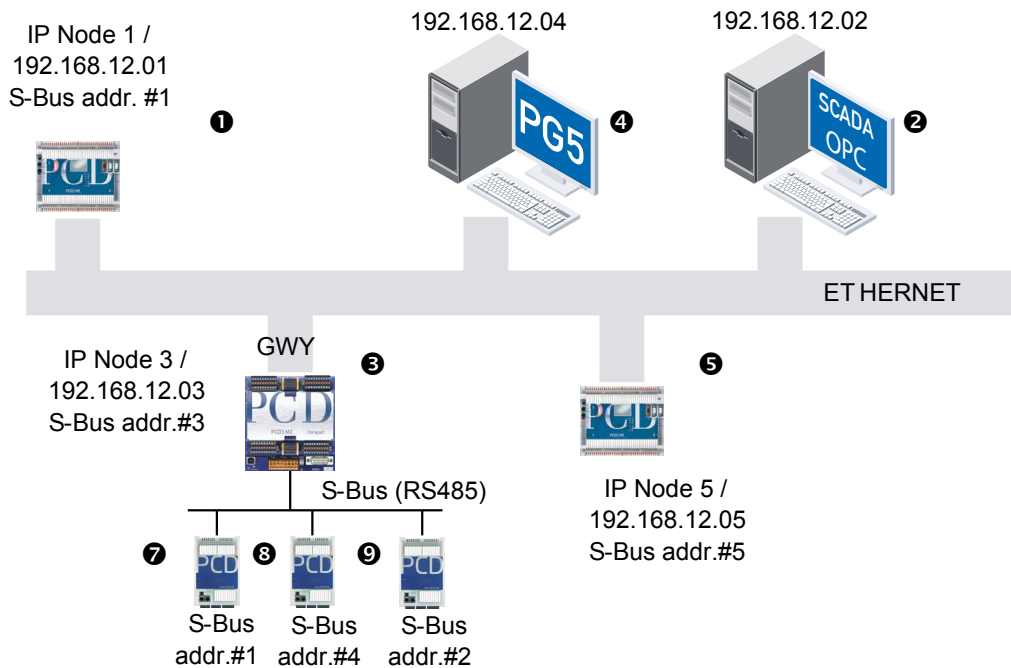
Représentation des couches d'application :

- Saia PG5® basé sur SCOM.dll
- Système SCADA (GTB) prenant en charge Ether-S-Bus
- Système SCADA (GTB) relié à un serveur OPC
- PCD avec ports serveur et client



Ether-S-Bus peut aussi bien être pris en charge par un système SCADA (GTB), que par le serveur OPC sur lequel se connecte un système SCADA.

### 3.2.1 Topologie et adressage du réseau



3

Chaque module IP dans Ethernet est identifié par une adresse MAC fixe et unique dans le monde entier. La station est configurée avec une adresse IP. Celle-ci dépend souvent de l'emplacement au sein de l'entreprise et n'est pas enregistrée sur Internet (adresse IP privée). Un nœud IP est défini selon les spécifications de SBC.

La table de correspondance (adresse MAC ↔ adresse IP) est gérée en interne par les piles TCP/IP- ARP. Pour faciliter au maximum la gestion des adresses pour l'utilisateur, une couche d'abstraction supplémentaire a été ajoutée au nœud IP. Le module IP est configuré dans le configurateur matériel Saia PG5® via l'affectation d'une adresse IP, d'un nœud IP simple et d'une adresse S-Bus. Ce même numéro est souvent utilisé pour l'adresse S-Bus et le nœud IP d'une station.

Ultérieurement, dans le programme, le numéro de nœud IP et l'adresse S-Bus de la station sont uniquement utilisés pour la communication sur Ethernet. Après la configuration d'une station, l'utilisateur ne doit plus s'occuper de l'adresse IP.

La table avec toutes les combinaisons d'adresses IP, de nœuds IP et d'adresses S-Bus pour l'ensemble du projet est générée par Saia PG5® et enregistrée dans le PCD.

Table des stations du point de vue du PCD constituant les nœuds IP #1 (adresse IP 192.168.12.01, adresse S-Bus #1).

Station	Adresse IP	Nœuds IP	Adresse S-Bus
5	192.168.12.05	5	5
3	192.168.12.03	3	3
7	192.168.12.03	3	1
8	192.168.12.03	3	4
9	192.168.12.03	3	2

Pour de plus amples détails sur l'utilisation de la table d'adresses, voir l'aide Saia PG5® : Paramétrages matériels - page TCP/IP. Avant d'appeler l'instruction STXM/SRXM, l'adresse de destination, tout comme chaque protocole S-Bus standard, est chargée dans le registre d'adresses. Comme illustré dans les exemples suivants, l'adressage a lieu dans deux champs d'adresse.

Station d'adressage ⑤	LDL R 100 5 ; adresse S-Bus LDH R 100 5 ; nœuds IP STXM 9 10 F 500 O 32
Station d'adressage ⑥	LDL R 100 3 ;Adresse S-Bus LDH R 100 3 ;Nœuds IP STXM 9 10 F 500 O 32
Station d'adressage ⑧	LDL R 100 4 ;Adresse S-Bus LDH R 100 3 ;Nœuds IP STXM 9 10 F 500 O 32

3



### 3.2.2 Programmation, dépistage des pannes et dépannage via Ethernet

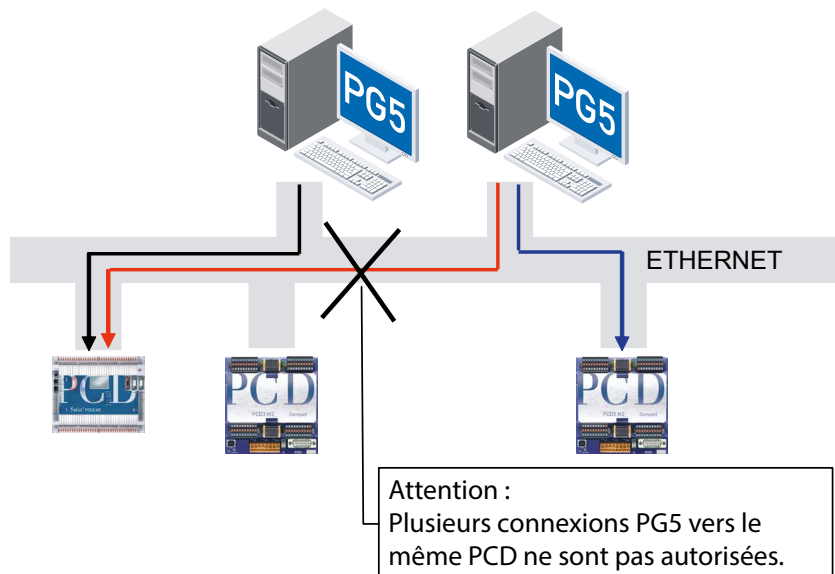
Il est possible de configurer, programmer, télécharger et déboguer une station PCD via Ethernet avec l'outil de programmation Saia PG5® et le protocole S-Bus. Les programmes peuvent être téléchargés plus rapidement via Ethernet qu'avec S-Bus via une interface série. Ceci est très utile pour les programmes utilisateurs globaux.

La programmation via Ethernet avec le protocole S-Bus est possible à partir de la version 1.1 du Saia PG5® en utilisant :

- la liste d'instructions (IL) ou
- les boîtes de fonctions FUPLA pratiques.

Les données sont échangées dans l'IL avec les instructions STXM/SRXM classiques. La syntaxe est très similaire au télégramme S-Bus standard. L'accès aux données est sécurisé au moyen de la protection par mot de passe S-Bus.

Il est possible d'utiliser plusieurs outils de programmation Saia PG5® sur un réseau Ethernet. Ceci permet de gagner du temps grâce au développement parallèle et à la mise en service de gros projets par plusieurs programmeurs ou équipes. En revanche, une seule liaison S-Bus UDP est possible en même temps sur les PCD.

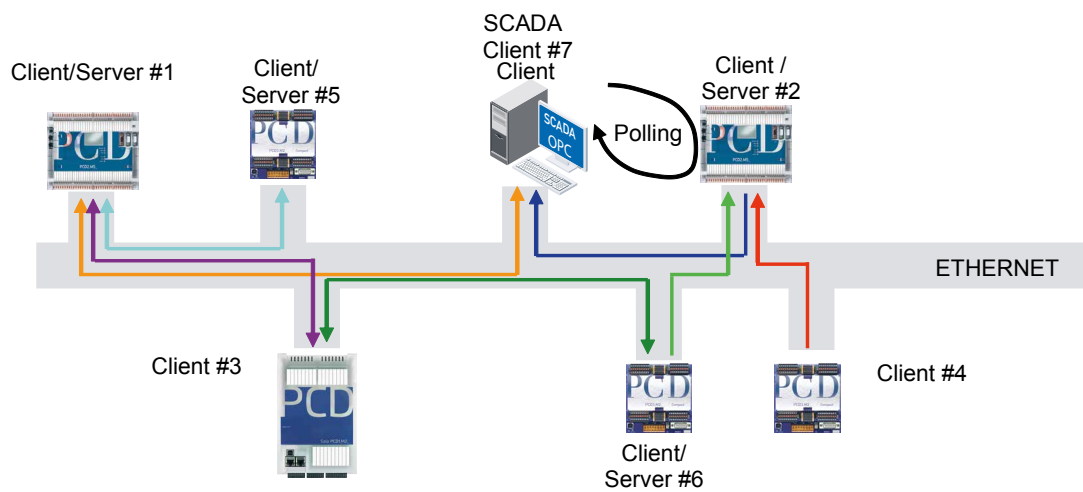


### 3.2.3 Communication multi-maître

La nouvelle fonction maître-maître pour Ethernet constitue une amélioration considérable par rapport au protocole S-Bus standard. Le protocole S-Bus par RS-485 standard ne permettait qu'un client par réseau. Avec le S-Bus via Ethernet, toutes les stations peuvent être clients et serveurs.

Grâce à la conception du réseau pour le fonctionnement multi-maître, il est possible d'établir des communications performantes et événementielles entre stations PCD. Un port à la fois client et serveur est ainsi implémenté dans chaque PCD.

Ainsi, un PCD (client/serveur #2) reçoit ainsi des informations de plusieurs autres PCD (client/serveur #6 et client #4) et est périodiquement consulté par un système SCADA ou un serveur OPC.





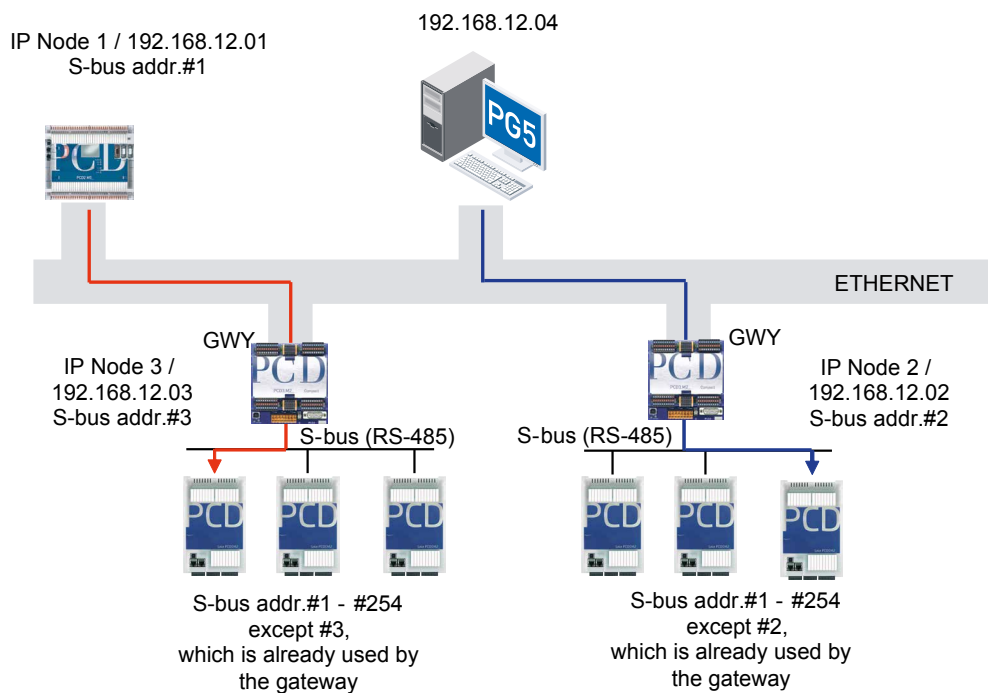
### 3.2.4 Passerelles et sous-réseaux S-Bus

Il est possible d'établir un sous-réseau S-Bus sous une station passerelle (GWY) reliée à Ethernet.

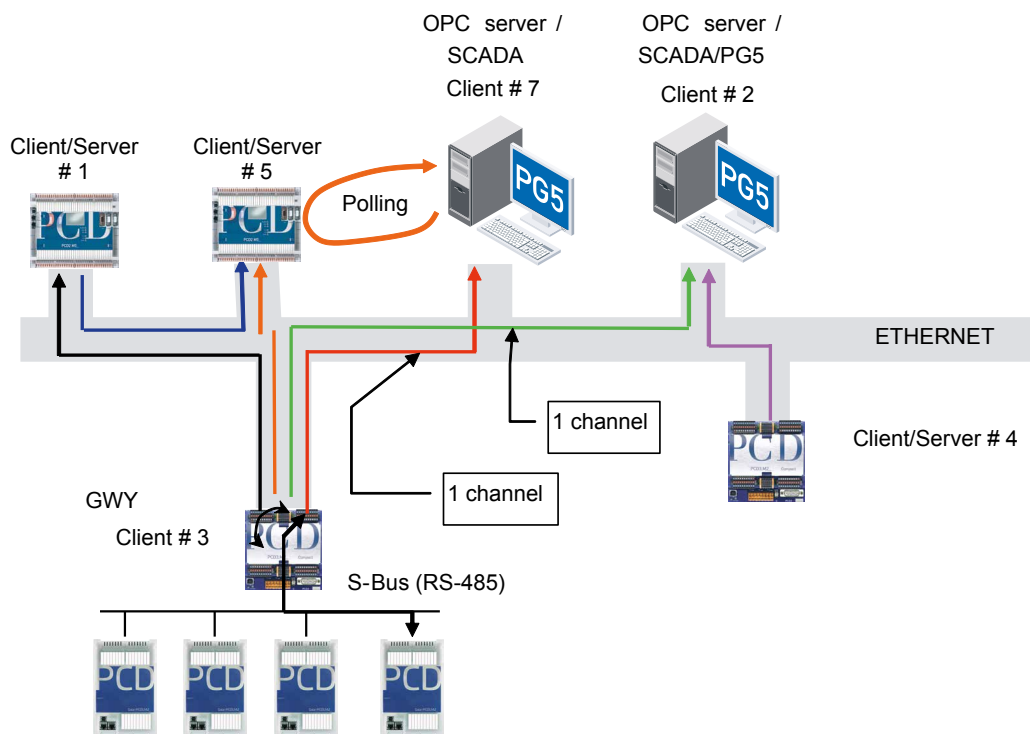
Tout PCD de ce sous-réseau S-Bus est indirectement accessible par le PCD configuré en passerelle Ethernet (GWY). Celui-ci envoie alors les messages reçus d'Ethernet au sous-réseau S-Bus. Cette station passerelle est l'unique station client autorisée dans le sous-réseau S-Bus.

Plusieurs réseaux S-Bus peuvent ainsi facilement s'intégrer à Ethernet. Il est, en outre, possible sur Ethernet, d'adresser 65 535 nœuds IP ou 65 535 x 254 PCD (254 PCD dans un sous-réseau S-Bus ainsi que le PCD configuré comme une station passerelle).

Un PCD dans un sous-réseau S-Bus nécessite une seule adresse S-Bus (aucune adresse IP).



### 3.2.5 Règles de communication d'une passerelle S-Bus



Dans une station passerelle, un nombre limité de tampons est réservé pour le traitement des télégrammes entrants.

PCD1 GWY : 1 tampons (= 1 télégramme ; max. 1 client)  
 PCD2 GWY : 4 tampons (= 4 télégrammes ; max. 4 clients)

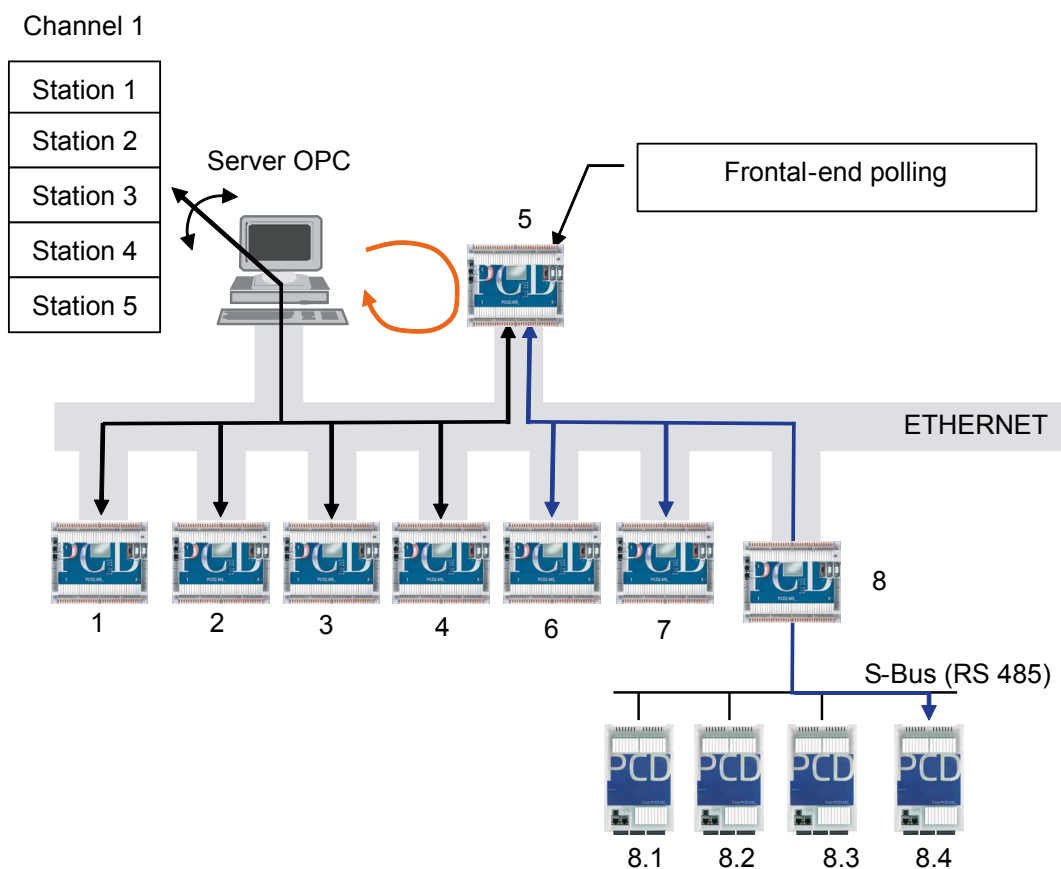
Les télégrammes sont traités de manière séquentielle.

Il est conseillé d'utiliser un seul canal par serveur OPC ou par système SCADA pour communiquer avec les stations S-Bus dans une station passerelle.

Si, pour un PCD passerelle PCD2, plus de quatre clients accèdent en même temps aux stations PCD reliées à la station passerelle, les télégrammes peuvent se perdre. Concernant le PCD1 comme passerelle, cette restriction s'applique déjà pour un client.

### 3.2.6 Règles de communication d'un serveur OPC

#### Solution standard avec un canal

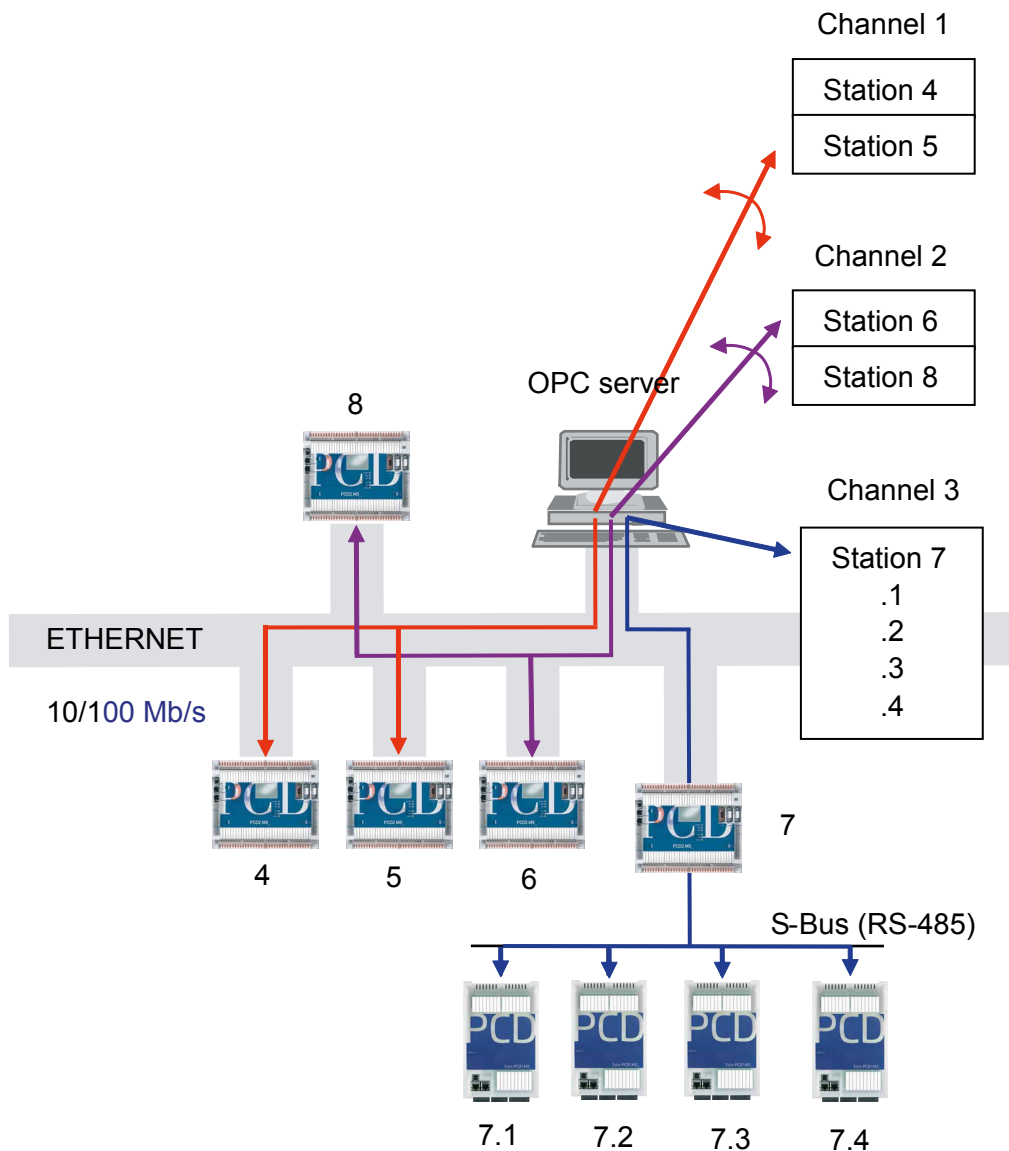


La solution standard consiste à traiter plusieurs stations PCD par le biais d'un seul canal sur le serveur OPC. Les stations PCD 1, 2, 3, 4 et 5 sont reliées en série les unes après les autres.

Il est également possible de consulter un « PCD frontal » qui reçoit les données de plusieurs autres stations PCD, ex. : 6, 7 et 8.4.

Solution avec plusieurs canaux

3



Si la solution standard n'est pas possible car les données des stations PCD doivent être traitées plus rapidement, plusieurs canaux parallèles peuvent être orientés sur le serveur OPC. Les données des stations PCD peuvent ainsi être traitées plus rapidement.



Chaque canal correspond à une tâche de programme sur le serveur OPC. Plus le nombre de tâches est élevé, plus la charge de l'unité centrale est élevée.

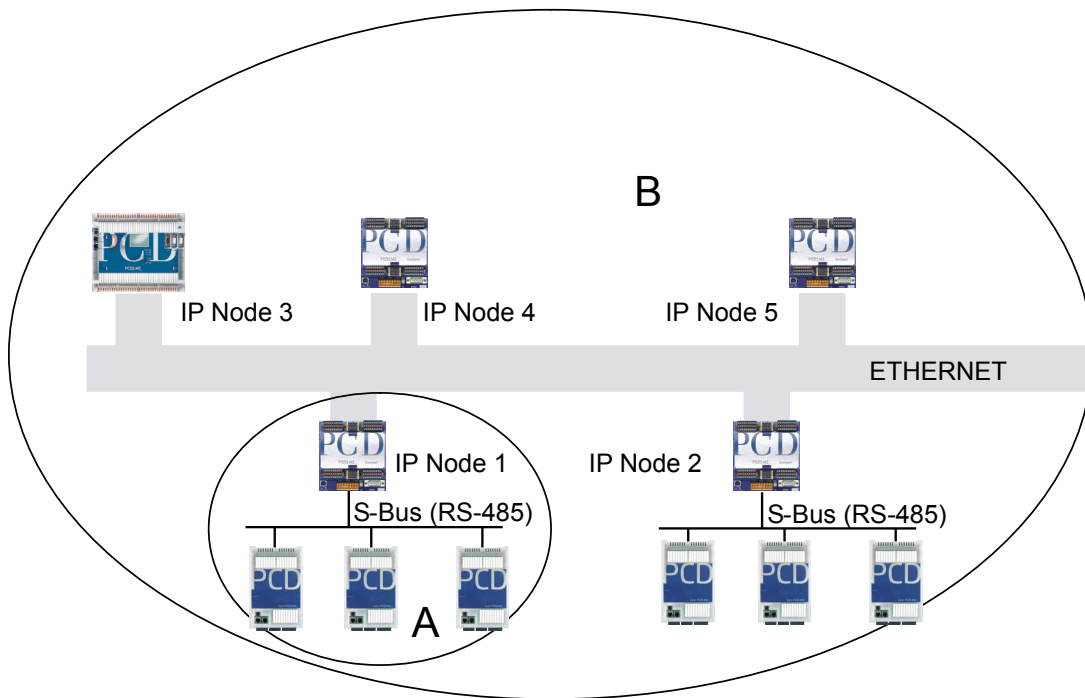
Les stations PCD1 dans un réseau S-Bus dans une station passerelle doivent uniquement être reliées par un canal avec le serveur OPC car il existe un seul tampon de télégramme.

### 3.2.7 Télégrammes de diffusion

Il est possible de transmettre de deux manières différentes les télégrammes de diffusion pour la synchronisation avec une station client.

- A Diffusion limitée au réseau S-Bus
- B Diffusion sur tout le réseau Ethernet, y compris toutes les stations S-Bus qui sont adressées dans une station passerelle.

3



Le nœud IP 65 535 (0xFFFF) est réservé à la transmission de télégrammes de diffusion via IP. Il peut être adressé dans l'IL et avec les boîtes de fonctions FUPLA.

<b>A</b>	Diffusion limitée à un réseau S-Bus	Envoi au nœud IP	X<65'535
		S-Bus	255
<b>B</b>	Diffusion sur tout le réseau Ethernet, y compris le réseau S-Bus dans la station passerelle	Envoi au nœud IP	65'535
		S-Bus	255
<b>C</b>	Ce type de diffusion n'est pas autorisé	Envoi au nœud IP	65'535
		S-Bus	X<255

### Côté client :

Si des télégrammes de diffusion de type C sont envoyés, le diagnostic de la station client émet des messages d'erreur. Le télégramme n'est pas envoyé par la station client.

- L'indicateur NEXE est placé
- Les indicateurs 29 et 30 dans le diagnostic d'envoi du registre de diagnostic sont placés
- La DEL d'erreur s'allume

3

En outre, le Saia PG5® n'autorise pas l'envoi de télégrammes de diffusion de type C.

### Côté serveur :

Côté serveur, les télégrammes de diffusion de types A et B sont exécutés sans réponse.

Le serveur ne fait pas la différence entre les télégrammes S-Bus standard et les télégrammes de diffusion de type C. La station serveur PCD répond à ce type de télégrammes reçus.

Voir exemple suivant pour l'envoi d'un télégramme de diffusion (envoi binaire) de type B.

The screenshot displays the SBC Fupla Editor interface. The main workspace shows a ladder logic diagram with several rungs. A 'Channel S-Bus Master IP' block is connected to output 2. A 'Blink' block is connected to output 0. A 'ref.Channel SEND' block is also visible. An 'Adjust: Transmit Binary' dialog box is open, showing the following configuration:

Parameter	Value
Initialization	No
Node	65535
Destination station	255
Destination element	Output
Destination address	0
Manual transmit	

Callouts in the image point to the 'Node' field (65535) and the 'Destination station' field (255), with labels: 'Définit diffusion via IP' and 'Définit diffusion sur S-Bus' respectively.

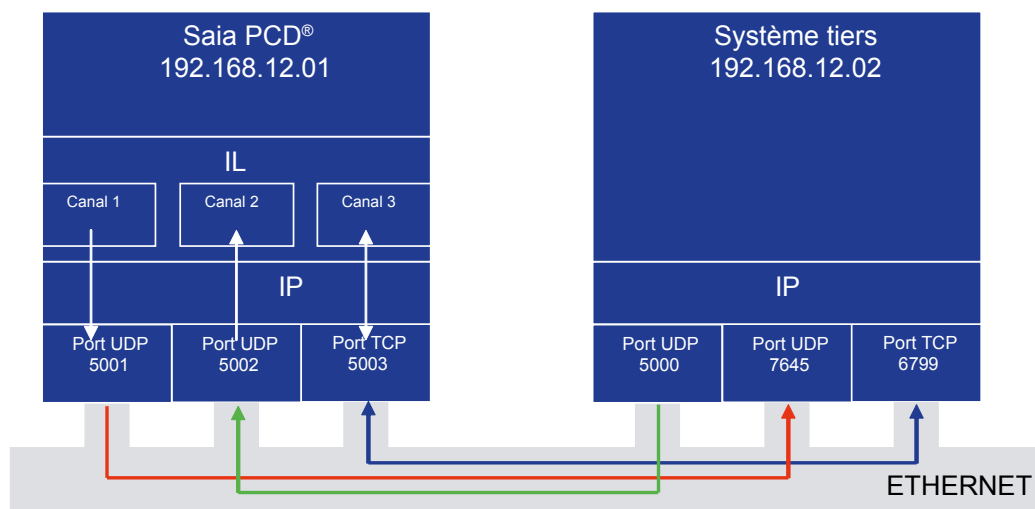
### 3.3 Open Data Mode via TCP/IP ou UDP/IP

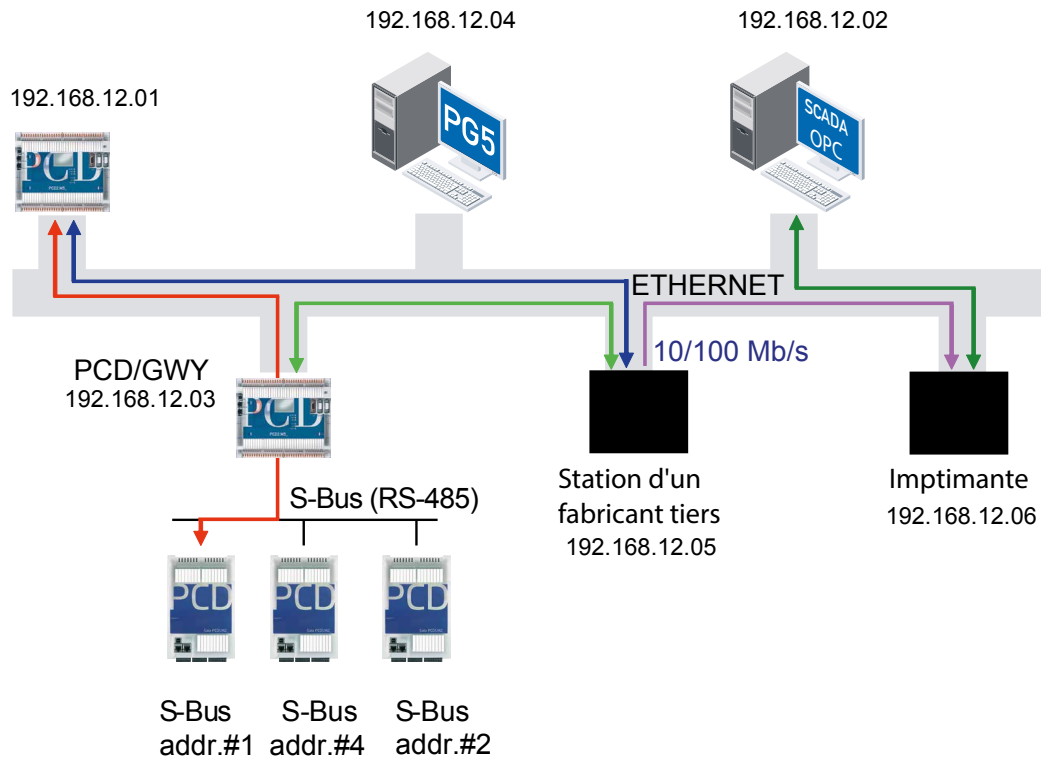
L'utilisation du mode de données ouvert (Open Data Mode) permet à un PCD de communiquer avec une station d'un fabricant tiers, qui ne prend pas en charge le protocole S-Bus. Deux PCD (si nécessaire) peuvent également communiquer entre eux en Open Data Mode.

Les stations de fabricants tiers (ex. : imprimante, autres API, etc.) ne prennent pas en charge le protocole S-Bus ; seules les données brutes (caractères, chaînes) peuvent être échangées. Un PCD peut envoyer les paquets de données TCP/IP ou UDP/IP bruts et transparents. L'implémentation du protocole est alors traitée par l'application de l'utilisateur sans pratiquement aucune réserve ni limitation. Les règles d'implémentation de l'application sont laissées entièrement à la discrétion de l'utilisateur. Parmi les possibilités, citons la communication multi-maître, la communication événementielle, etc.

L'utilisateur peut choisir entre la communication TCP/IP ou UDP/IP. L'UDP fonctionne en mode non connecté et le TCP en mode connecté. Dans TCP/IP, on fait la différence entre client et serveur lors de l'établissement d'une communication.

Dans ce type de protocole, le programmeur utilise la socket API SBC pour l'accès direct à la couche UDP ou TCP/IP. Un canal de communication sur le PCD est défini avec l'adresse IP de l'appareil et le port de communication (port UDP / port TCP).







### 3.4 Conception et structure du réseau

Outre le module Ethernet TCP/IP PCD7.F65x, d'autres composants sont nécessaires à la réalisation des réseaux Ethernet.

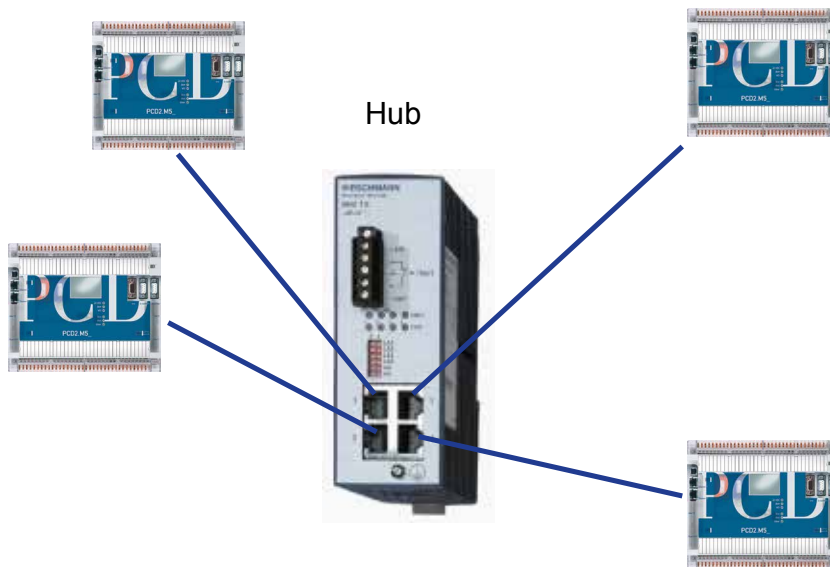
#### 3.4.1 Concentrateurs (réseau en étoile)

Le schéma d'un réseau de concentrateur est semblable à une étoile - un concentrateur central avec des câbles qui conduisent à chaque hôte raccordé. Un concentrateur transmet seulement le signal qu'il reçoit sur le câble. Il n'existe aucune liaison logique. Contrairement à un réseau en bus ou en anneau, un réseau avec la topologie en étoile réduit la vulnérabilité de l'ensemble du réseau à chaque rupture d'un câble. Si le câble de raccordement reliant un hôte au réseau est endommagé ou déconnecté, seul l'accès à cet hôte est perdu.

3

Arguments en faveur de l'utilisation de concentrateurs :

- le faible coût de connexion des unités
- les faibles retards lors de la transmission de données
- un concentrateur permet de lire l'ensemble de l'ensemble de données avec l'analyseur Ethernet. Contrairement au commutateur qui canalise la communication.



### 3.4.2 Les commutateurs

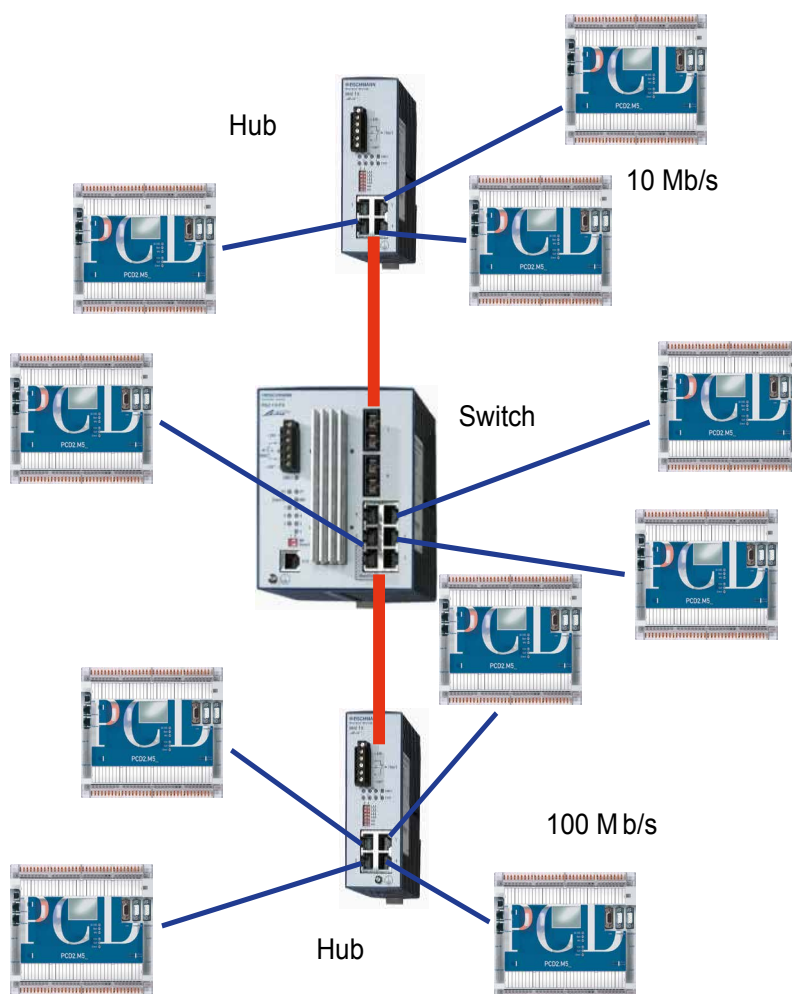
Les commutateurs ne se bornent pas à transmettre ou répéter les signaux envoyés sur un câble, ils reçoivent tous les signaux, les traitent et les retransmettent. Résultat : seules les trames Ethernet valides sont envoyées à un autre réseau. Ce filtrage s'effectue sur la couche Ethernet par rapport à l'adresse MAC. La décision de routage est prise une seule fois pour chaque adresse IP, puis chaque trame avec la même adresse IP est retransmise à partir de cette décision.

L'échange de données avec une destination locale reste local. Ceci permet de réduire le nombre de collisions et d'optimiser la performance du réseau. On obtient alors des zones sans collision.

Deux réseaux Ethernet peuvent être reliés avec des commutateurs qui transfèrent l'échange de données d'un réseau à l'autre. Un commutateur peut traiter simultanément plusieurs télégrammes.

Arguments en faveur de l'utilisation de commutateurs :

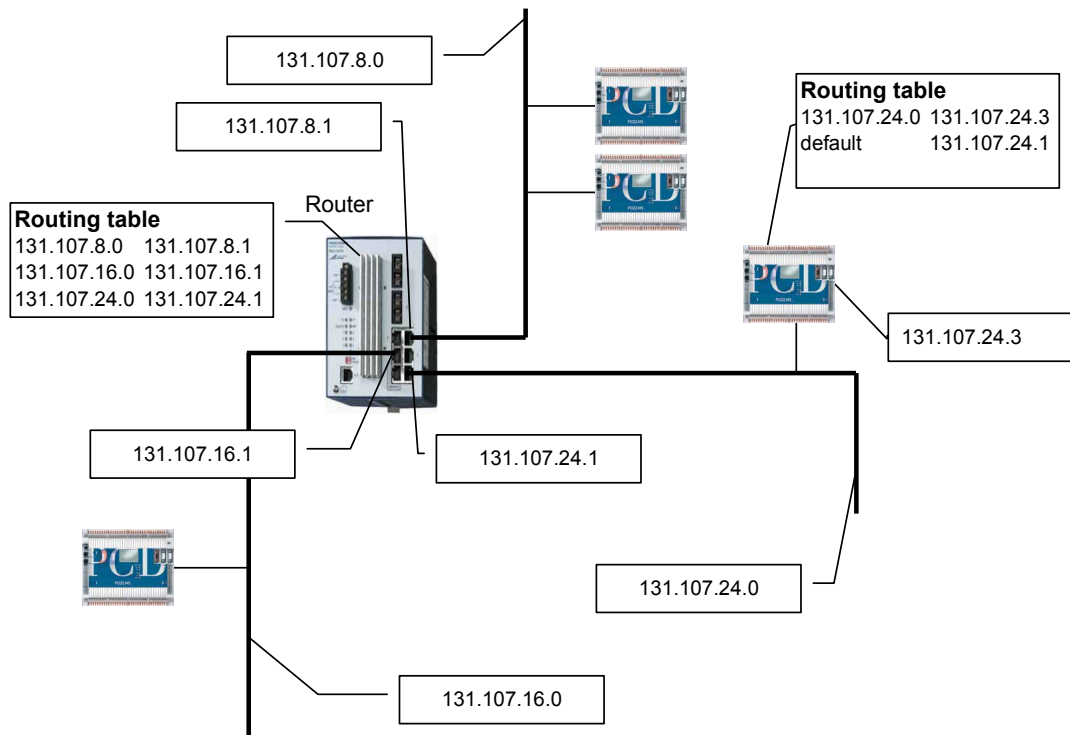
- la création de zones sans collision : l'amélioration du comportement déterministe du réseau en raison du nombre faible de collisions
- la combinaison de différentes connexions (10/100 Mbit/s, semi-duplex HDX / duplex intégral FDX)
- le flux de données plus efficace en raison de transmissions point à point et du mode duplex intégral
- l'amélioration des performances du réseau grâce à des filtres (filtres Broadcast et Multicast) et un mécanisme de gestion des priorités.



### 3.4.3 Routeur

Un routeur est un périphérique de connexion qui transmet les télégrammes d'un réseau physique à un autre. L'hôte émetteur et le routeur doivent sélectionner une destination de transmission pour le télégramme. Le choix de ce routage a lieu lorsque la pile IP consulte la table de routage. La table de routage comporte les adresses IP et identifie les interfaces réseau pour le routeur. Par défaut, un routeur envoie uniquement des télégrammes aux réseaux pour lesquels il dispose d'une interface configurée.

- Si un hôte souhaite communiquer avec un autre hôte, le protocole IP décide d'abord si l'hôte de destination se trouve dans le réseau local ou distant
- Si l'hôte de destination se trouve dans un réseau distant, le protocole IP recherche dans sa table de routage un routage jusqu'au réseau distant
- Si aucun routage explicite n'est trouvé, l'adresse par défaut du routeur est utilisée
- La table de routage sur le routeur est consultée, pour transmettre le télégramme au réseau spécifié.



### 3.4.4 Composants de réseau

Saia Burgess Controls ne fournit aucun composant de réseau Ethernet spécifique. La société possède une solide expérience dans les composants réseau de la Ethernet Rail Family de Hirschmann.

L'Industrial Line ETHERNET Rail Family a été spécialement développée pour les applications d'automatisation industrielle. Elle prend en charge les fonctions de redondance (chemin de données alternatif via les structures en anneau HIPER, couplage redondant de segments de réseau) et garantit une grande disponibilité du réseau.

Hirschmann assiste également ses clients dans leurs projets de conception de réseaux : [www.hirschmann.com](http://www.hirschmann.com)

### 3.4.5 Amélioration des performances

#### Duplex intégral, auto-négociation, auto-détection

Le module Ethernet TCP/IP PCD7.F65x prend en charge les modes de fonctionnement « duplex intégral » et « semi-duplex ». Duplex n'est pas une topologie de réseau, mais une méthode pour l'échange de données entre deux nœuds.

Le module Ethernet TCP/IP du PCD7.F65x utilise les fonctions d'auto-négociation et d'auto-détection pour installer un mode compatible entre deux nœuds. Dès que deux nœuds sont reliés ensemble, le mode de communication de données privilégié est installé d'après la liste suivante :

- Mode 100Base TX et duplex intégral
- Mode 100Base TX et semi-duplex intégral
- Mode 10Base T et duplex intégral
- Mode 10Base T et semi-duplex intégral

Si aucune auto-négociation n'est possible entre le PCD et la station partenaire, le mode 10Base T et semi-duplex est installé.

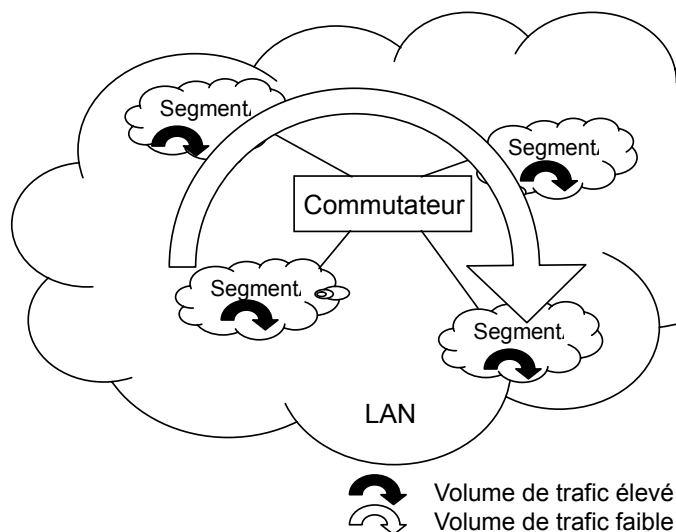
L'auto-détection est la fonction qui détecte le débit de données d'un signal (10 Mbit/s ou 100 Mbit/s). L'auto-détection est également possible sans auto-négociation.

#### Réseau LAN commuté / réseau LAN partagé

Il est également possible de diviser le réseau en différents groupes pour atteindre un réseau Ethernet isolé, sans surcharger inutilement le trafic global.

La technologie avec des commutateurs standard doit ainsi être utilisée pour créer des zones sans collision.

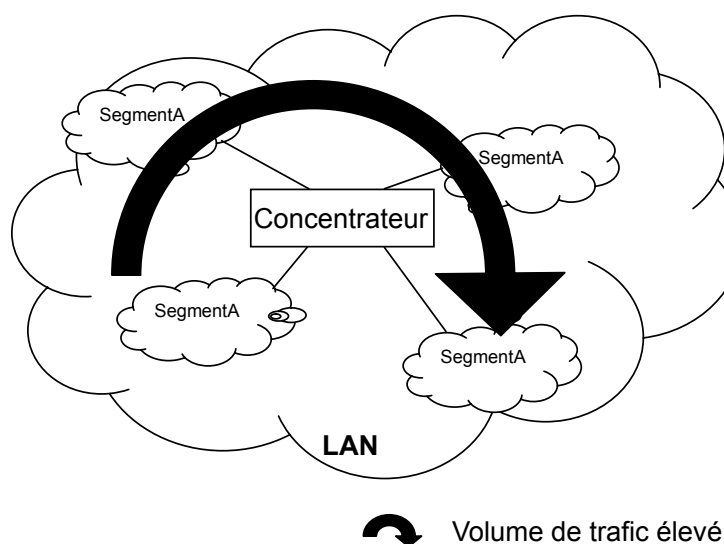
### Réseau LAN commuté



- Chaque segment dispose de la capacité totale / du débit en bauds total
- Trafic de données simultané dans plusieurs segments
- Le trafic de données local reste local. Seuls les télégrammes sélectionnés quittent le segment local

3

### Réseau LAN partagé



- Tous les segments utilisent la capacité / le débit en bauds en commun
- Tous les télégrammes traversent tous les segments
- Un seul télégramme se trouve toujours dans le réseau
- Les collisions réduisent la performance à 40 %

### Maximisation de la charge utile (payload)

Il faut noter que TCP/IP et UDP/IP (mais notamment TCP/IP) présentent une surcharge de trafic assez importante. Cette surcharge de trafic devient considérable lorsque la charge utile d'un télégramme est de petite taille. Il est donc recommandé de grouper et de transmettre de grosses charges utiles (ex. : 32 registres sur S-Bus via UDP/IP), pour exploiter la puissance d'Ethernet.

## 4 Configuration et programmation

### 4.1 Configuration et programmation

#### 4.1.1 Configuration du port IP S-Bus (serveur)

Le module IP est configuré dans Saia PG5® → Hardware Settings (Device Configurator)

The screenshot shows the SBC Device Configurator interface. The main window displays various hardware settings for a device. On the right side, the 'Properties' panel is expanded to show 'Onboard : Ethernet' settings. The 'TCP/IP' section is active, showing the following configuration:

- Channel Number: 9
- TCP/IP Enabled: Yes
- Ethernet RIO Netw: None
- IP Address: **172.23.16.123**
- Subnet Mask: **255.255.252.0**
- Default Router: **172.23.16.252**

The 'DHCP Client Protocol' section is also visible, with 'DHCP Client Enable' set to 'No'. The 'Ether-S-Bus' section shows 'IP Node' set to '123' and 'PGU Port' set to 'Yes'. The 'Ether-S-Bus Master Gateway' section shows 'Channel Number G' set to '9' and 'Use Ether-S-Bus Fc' set to 'No'.

The main window also displays the following tables:

Type	Description
PCD3.M5540	CPU with 256/512/1024 KBytes RAM, 4 I/O slots (expandable), USB.

Slot	Type	Description
M1		
M2		

Type	Description
RS-485/S-Net	RS-485 port for Profi-S-Bus or general-purpose communications (D-Sub #1).
USB	Universal Serial Bus port, PGU or general-purpose.
RS-232/PGU	RS-232, PGU or general-purpose serial port (D-Sub #1).
RS-485	RS-485 port for general-purpose communications (Terminal block).
Ethernet	Ethernet port.

Section	Description
IP Transfer Protocols	FTP, HTTP Direct Protocols, ODM.
IP Protocols	DNS, SNMP, SNMP protocols.
HTTP Portal	HTTP Portal Communication For PCD Over Private Network

Slot	Type	Description
Slot 0	PCD3.S100	Workshop Simulator Unit with 8 digital inputs, 8 digital outputs
Slot 1	<reserved>	

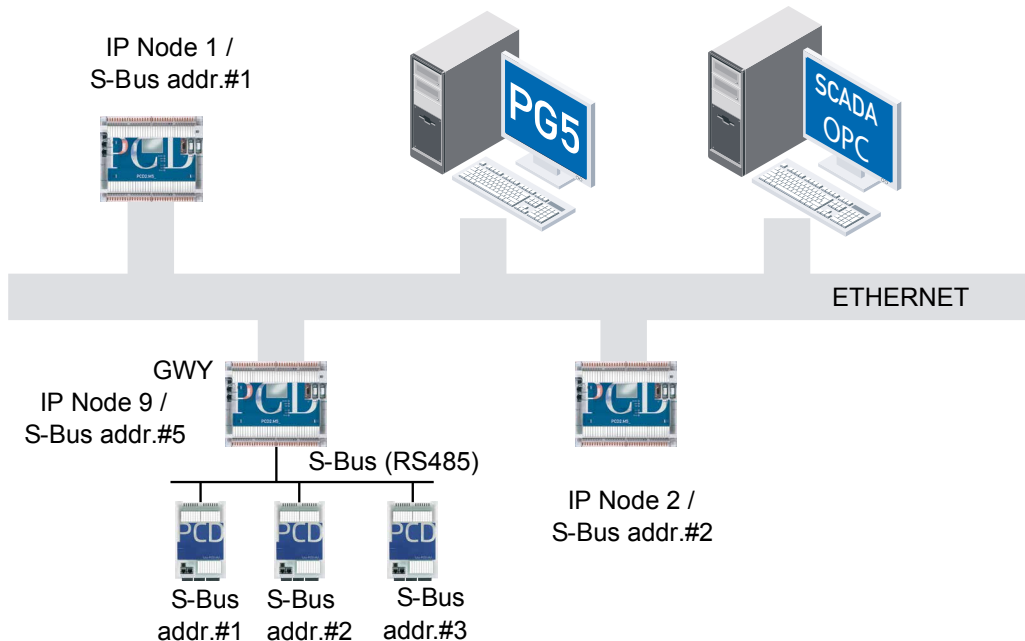
### 4.1.2 Adressage de la station serveur IP

En principe, chaque module IP est défini par une adresse Ethernet unique au monde (MAC) et une adresse IP liée à l'emplacement. La correspondance (adresse MAC ↔ adresse IP) est assurée par le protocole ARP (Address Resolution Protocol) dans la pile TCP/IP.

Le PG5 génère une table de correspondance (DBX) pour associer les adresses IP, nœuds IP et adresses S-Bus à toutes les données matérielles des unités centrales du projet. Pour la programmation, l'utilisateur a uniquement besoin des adresses S-Bus et nœuds IP. Le DBX est calculé pendant la mise au point du projet et téléchargé dans le PCD à l'aide du programme utilisateur. Si une adresse IP, une adresse S-Bus ou un nœud IP est modifié dans le projet, le projet doit être à nouveau généré (génération d'un nouveau DBx) et téléchargé dans le PCD. Pour de plus amples informations, voir l'aide PG5.

Un adressage codé Word pour les nœuds IP et une adresse S-Bus codée en octets offrent de bonnes capacités d'adressage appropriées dans le réseau Ethernet. Ils permettent l'adressage de 65 536 nœuds IP ou de 65 536 x 254 PCD (254 PCD dans un réseau S-Bus via une passerelle S-Bus Ethernet, voir illustration). Le PG5 génère un DBx mettant en relation nœud IP et adresse IP.

Remarque : dans les anciennes versions, le nœud IP est codé en octets (PG5 V\$1.1.007, PCD1.M130 \$76, PCD2.M150 \$B6, PCDx.M170 \$0K, PCD7.F650 \$05 inclus). Les versions plus récentes prennent en charge le format Word pour le nœud IP. Dans le micrologiciel du PCD, l'ancien format de DBX est encore pris en charge.



Pour les stations dans un réseau S-Bus via la passerelle dans le nœud IP #9, l'adresse S-Bus #5 ne doit plus être utilisée, car elle est déjà utilisée par la passerelle.

Table de correspondance Adresse IP - Nœud IP - Adresse S-Bus

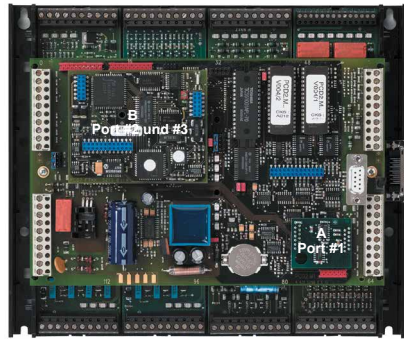
Adresse IP ↔...	...Nœud IP	...Adresse S-Bus
192.168.2.151	2	2
192.168.2.155	3	9
192.168.2.168	6	--
192.168.2.201	7	--
192.168.2.105	9	1...254 sauf 5

### 4.1.3 Logements et numéros de canal

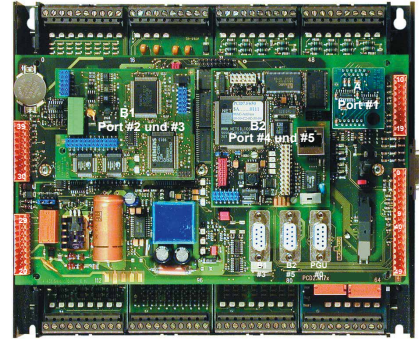
Les logements/canaux pour le PCD7.F65x sont définis comme suit sur le Saia PCD® (voir aussi chapitre 2) :



PCD1.M130  
Canal 9  
Logement B

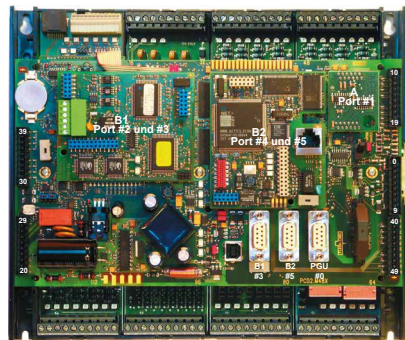


PCD2.M15x  
Canal 9  
Logement B

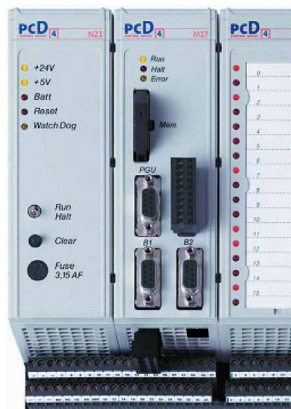


PCD2.M17x  
Canal 8  
Logement B2

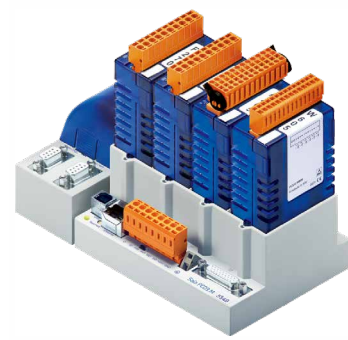
4



PCD1.M480  
Canal 9  
Slot B1  
et également  
Logement 8



PCD4.M17x  
Canal 9  
Logement B1



SansPCD7.F65x  
Compatible Ethernet  
PCD3.M5540  
PCD3.M3120  
PCD3.M3330  
PCD3.M6340  
PCD3.M6540  
Canal 9



## 4.2 Programmation de S-Bus via Ethernet

Le protocole S-Bus Ethernet permet la communication entre deux PCD, un PCD et un outil de programmation (PGX) ou une unité tierce (PLS avec scom.dll ou API compatible S-Bus).

La nouvelle fonctionnalité maître-maître dans un réseau est une amélioration du protocole S-Bus standard. Le protocole S-Bus standard ne permet qu'un maître par réseau. Dans S-Bus via Ethernet, toutes les stations peuvent servir de maîtres. Ce type de communication fait appel aux instructions STXM/SRXM traditionnelles. La syntaxe est similaire aux télégrammes S-Bus standard.

4

Caractéristiques	
Mécanisme	Demande d'instruction Instruction (client) • Affichage (serveur) • Réponse (serveur) • Confirmation (client)
Communication sécurisée	3 tentatives maxi. en arrière-plan. Le contrôle d'erreur CRC 16 de S-Bus est appliqué. Aucune couche de sécurisation particulière n'est utilisée via IP.
Protocole via IP	Le protocole UDP est utilisé pour la communication sur S-Bus via Ethernet. La socket de réception est ouverte et reliée en permanence au port 5050. La socket de sortie peut être reliée au port 5050 (par défaut) ou à un port libre dans le système

### 4.2.1 Description des instructions du Saia PCD®

Les instructions suivantes sont prises en charge avec IP :

SASI(I)	Affectation de l'interface série
STXM(X)	Emission de données série
SRXM(X)	Réception de données série

#### Initialisation de SASI

Outre la configuration dans le configurateur matériel hors ligne, une interface SASI est mise à disposition pour le client et l'esclave. Par souci de compatibilité ascendante, le nombre de paramètres SASI est inchangé. Comme S-Bus via Ethernet utilise un port d'entrée fixe et prédéfini (5050), aucun port de communication spécial ne doit être défini. Un port libre du système entre 1024 et 4999 est sélectionné comme porte de sortie pour la communication. Le port de sortie peut, également au format texte SASI, comme décrit ci-après, être relié au port 5050.

Application :           SASI   Channel           ; logement (8 ou 9)  
                                  Texte n°           ;texte de définition n° 0...7999

Exemple :                SASI   8                   ;initialisation du logement 8  
                                  Texte IP               ;texte de définition pour SASI

Définition du texte SASI :

PCD	« <mode_def>,<dest_reg>;   <diag_def>;           <option_def>»
Exemple	« MODE:EM,R100;           DIAG:F1000,R1000;   TOUT:500,PORT:5050
PCD	« <mode_def>,<dest_reg>;   <diag_def>;
Exemple	« MODE:ES;                DIAG:F1000,R1000;

« <mode\_def>,<dest\_reg>; Définit le mode

En mode maître Ethernet, l'adresse de la station distante est ajoutée à EM.

PCD	« MODE:EM,<dest_reg>; »	EM : mode maître Ethernet. <dest_reg> : ce registre définit l'adresse de la station distante.
PCD	« MODE:ES;	ES : mode esclave Ethernet (uniquement utilisé pour le diagnostic esclave). Remarque : l'esclave Ethernet est automatiquement configuré, lorsque le module TCP/IP est configuré dans les « Hardware Settings » (Paramétrages matériels) du PG5.

<dest\_reg> : Représente le registre qui définit la station partenaire. (type : Rxxxx)

Avant de pouvoir envoyer des instructions STXM/SRXM, l'adresse de destination doit être enregistrée dans un registre d'adresses comme pour le S-Bus standard. Cet adressage utilise deux champs d'adresse.

Champ d'adresse du registre (32 bits)		
Mot de point fort		Mot de point faible
Numéro de nœud IP du PCD de destination	Non utilisé	Adresse S-Bus de l'esclave de destination

<diag\_def> ; Définit les éléments de diagnostic pour la communication Ethernet.  
Format: « DIAG:<diag\_flag>,<diag\_reg>; »

Type		Description
<diag_flag>	Fxxxx Oxxxx	Adresse de base de 8 indicateurs ou sorties consécutives
<diag_flag>	Rxxxx	Adresse de base du registre de diagnostic

<option\_def>“ Définit les éléments d'option pour la communication Ethernet.  
Format: « [<tout\_def>],[<port\_def>; »

Type		Description
<tout_def>	TOUT : xxx	Définit la valeur de temporisation du mode EM en ms. La valeur de temporisation par défaut est de 500 ms. (valeur limite inférieure = 100 ms)
<port_def>	PORT : xxx	Sélection du port de sortie pour la communication UDP S-Bus Par défaut : xxx = 0 0 = un port libre dans le système entre 1024 et 4999 est sélectionné xxx = 5 000 à 65 535, indique une affectation fixe de la part de l'utilisateur

**Indicateurs :** Lorsque le texte de définition est manquant ou non valide ou lorsque le microprogramme n'offre pas de prise en charge IP, l'indicateur d'erreur (E) est placé.

## Diagnostic

Indicateurs de diagnostic

Adresse	Nom	Description
xxx	RBSY	Récepteur occupé
xxxx+1	RFUL	Tampon de réception plein
xxxx+2	RDIA	Diagnostic de réception
xxxx+3	TBSY	Emetteur occupé
xxxx+4	TFUL	Emetteur plein
xxxx+5	TDIA	Diagnostic d'émission

4

**Emetteur occupé (TBSY) ↑** Indique qu'une transmission est en cours.  
 Signification Station maître : Sur « H » (high) pendant l'exécution d'une instruction STXM ou SRXM L'indicateur est réinitialisé dès qu'une réponse valide a été reçue.  
 Station esclave : Sur « H » lorsque la réponse est envoyée.

**Diagnostic d'émission (TDIA) ↑** Indique qu'une erreur a été détectée pendant la transmission d'un télégramme. Une description détaillée de l'erreur figure dans le registre de diagnostic (bits 16 à 31). L'indicateur est réinitialisé dès que tous les bits de diagnostic d'émission (16 à 31) ont été réinitialisés dans le registre de diagnostic.

**Non-exécution (NEXE) ↑** Signale l'échec d'une instruction (STXM ou SRXM) après trois tentatives imparties. L'indicateur est réinitialisé avec l'instruction S-Bus suivante.

**Liaison physique (XBSY) ↑** Indique si une liaison physique est disponible ou non. Sur « H » (high) lorsque le module Ethernet TCP/IP détecte une liaison physique et sur « L » (low) lorsque la liaison physique est perdue. Dès que l'instruction SASI (maître ou esclave) est exécutée, cet indicateur est assigné et peut être utilisé comme diagnostic. Si le PCD envoie des télégrammes sans liaison physique établie, le bit 23 est défini dans le registre de diagnostic.

## Registre de diagnostic

	Bit	Désignation	Description
RECEPTION	0	Overrun error	Dépassement du tampon de réception interne
	1	Non utilisé	
	2	Framing error	Est normalement due à une vitesse de transmission incorrecte
	3	Break error	Interruption de la ligne de données *)
	4	BCC error	Bad Block Check Code ou CRC-16
	5	Non utilisé	
	6	Non utilisé	
	7	Non utilisé	
	8	Length error	La longueur de télégramme est incorrecte
	9	Non utilisé	
	10	Inv tlg	Télégramme non valide ou non autorisé
	11	Non utilisé	
	12	Range error	Adresse d'élément non valide
	13	Value error	Erreur dans la valeur reçue
	14	Missing Media	Supports non valides ou inexistantes
EMISSION	15	Non utilisé	
	16	Retry count	Indique le nombre de répétitions (binaire). (répétitions des télégrammes au format binaire)
	17		
	18		
	19		
	20	NAK response	Réception d'une réponse négative (NAK)
	21	Missing response	Aucune réponse reçue dans les délais
	22	Inv. response	Réception ACK/NAK au lieu de données, ou vice-versa
	23	Target not present	Station de destination non présente ou liaison physique perdue
	24	Non utilisé	
	25	Non utilisé	
	26	Non utilisé	
	27	Non utilisé	
	28	Range error	Adresse d'élément non valide
	29	TxNode_error	Le nœud n'est pas disponible
	30	TxBroadcast_error	Erreur lors de l'envoi d'un télégramme de diffusion via IP
	31	Program error	Tentative d'envoi non autorisée

\*) N'a aucune signification en mode SM0/SS0

Chaque bit défini dans le registre de diagnostic « H » (high), reste en place jusqu'à ce qu'il soit réinitialisé manuellement par le programme utilisateur ou le débogueur.

**Overrun Error (bit 0)**

Cause :

→

Est défini sur « H » en cas de dépassement du tampon interne DUART.

La vitesse de transmission définie est trop élevée  
L'unité centrale ne peut plus traiter tous les caractères reçus. Ceci peut arriver lorsqu'une unité centrale participe à des communications nécessitant un haut débit de transmission via plusieurs interfaces simultanément. En théorie, toutes les interfaces d'une unité centrale (hormis la boucle de courant 20 mA) peuvent être configurées en même temps à la vitesse de transmission maximale de 19 200 bits/s. En pratique, cette erreur peut cependant survenir lorsque plusieurs interfaces gèrent un trafic très

dense. Le programme système gère les interfaces avec différentes priorités. La plus grande priorité est affectée à l'interface 0 et la moins grande à l'interface 3.

Solution : • Réduire la vitesse de transmission  
• Pour une communication plus rapide, réserver, si possible, une interface prioritaire.

#### **Framing Error (bit 2)**

Est défini lorsqu'un caractère mal formaté est reçu (bit d'arrêt manquant). Ceci est normalement dû à une vitesse de transmission mal paramétrée.

4

#### **Break Error (bit 3)**

Est défini sur « H » en cas d'interruption pendant la réception d'un caractère.

Cause : ligne de données coupée ou vitesse de transmission mal paramétrée.

#### **BCC or CRC-16 Error (Bit 4)**

Est défini sur « H » lorsqu'une erreur CRC-16 est détectée dans le télégramme entrant. Ce dernier est alors rejeté.

Réaction de l'esclave : le télégramme reçu est ignoré  
Réaction du maître : le télégramme reçu est ignoré et le dernier télégramme est à nouveau envoyé.

Cause : défaut sur la ligne de données.  
Solution : vérifier l'installation électrique.

#### **Length Error (bit 8)**

Est défini sur « H » en cas de réception d'un télégramme d'une longueur non valide. Cette erreur ne peut pas survenir dans un réseau composé exclusivement de stations PCD (aucune station de fabricants tiers). L'erreur indique qu'un télégramme non valide a été reçu par un système externe. Ceci conduit à une réponse négative (NAK).

#### **Address Error (bit 10)**

Cause : identique à Length Error (aucune réponse négative n'a lieu).

#### **Range Error (Bit 12)**

Est défini sur « H » lorsqu'un télégramme entrant comporte une adresse d'élément PCD non valide. Cette erreur ne peut pas survenir dans un réseau composé exclusivement de stations PCD car le PCD maître contrôle la zone des adresses d'élément lors de la transmission. La station esclave donne une réponse négative à cette erreur (NAK).

#### **Value Error (bit 13)**

Est défini sur « H » lors de la réception d'une valeur de donnée non valide.

Exemple :	L'instruction STXM est utilisée pour télécharger l'horloge. La valeur reçue pour l'heure est 30. La plage horaire autorisée n'est que de 0...23. La station esclave donne une réponse négative à cette erreur (NAK).
<b>RxBroadcast Error (bit 14)</b>	Est défini lorsqu'un télégramme de diffusion non valide est reçu (diffusion IP → nœud IP = 65 535 et adresse S-Bus < 255).
<b>Retry Count (bits 16 à 19)</b>	Indique le nombre de télégrammes répétés au format binaire, qui ont été envoyés pendant l'exécution d'une instruction SRXM ou STXM. Le bit 16 est le bit de point faible. La qualité d'un réseau S-Bus peut être évaluée par la surveillance de ces deux bits.
<b>Negative Response (bit 20)</b>	Est défini sur « H » en cas de réception d'une réponse négative (NAK) d'un esclave. Cela signifie que le maître a préalablement envoyé un télégramme non valide. Contrôler les erreurs suivantes : Value Error, Range Error et Length Error.
<b>Missing Response (bit 21)</b>	Est défini sur « H » lorsqu'aucune réponse n'est reçue de la station esclave dans les délais impartis. Dans ce cas, le télégramme est à nouveau envoyé (deux fois maximum).
Causes possibles :	L'esclave adressé n'est pas disponible. Erreur d'installation dans le réseau (câblage). L'esclave a reçu un télégramme incompréhensible avec une erreur CRC-16.
Solutions :	Contrôler l'esclave (connexions, numéro de station) La terminaison de ligne correcte et les résistances pull-up/down ont-elles été reliées à la première et à la dernière station?
<b>Multiple NAK (Bit 22)</b>	Est défini sur « H » lorsqu'une autre réponse est reçue d'un esclave à la place d'une réponse positive (ACK) ou négative (NAK).
Causes possibles :	<ul style="list-style-type: none"> <li>● Plusieurs esclaves avec le même numéro de station.</li> <li>● Plusieurs maîtres dans le réseau.</li> <li>● Perturbation de la ligne de bus.</li> </ul>
Solutions :	Identique à l'erreur Missing Response
<b>Target not present (bit 23)</b>	Est défini sur « H » lorsque la station de destination ne répond pas dans le réseau. Soit au bout de trois tentatives, soit lorsque la liaison physique n'existe pas.
Causes possibles :	câble de raccordement défectueux ou coupure d'alimentation électrique de la station de destination
<b>Range Error (bit 28)</b>	Est défini sur « H » lorsque les instructions SRXM ou STXM affichent une adresse d'élément (adresse source ou de destination) en dehors de la plage autorisée.

Cause possible : Erreur dans le programme utilisateur  
 Plages surveillées :  
 Entrées/sorties 0...8191  
 Flags 0...8191  
 Minuterie/compteur 0...1599  
 Registre 0...4095

Exemple : Pendant l'exécution de l'instruction STXM suivante, le bit est défini sur « H » pour l'erreur Range Error.

```
STXM 1 ; canal 1
      25 ; 25 registres
      R 1000; adresse de base source
      R 4072; adresse de base de destination
```

Il y a une tentative de transférer le contenu des registres 1000 à 1024 au maître et les registres 4072 à 4096 à l'esclave.

4

**TxNode Error (bit 29)** Est défini sur « H » lorsque le nœud ne figure pas dans la liste de nœuds, s'il n'est pas configuré ou si un télégramme de diffusion non valide a été envoyé (diffusion IP → nœud IP = 65 535 et adresse S-Bus < 255).

**TxBroadcast Error (bit 30)** Est défini sur « H » lorsqu'un télégramme de diffusion non valide a été envoyé (diffusion IP → nœud IP = 65 535 et adresse S-Bus < 255).

**Program Error (bit 31)** Est défini sur « H » pendant l'exécution d'une instruction STXM ou SRXM lorsque l'interface a été définie en mode SS1 ou lorsqu'une instruction similaire a déjà été exécutée (l'indicateur TBSY n'a pas été consulté avant l'exécution de l'instruction).

#### Emission/réception des données STXM/SRXM vers/de l'esclave

Les instructions sont les mêmes que celles de S-Bus. La seule différence est le numéro de canal (8 ou 9) et l'adresse de nœud IP de la station partenaire.



Pour de plus amples détails, voir manuel S-Bus.

**Exemple d'application :**

```
LDL   R 100   ; registre des adresses de destination
      3       ; Adresse S-Bus
      R 100
      15      ; Nœuds IP
STH   TBSY    ; contrôler l'indicateur « émetteur occupé » pour la transmission
JR    H
next

STXM  9       ; logement 9 / B1
      4       ; Nombre d'éléments
      F 500   ; adresse source
      O 32    ; adresse de destination
next :
```

4

**4.2.2 Programmation de Ethernet-S-Bus avec des boîtes de fonctions  
(Saia PG5® FBox)****Exemple :**



La station A émet et reçoit 4 registres quasiment simultanément à destination et en provenance de la station A.

PG5

4

**Paramétrage maître SASI IP**

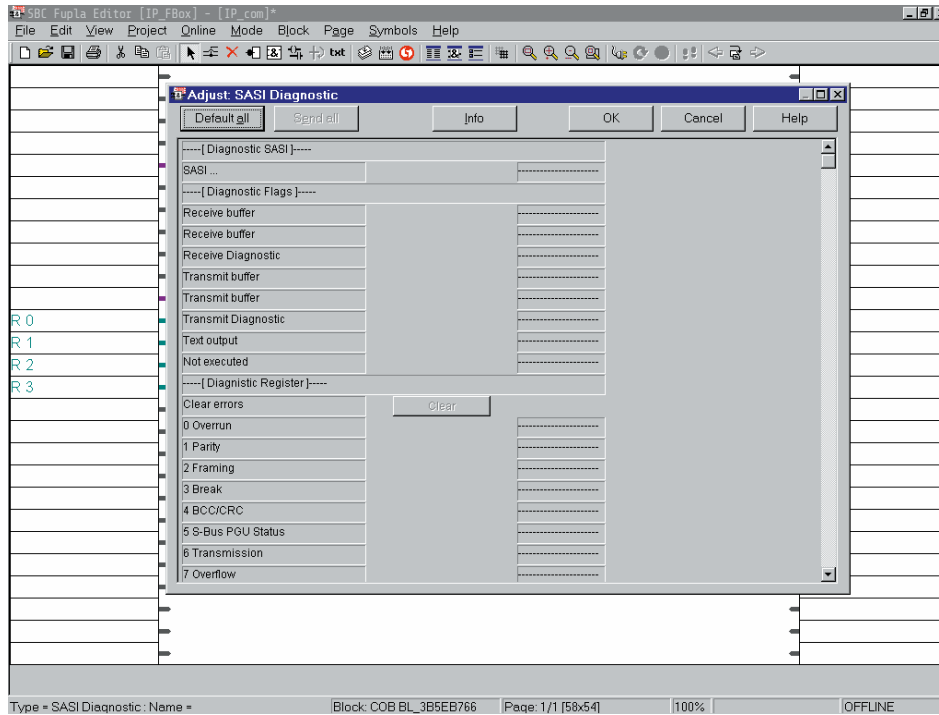
Sélection du canal dans le paramétrage SASI :

- Canal 9 pour PCD1.M130, PCD2.M150
- Canal 8 pour PCD2/4.M170

**Adjust: SASI S-Bus Master IP**

Channel	8 Module B2
Gateway	No
-----[ General ]-----	
SASI	
Clear errors	Clear
-----[ Activity ]-----	
Node	
Station	
Media	
Address	
-----[ Receive diagnostic ]-----	
Receive busy	
Diagnostic	
-----[ Transmit diagnostic ]-----	
Transmit busy	
Diagnostic	
Not executed	

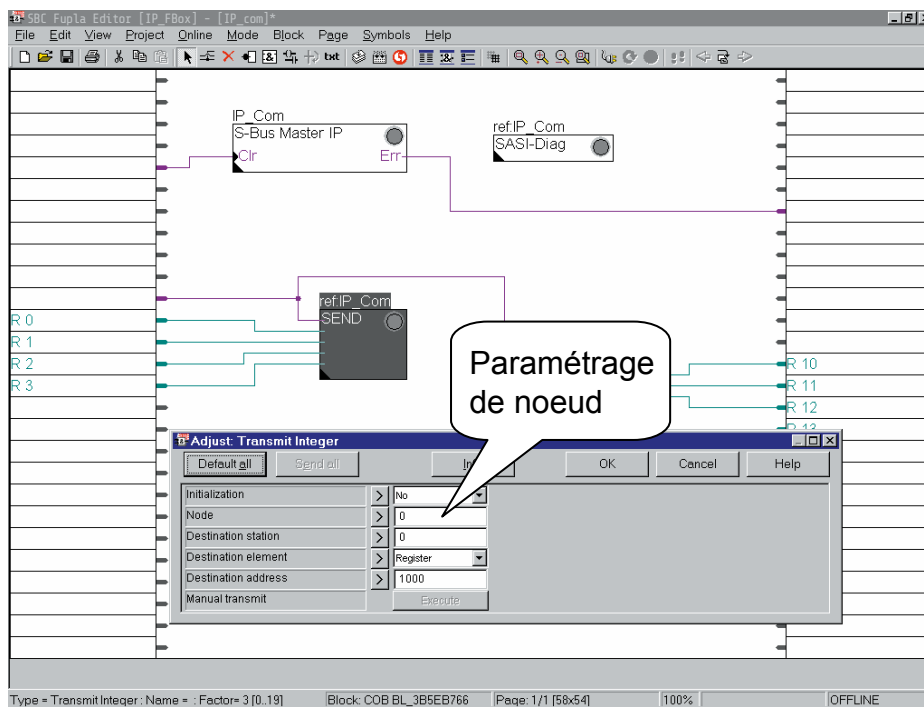
### Paramétrage de la boîte de dialogue SASI IP



4

### Paramétrage de l'émission et de la réception

La seule différence entre l'IP-S-Bus et le S-Bus série est le nœud dans le paramétrage de l'adressage.



### 4.2.3 Multi-maître S-Bus IP

Plusieurs maîtres peuvent être reliés à un port esclave S-Bus IP. Chaque passerelle maîtresse (MGWY) dispose de 4 tampons (PCD1, 1 tampon), qui sont réservés pour le traitement de 4 requêtes en parallèle adressées à MGWY.



Pour plus de détails, voir chapitre 3 « Caractéristiques et fonctions ».

### 4.2.4 Télégrammes de diffusion via S-Bus IP

4

Deux types de transmission différents sont autorisés pour les télégrammes de diffusion (pour la synchronisation avec un client) :

- Diffusion limitée au réseau S-Bus (hors Ethernet)
- Diffusion sur tout le réseau Ethernet, y compris toutes les stations S-Bus dans la station passerelle adressée.



Pour plus de détails, voir chapitre 3 « Caractéristiques et fonctions ».

## 4.3 Programmation de l'Open Data Mode via Ethernet

L'Open Data Mode est généralement utilisé pour la communication du PCD avec une station d'un fabricant tiers, qui ne prend pas en charge le S-Bus. Tous les protocoles tiers peuvent ainsi être implémentés. Deux PCD peuvent, si nécessaire, également communiquer en mode Open Data.

Les stations tierces ne prennent en charge aucun protocole propriétaire (ex. : S-Bus). De plus, seuls les blocs de données brutes (caractères, chaînes sans en-tête) doivent être transmis via IP.

Le PCD peut envoyer les données à une station distante mais il ne peut pas lui demander directement les données en mode client. Les données reçues en mode Open Data sont envoyées à la couche application.

Si le protocole de transmission UDP est utilisé, il n'est pas possible de savoir si la liaison entre deux stations a été interrompue. Cette fonction doit être implémentée par l'utilisateur dans la couche application. Un mécanisme de contrôle pouvant détecter les coupures de communication, a été implémenté dans le protocole TCP.

### 4.3.1 Description du mode Open Data

En mode Open Data, les données brutes sont jointes à l'en-tête UDP ou TCP, puis envoyées. Dans S-Bus sur UDP, les données sont toujours jointes à l'en-tête UDP.



### 4.3.2 Configuration

Le module IP doit être configuré avec l'adresse IP, le masque de sous-réseau et le routeur par défaut comme pour le S-Bus sur IP, à l'aide des paramètres matériels du PG5. Aucune autre configuration n'est requise. Le mode Open Data est initialisé avec l'instruction InitODM dans le programme utilisateur.

### 4.3.3 Remarques importantes dans UDP et TCP en mode Open Data

#### Remarque générale :

En mode Open Data UDP et TCP, en raison de la taille de la boîte de réception entre le module Ethernet TCP/IP et le PCD, il faut noter que la longueur maximale autorisée de la charge utile envoyée ne doit pas dépasser 1536 octets (720 octets avec PCD7F650/F655) par télégramme. Au-delà, les données sont perdues lors de la transmission.

#### Caractéristique TCP :

TCP est un protocole orienté connexion et débit. Il peut arriver que les différents éléments d'un télégramme ou que quelques télégrammes regroupés soient envoyés via Ethernet. Il incombe à l'utilisateur, lors de la réception des télégrammes, de les rassembler, si nécessaire. TCP garantit cependant que les données arrivent dans le bon ordre. TCP a implémenté une couche de contrôle qui garantit les éventuelles répétitions lors de l'envoi. Aucune donnée n'est perdue sur le réseau Ethernet dans TCP.

TCP fournit des informations sur le statut de connexion.

Le principe client-serveur s'applique dans TCP. Avant la connexion des deux terminaux de communication, on fait la distinction client et serveur. Le serveur attend une connexion d'un client. Un serveur peut se connecter via le même port d'entrée avec différents clients. Le client ne peut cependant se connecter qu'à un seul serveur via le même port de sortie. Lorsque la connexion entre le client et le serveur est établie, les deux parties peuvent envoyer des données. Les deux parties sont alors quasiment des clients et peuvent également mettre un terme à une connexion.

#### Caractéristique UDP :

UDP est un protocole sans connexion et orienté télégramme. Les télégrammes sont envoyés séparément et en une seule fois. Il n'y a cependant pas de couche de contrôle via le trafic de données comme pour les répétitions dans TCP. Il ne peut pas être garanti que les télégrammes atteignent le point final. L'utilisateur doit lui-même programmer les mécanismes de contrôle, ex. : handshake. Chaque station peut envoyer des télégrammes à une autre station. Il n'existe également pas de relation client-serveur. Dans UDP également, il n'est pas possible de recevoir des télégrammes de plus de 1536 octets (720 octets avec PCD7F650/F655) (taille de la boîte de réception).

Un indicateur de diagnostic global indique s'il existe une connexion avec le concentrateur/commutateur en amont. Cette connexion peut être testée à tout moment. Avant l'envoi d'un télégramme UDP, par exemple.

#### 4.3.4 Programmation avec une liste d'instructions (IL)

Le mode Open Data est programmé à l'aide des fonctions système.

##### **CSF (Call System Function)**

Il s'agit d'une sorte d'extension d'instruction avec différentes bibliothèques. L'appel s'apparente à un appel de bloc de fonctions. Le module Ethernet TCP/IP dispose d'une bibliothèque. Cette bibliothèque fait partie du microprogramme. Les fonctions sont appelées avec l'instruction CSF. La communication entre le module Ethernet TCP/IP et le PCD a lieu via une boîte de réception commune (tampon d'émission, tampon de réception) à l'aide de ces instructions CSF. L'utilisateur de ces instructions CSF gère lui-même la communication du mode Open Data et est responsable du flux de communication correct via la boîte de réception. Aucun télégramme ne doit ainsi rester en suspens dans la boîte de réception.

Définition de la fonction CSF (Call System Function) :

```
CSF [cc]    <Bibliothèque>
            <Fonction>
            [<paramètre 1>]
            [<paramètre 2>]
            ....
            [<paramètre n>]
```

Exemple :

```
CSF [cc]    S.IP.Library
            S.IP.InitODM
            F 1000
            R 1000
            1000
```

Le PCD2.M480 peut être équipé de deux PCD7.F65x. Selon le logement (B1 ou B2), la désignation de la bibliothèque doit être adaptée pour l'initialisation du mode Open Data.

```
Exemple de logement B1 : CSF [cc]    S.IP.Library_SlotB1
                                   S.IP.InitODM
                                   F 1000
                                   R 1000
                                   1000
```

```
Exemple de logement B2 : CSF [cc]    S.IP.Library_SlotB2
                                   S.IP.InitODM
                                   F 2000
                                   R 2000
                                   1000
```

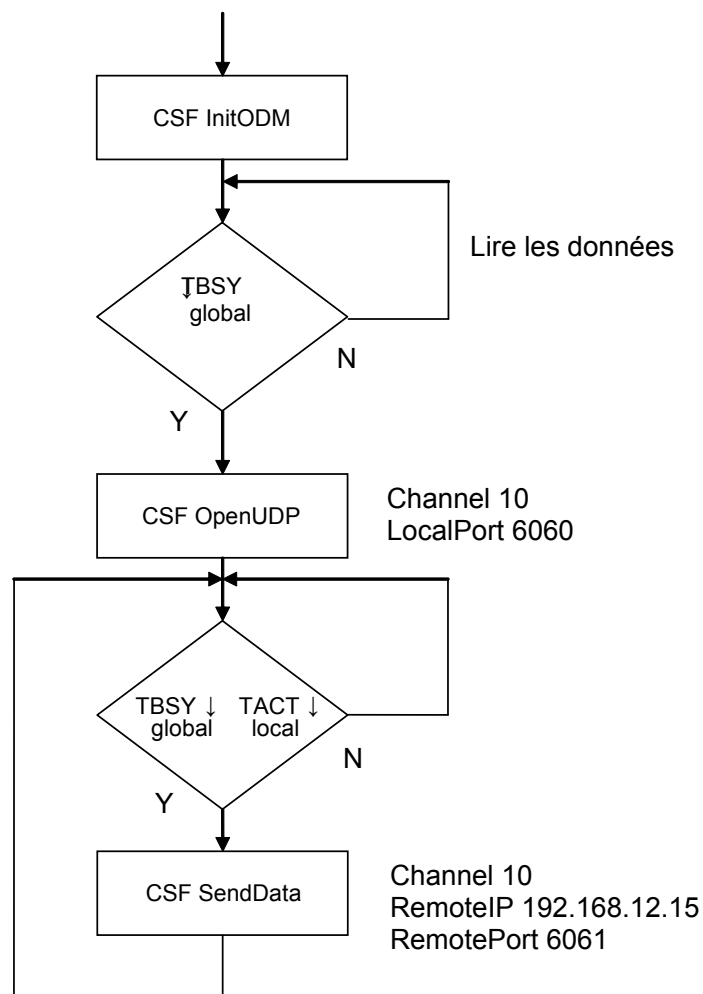
Présentation des CSF disponibles pour le mode Open Data. Les fonctions sont présentées en détail dans les pages suivantes.

CSF 0	IPODMInit
CSF 1	IPODMOpenUDP
CSF 2	IPODMOpenClientTCP
CSF 3	IPODMOpenServerTCP
CSF 4	IPODMClose
CSF 5	IPODMConnectTCP
CSF 6	IPODMDisconnectTCP
CSF 7	IPODMGetConnectionTCP
CSF 8	IPODMAcceptTCP
CSF 9	IPODMSend
CSF 10	IPODMRead
CSF 11	IPODMSendRev
CSF 12	IPODMReadRev

Vous trouverez ci-après deux exemples de logique de schéma fonctionnel expliquant comment travailler avec ces CSF. Pour faciliter la représentation, pour les questions OUI/NON, les étapes « JR -1 » ont été utilisées.

D'abord une communication UDP : La CSF « Open Data Mode » est utile pour initialiser le mode Open Data. Comme l'UDP est sans connexion, aucune liaison n'est établie entre les participants à la communication. Il n'existe ainsi aucune relation client - serveur et chaque participant peut communiquer avec un autre participant UDP de son choix. La condition préalable est que la destination ait ouvert une socket UDP sur un port connu de l'émetteur. Dans l'exemple suivant, on suppose qu'un autre participant se trouve sur le réseau avec l'adresse IP 192.168.12.15 et une socket UDP ouverte sur le port 6061, que nous pouvons adresser. Comme expliqué ultérieurement, le traitement des indicateurs de diagnostic « RDATA » (données arrivées) doit avoir la priorité absolue.

# UDP

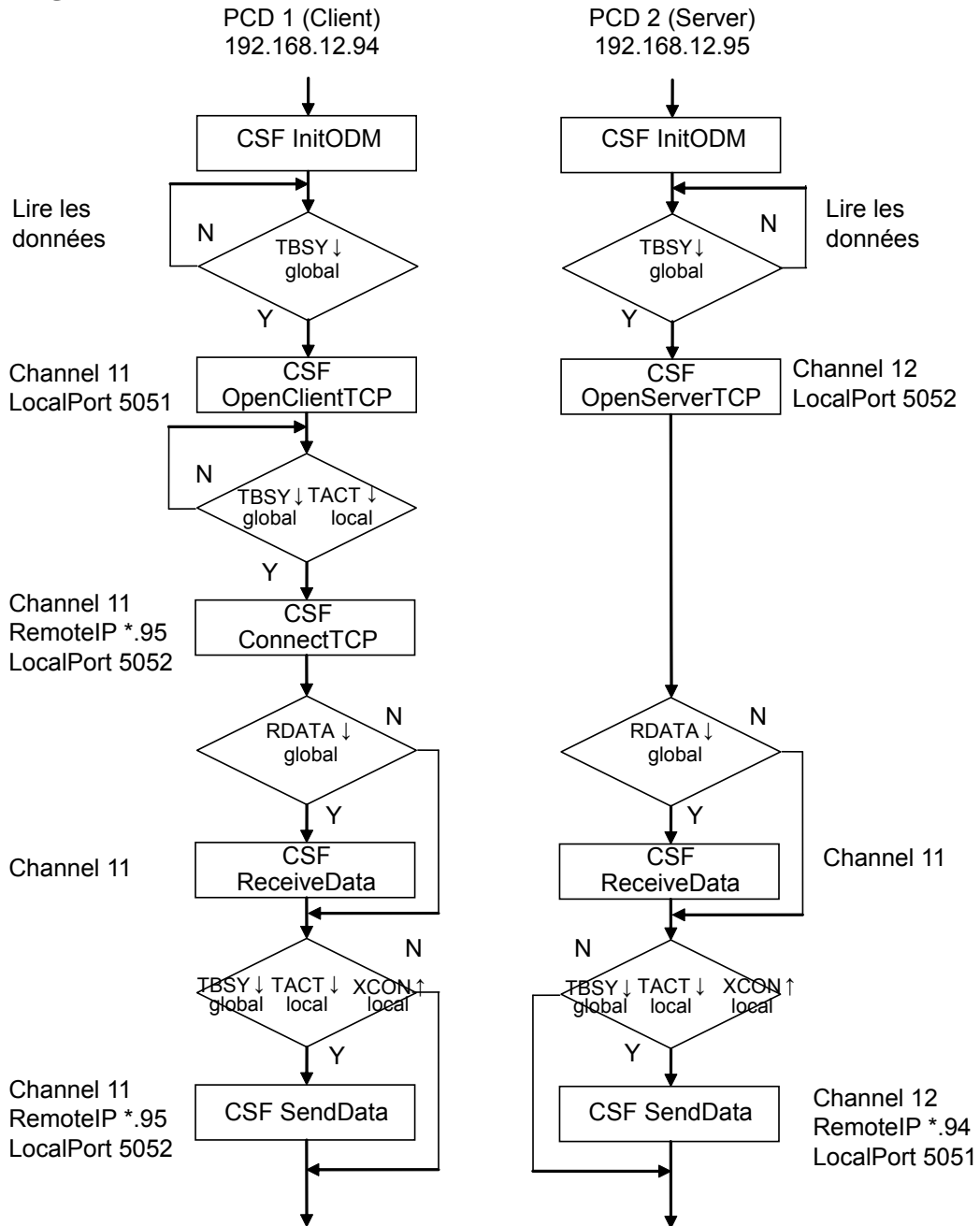


4

Exemple de communication TCP : La CSF « Open Data Mode » est utile pour initialiser le mode Open Data. Comme TCP est orienté connexion, le client sur un serveur doit établir une connexion via laquelle les télégrammes sont échangés. Le client ouvre une socket TCP client avec « OpenClientTCP » et le serveur une socket TCP serveur avec « OpenServerTCP ». Le serveur émet une requête de connexion sur ce port. C'est pourquoi on parle alors souvent de « Listener Socket ». Avec la CSF « ConnectTCP », le client sur le serveur se connecte au participant avec l'adresse 192.168.12.95 et le port 5052. Si la connexion a lieu une fois, il n'y a alors plus aucune différence entre client et serveur. Les deux participants peuvent alors prendre l'initiative et envoyer d'autres télégrammes et requêtes. Les deux parties peuvent également mettre un terme à tout moment à la communication. Comme expliqué ultérieurement, le traitement des indicateurs de diagnostic « RDATA » (données arrivées) doit avoir la priorité absolue.



# TCP



### 4.3.5 InitODM (Open Data Mode)

Sous PG5 1.4 un mécanisme provisoire a été introduit afin de partager le mode ODM entre les bibliothèques de SBC et d'Engiby. Parce que peu d'applications utilisateur sont concernées par cette affaire sous PG5 1.4, cet ancien principe n'est pas décrit ici.

Pour PG5 2.0 un principe plus universel a été mis en place et devrait être respecté par tous les développeurs d'applications de communication en mode ODM. Cette version est décrite dans ce document et permet l'utilisation de l'ODM par plusieurs bibliothèques en parallèle.

4

#### 4.3.5.1 Initialisation du mode ODM

Le mode ODM doit être initialisé une unique fois dans le CPU pour toutes les applications sous la forme d'une instruction SCF (System Code Function). Cette instruction ne peut pas être placée dans l'application utilisateur ni dans la bibliothèque de communication, car une seule fonctionnera dans le cas d'un ODM partagé.

Pour résoudre cela, l'initialisation du mode ODM est à présent exécutée par le gestionnaire de projet SBC (PG5) lorsque l'option correspondante est activée dans le Device Configurator sous l'option Ethernet, dans le groupe TCP/IP.

Initialize Open Data Mode	Yes
Telegram Reading Timeout	1000

En paramétrant Yes, 3 actions sont exécutées gestionnaire de projet :

- Le code nécessaire à l'initialisation de l'ODM est généré
- Le délai d'attente de lecture est défini pour toutes les applications
- Un groupe de symboles système est publié

#### 4.3.5.2 Timeout de lecture des télégrammes

La valeur par défaut (et recommandée) pour le timeout de lecture des télégrammes est de 1000 millisecondes. Cette valeur est valable pour toute partie d'application utilisant l'ODM. Il s'agit du temps maximal autorisé pour la lecture des datagrammes reçus. Chaque application est responsable pour la lecture des paquets (de télégrammes) qui lui sont adressés. Aussi longtemps que le paquet n'est pas lu dans la mémoire tampon de réception, aucun autre (pour une autre application) ne peut être lu. Par conséquent, le timeout de lecture est un mécanisme de sécurité pour s'assurer qu'une application ne peut pas verrouiller le tampon de réception utilisé par une autre application.

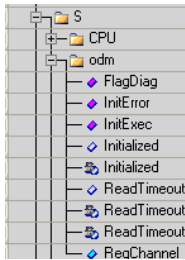
Toutefois, si une application n'est plus en train de lire les datagrammes reçus, cela va ralentir la communication des autres parties d'application de manière drastique et peut même causer la perte des paquets adressés aux autres.

Donc, une application qui utilise l'ODM partagé, doit toujours vérifier s'il y a des paquets entrants et les lire aussi rapidement que possible. Si le traitement immédiat du paquet est impossible, ce dernier doit être lu et stocké dans une mémoire temporaire

(ou ignoré avec indication d'erreur).

### 4.3.5.3 Symboles système

Voici la liste des symboles système publiés.



4

#### Description

s.odm.FlagDiag	Adresse de base de 8 flags de diagnostique. Voir le manuel Ethernet pour les détails.
s.odm.FlagDiag+0	Transmission occupée
s.odm.FlagDiag+1	Réception occupée
s.odm.FlagDiag+2	Reçu les changements d'état de la connexion
s.odm.InitError	Échec de l'initialisation. Valable seulement si InitExec est défini
s.odm.InitExec	L'initialisation a été exécutée
s.odm.Initialized	L'option d'initialisation de l'ODM est définie
s.odm.ReadTimeout	Valeur du timeout de lecture en millisecondes
s.odm.RegChannel	Registre indiquant le canal concerné par la réception d'un paquet

L'application utilisant l'ODM partagé doit tout d'abord contrôler que l'option ODM à été définie et délivrer un message (\$warning (avertissement), \$error (erreur) ou même \$fatal) si ce n'est pas le cas. Les directives assembleur devraient également être utilisées pour éviter d'autres erreurs assembleur si l'ODM n'est pas initialisé. Ceci facilite compréhension et correction des problèmes pour les utilisateurs.

#### Exemple :

```

$ifndef s.odm.initialized

    $error Driver xyz, ODM not initialized. See Device configura-
tor.

    exitm    ;exit macro to avoid further errors

$else

    $if s.odm.initialized = 0

```

```
    $error Driver xyz, ODM not initialized. See Device configura-  
tor.
```

```
    exitm    ;exit macro to avoid further errors
```

```
$endif
```

```
$endif
```

#### 4.3.5.4 Exécution de l'initialisation et erreur(s)

Avant d'utiliser l'ODM partagé, l'application doit s'assurer que l'initialisation s'est exécutée sans erreur. Parce qu'il n'est pas possible de prédire où le code est placé dans la section \$init, cette vérification doit être faite dans le segment COB en utilisant les deux symboles système : s.odm.InitExec et s.odm.InitError.

#### Exemple:

```
$cobseg  
  
    sth    s.odm.InitExec    ;;Executed  
  
    anl    s.odm.InitError   ;;wihtout error  
  
    jr     l l_skip         ;;->Not yet initialized or error  
  
                                ;;Else ready to use ODM  
  
$endcobseg
```

#### 4.3.5.5 Choisir le canal ODM

Ne mélangez pas les indicateurs de canaux S-Bus (8 ou 9) dans le configurateur de dispositif (Device Configurator). Ces canaux sont valables seulement pour le S-Bus et sélectionnent la carte Ethernet accédée.

Pour le mode ODM, chaque CPU a un nombre limité de canaux logiques. Chaque pilote doit utiliser un canal différent. Par conséquent, le numéro de canal ODM doit être réglable par l'utilisateur afin de pouvoir éviter les collisions de canaux.

Les numéros de canaux sont spécifiés dans le chapitre 4.3.7

Un mécanisme pour une allocation dynamique des numéros de canaux est en étude.

Il n'est pas encore introduit dans le présent document.

En attendant les règles suivantes s'appliquent :

- Les canaux 1 à 10 (réglables) devraient être utilisés pour les applications utilisateur et les librairies tierces.
- Les canaux 11 à 19 sont réservés pour des usages à venir
- Les canaux 20 à 29 (non-réglables) sont utilisés pour les boîtes de fonctions (Saia PG5® FBoxes) standard de SBC (Librairie WAA)

4

#### 4.3.5.6 Réception des télégrammes

Comme expliqué plus haut, chaque application utilisant l'ODM partagé doit vérifier la réception des télégrammes de manière cyclique et lire les télégrammes reçus et seulement ceux adressés à son propre canal, cela aussi rapidement que possible.

L'application doit vérifier le canal auquel le paquet est adressé en utilisant le symbole système `s.odm.RegChannel`.

#### Exemple:

```
sth  s.odm.FlagDiag+1

jr   l l_no_data      ;;->No reception

cmp  .s.odm.RegChannel

     My_Channel

acc  z

jr   l l_no_data      ;;->Not for my channel

CSF  ReceiveData
```

#### 4.3.5.7 Réception des changements d'états de la connexion

Les mêmes règles s'appliquent pour la réception des événements de connexion si vous utilisez le filtre de connexion TCP client.

L'application doit vérifier la réception des événements de connexion de manière cyclique et lire les événements reçus et seulement les événements adressés à son canal aussi rapidement que possible.

L'application doit vérifier le canal auquel l'évènement est adressé en utilisant le symbole système `s.odm.RegChannel`.

Notez que le même registre est utilisé pour indiquer le canal des paquets de données et des événements de connexion. Le FW assure qu'un seul des deux drapeaux (réception des données et réception de l'évènement) soit défini en même temps. Donc, il est également important de toujours lire les événements de connexion, même s'ils doivent être ignorés dans certaines situations.

4

Exemple:

```
sth  s.odm.FlagDiag+2

jr   l l_no_event    ;;->No event reception

cmp  s.odm.RegChannel

     My_Channel

acc  z

jr   l l_no_event    ;;->Not for my channel

CSF  Get connection event
```

### 4.3.6 Diagnostic

#### Indicateurs de diagnostic global

Adresse	Nom	Description :
Xxxx	TBSY	Emetteur occupé
xxxx + 1	RDATA	Serveur de réception de données
xxxx + 2	RCON	Connexion
xxxx + 3	Réservé	
xxxx + 4	Réservé	
xxxx + 5	Réservé	
xxxx + 6	XBSY	Liaison physique
xxxx + 7	Réservé	

**Emetteur occupé (TBSY)**↑ Est défini lorsqu'un télégramme se trouve dans le tampon d'émission. Il n'est pas possible d'envoyer des données sur un canal quelconque lorsque cet indicateur est défini.

**Réception de données (RDATA)**↑ Est défini si un télégramme se trouve dans le tampon de réception. Il s'agit de l'indicateur central en mode Open Data et doit être traité en conséquence avec la première priorité. Dès que les données se trouvent dans la boîte de réception du PCD - ceci est affiché avec RDATA « H » (high) - la boîte de réception doit être lue du côté du PCD, les données doivent être traitées et la boîte de réception doit être activée pour d'autres canaux au moyen de la CSF Receive Data. Le tampon d'émission et de réception peut recevoir par transaction (émission/réception) maximum 1536 octets (720 octets avec PCD7F650/F655) de charge utile.

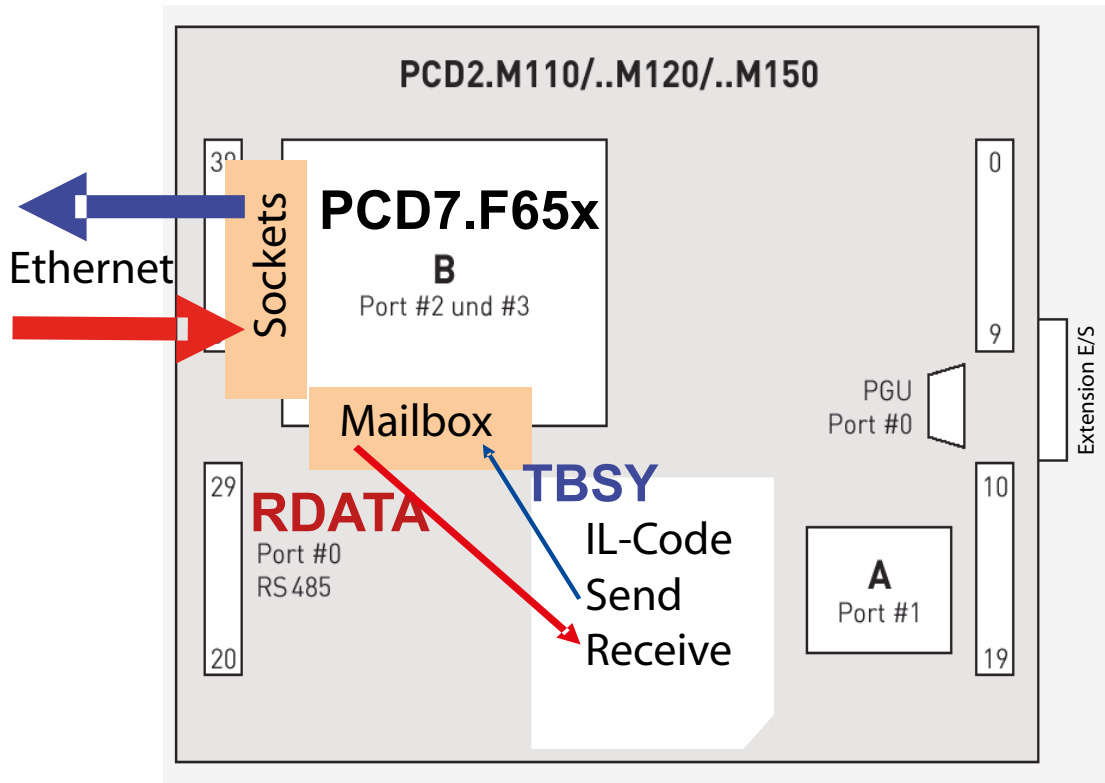
4

Lors de la réception d'un télégramme de données, RDATA « H » (high), le contenu du « registre du canal » (diagnostic global pour InitODM) indique pour quel canal les données sont arrivées. Remarque importante : Si la station locale communique avec plusieurs stations (ex. : un serveur TCP qui est relié à plusieurs clients TCP), le contenu du « registre du canal » doit être sauvegardé avant l'appel de la CSF « Receive Data / ReceiveDataRev » pour continuer de travailler après la récupération des données avec la copie du « registre du canal ». Puis, juste après l'appel de la CSF « ReceiveData / ReceiveDataRev », on passe au prochain télégramme entrant, et le contenu du « registre du canal » peut être modifié.



Voir aussi l'entrée FAQ

#100584 « Receive data in Open Data Mode ».



**Connexion serveur (RCON)↑**

Est défini si un changement de connexion a lieu sur un serveur TCP et la fonction « GetConnectionTCP » doit être exécutée.



**Liaison physique (XBSY)** Indique s'il existe une liaison physique avec le prochain concentrateur/commutateur. Est défini sur « H » (high), si le module Ethernet TCP/IP détecte une liaison physique et sur « L » (low) si la liaison physique est perdue.

### Indicateurs de diagnostic de canal

En mode Open Data, l'utilisateur programme pratiquement directement dans un niveau d'abstraction via les sockets Berkeley. Pour faciliter la programmation et la rendre abstraite, on utilise des canaux. Pour pratiquement chaque socket sur le module Ethernet TCP/IP (mode Open Data UDP et TCP), il existe un canal virtuel via lequel a lieu la communication. Chaque canal possède ses propres indicateurs de diagnostic et un registre de diagnostic. Sur ces canaux a lieu l'accès aux fonctions « OpenUDP », « OpenClientTCP », « OpenServerTCP », « Close », « ConnectTCP », « Discon- nectTCP », « AcceptConnectionTCP », « SendData », « SendDataRev », « ReceiveData » et « ReceiveDataRev ».

4

Adresse	Nom	Description :
xxx	RDIA	Diagnostic de réception
xxxx + 1	TACT	Emetteur actif
xxxx + 2	TDIA	Diagnostic d'émission
xxxx + 3	XCON	Port connecté
xxxx + 4	NEXE	Non-exécution
xxxx + 5	Réservé	
xxxx + 6	Réservé	
xxxx + 7	Réservé	

**Diagnostic de réception (RDIA)↑** Un diagnostic de réception est disponible dans le registre de diagnostic. L'indicateur est défini sur « H »

lorsqu'une erreur est détectée lors de la réception d'un télégramme.

Une description détaillée de l'erreur figure dans le registre de diagnostic (bits 0 à 15). L'indicateur est réinitialisé dès que tous les bits de diagnostic d'émission ont été réinitialisés dans le registre de diagnostic

**Emetteur actif (TACT)↑**

Est défini lorsqu'un télégramme se trouve dans le tampon d'émission. Il n'est pas possible d'envoyer d'autres données à ce canal ou de se connecter à/déconnecter de ce canal tant que TACT n'est plus actif.

**Diagnostic d'émission (TDIA)↑**

Le registre de diagnostic comprend un diagnostic d'émission. Il est défini sur « H » si une erreur est détectée lors de l'envoi d'un télégramme. Une description détaillée de l'erreur figure dans le registre de diagnostic (bits 16 à 31). L'indicateur est réinitialisé, dès que tous les bits de diagnostic d'émission ont été réinitialisés dans le registre de diagnostic.

**Port connecté (XCON)↑**

Le port TCP est relié à une autre station TCP.

**Not execution (NEXE)↑** Est défini si une instruction n'est pas terminée ou si elle n'est pas exécutée. L'indicateur est réinitialisé par l'instruction de communication suivante.

### Registre de diagnostic de canal

	Bit	Désignation	Description
RECEPTION	0	Read timeout error	Pas de lecture de télégramme dans le tampon
	1	Non utilisé	
	2	Non utilisé	
	3	Non utilisé	
	4	Non utilisé	
	5	Non utilisé	
	6	Non utilisé	
	7	Non utilisé	
	8	Length error	Longueur du télégramme supérieure à celle du tampon
	9	Non utilisé	
	10	Address error	Erreur de conversion de l'adresse IP dans le tampon
	11	Non utilisé	
	12	Range error	Adresse d'élément non valide
	13	Status error	Statut non valide
	14	Rx Mailbox error	La boîte de réception ne contient aucune donnée
15	Rx Channel error	Numéro de canal incorrect	
EMISSION	16	Non utilisé	
	17	Non utilisé	
	18	Non utilisé	
	19	Non utilisé	
	20	Non utilisé	
	21	Non utilisé	
	22	Non utilisé	
	23	Stat. not present	Station de destination non présente
	24	Non utilisé	
	25	Non utilisé	
	26	Non utilisé	
	27	Non utilisé	
	28	Range error	Adresse d'élément non valide
	29	Add error	Port non disponible ou le port et l'adresse ne concordent pas
	30	No connection	La connexion avec cette station n'est pas ouverte
	31	Program error	Tentative d'envoi non autorisée

Chaque bit défini sur « H » (high) dans le registre de diagnostic, reste ainsi jusqu'à ce qu'il soit réinitialisé manuellement par le programme utilisateur ou le débogueur.

**Read timeout error (bit 0)** Est défini sur « H » en l'absence de lecture d'un télégramme dans le tampon. La temporisation du tampon est trop courte ou le canal n'a pas été lu.

**Length error (bit 8)** Est défini sur « H » lorsque la longueur du télégramme reçu dépasse celle du tampon

<b>Address error (bit 10)</b>	Est défini sur « H » lorsqu'une erreur est survenue lors de la conversion de l'adresse IP dans le tampon : <ul style="list-style-type: none"> <li>• Modification dans le registre</li> <li>• Texte trop court</li> </ul>
<b>Range error (bit 12)</b>	Est défini sur « H » lorsque l'adresse d'élément n'est pas valide.
<b>Mailbox error (bit 14)</b>	Est défini sur « H » lorsque la boîte de réception lue ne comporte aucune donnée.
<b>Channel error (bit 15)</b>	Est défini sur « H » lorsque le numéro de canal lu est incorrect.
<b>Target not present (bit 23)</b>	Est défini sur « H » lorsque la station de destination sur le réseau ne peut pas être adressée. Par exemple, lorsque la liaison physique avec le concentrateur/commutateur suivant n'existe pas ou lorsqu'un télégramme est envoyé via UDP (pas PCD7.F655). Ou lorsque la socket d'émission dans la pile TCP/IP signale une erreur et lorsque le télégramme n'a ainsi pas été envoyé sur le réseau Ethernet.
<b>Range error (bit 28)</b>	Est défini sur « H » lorsque l'adresse d'élément n'est pas valide.
<b>TxNode error (bit 29)</b>	Est défini sur « H » lorsque le nœud n'est pas disponible.
<b>No connection (bit 30)</b>	Est défini sur « H » lorsque la connexion avec cette station n'est pas ouverte.

#### Disposition générale relative à la gestion de canaux pour OpenUDP, OpenClientTCP et OpenServerTCP

#### 4.3.7. Nombre de canaux ODM

Le nombre total de canaux autorisés par PCD (TCP et UDP) dépend du système du PCD. Voir ci-après :

PCD system	PCD1.M130	PCD2.M150	PCDx.M170	PCD2.M480
Nombre de canaux	16	32	32	32
PCD system	PCD3.Mxxxx	PCD1.M2xx0	PCD2.M5	
Nombre de canaux	32	32	32	

Si S-Bus UDP nécessite un canal lorsque le port d'entrée (serveur S-Bus) et de sortie (client S-Bus) est défini sur 5050. Si le serveur S-Bus et le client S-Bus utilisent différents ports, le système a alors besoin de deux canaux. En mode Open Data, les autres canaux peuvent alors être utilisés. Dans cette table, tous les canaux ouverts avec « OpenUDP », « OpenTCPClient » et « OpenTCPServer » sont totalisés.

### 4.3.8 OpenUDP

Ouvre un canal UDP en mode IP Open Data (correspond à une « socket » et à un appel « bind » dans la norme Berkeley). Après cet appel, les télégrammes peuvent être envoyés via ce canal UDP.

Cette fonction peut uniquement être exécutée lorsque TBSY est défini sur « L » (low).

```
CSF      [cc]   S.IP.Library      ; Bibliothèque IP
          S.IP.OpenUDP ; Fonction Ouverture de canal UDP
          →     LocalPort      ; port IP local via lequel les données
          ; sont envoyées/reçues (R/K)
          →     Ch_Diag_Flag    ; base des indicateurs de diagnostic de canal
          ; (8 indicateurs) (F)
          →     Ch_Diag_Register ; registre de diagnostic de canal (R)
```

4

**LocalPort :** Sélection du port local pour la communication UDP en mode Open Data  
 0 = un port libre du système est sélectionné  
 X = affectation directe du port local par l'utilisateur  
 Le port local 5050 est réservé pour le mode S-Bus-UDP.

**TACT↑** lorsque l'instruction « OpenUDP » a été placée dans la boîte de réception pour le module Ethernet TCP/IP.

**TACT↓** après la réponse positive (ACK) du module Ethernet TCP/IP.  
 Une nouvelle instruction en mode Open Data peut être exécutée.

**NEXE↑** Lorsque, par exemple, le port IP local est déjà utilisé.

**Exemple :**

```
CSF      [cc]   S.IP.Library      ; bibliothèque
          S.IP.OpenUDP ; Fonction
          10      ; canal 10
          5000    ; Port 5000
          F 1010  ; base des indicateurs de
          ; diagnostic de canal
          R 1     ; registre de diagnostic de canal
```

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

### 4.3.9 OpenClientTCP

Ouvre un canal TCP en mode IP Open Data (correspond à une « socket » et à un appel « bind » dans la norme Berkeley). Après cet appel, le client TCP peut se connecter à un serveur TCP.



Cette fonction peut uniquement être exécutée lorsque TBSY est défini sur « L » (low).

4

```

CSF      [cc]  S.IP.Library      ; Bibliothèque IP
          S.IP.OpenClientTCP ; Fonction Ouverture de canal TCP
          → LocalPort      ; port IP local via lequel les données
          ; sont envoyées/reçues (R/K)
          → Ch_Diag_Flag    ; base des indicateurs de diagnostic de canal
          ; (8 indicateurs) (F)
          → Ch_Diag_Register ; registre de diagnostic de canal (R)
          ; Conn_Tout
          → Conn_Tout      ; temporisation de la connexion : 0=-, x=sec.
          ; (R/K)

```

**LocalPort :** Sélection du port local pour la communication TCP en mode Open Data  
 0 = un port libre du système est sélectionné  
 X = affectation directe du port local par l'utilisateur  
 Même si le port local 5050 est réservé pour la communication S-Bus UDP, ce port peut également être utilisé dans TCP.

**Conn\_Tout :** Si la valeur Conn\_Tout (secondes) est dépassée sans qu'un télégramme ne soit reçu via le canal, le canal est fermé (en fonction de la réception).

- 0 = aucun contrôle de temporisation
- x = contrôle de temporisation toutes les x secondes

**TACT↑** lorsque l'instruction « OpenClientTCP » a été placée dans la boîte de réception pour le module Ethernet TCP/IP.

**TACT↓** après la réponse positive (ACK) du module Ethernet TCP/IP. Une nouvelle instruction en mode Open Data peut être exécutée.

**NEXE↑** Lorsque, par exemple, le port IP local est déjà utilisé.

**Exemple :**

```

CSF      [cc]  S.IP.Library      ; bibliothèque
          S.IP.OpenClientTCP ; Fonction
          10      ; canal 10
          5555     ; Port 5555
          F 1010   ; base des indicateurs de
          ; diagnostic de canal
          R 1      ; registre de diagnostic de canal
          R 2      ; temporisation de la
          ; connexion

```

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

### 4.3.10 OpenServerTCP

Ouvre un canal TCP serveur en mode IP Open (correspondant à une « socket », et à un appel « bind » et « accept » dans la norme Berkeley). Après cet appel, le serveur TCP est prêt à accepter une connexion d'un client TCP.



Cette fonction peut uniquement être exécutée lorsque TBSY est défini sur « L » (low).

4

CSF [cc]	S.IP.Library	; Bibliothèque IP
	S.IP.OpenServerTCP	; Fonction Ouverture de canal UDP
→	Channel	; n° de canal (R/K)
→	LocalPort	; port IP local via lequel les données ; sont envoyées/reçues (R/K)
→	Ch_Diag_Flag	; base des indicateurs de diagnostic de canal ; (8 indicateurs) (F)
→	Ch_Diag_Register	; registre de diagnostic de canal (R) Conn_Tout
→	Connection_filter	; filtre de connexion (R/K)
→	Conn_Tout	; temporisation de la connexion : 0=-, x=sec. ; (R/K)

**LocalPort :** Sélection du port local pour la communication UDP en mode Open Data  
 0 = un port libre du système est sélectionné  
 X = affectation directe du port local par l'utilisateur  
 Même si le port local 5050 est réservé pour la communication S-Bus UDP, ce port peut également être utilisé dans TCP.

**Connection\_filter** • 0 = aucun filtre, aucune information de connexion disponible (acceptation automatique de n'importe quel client)  
**(uniquement sur TCP-serveur)** • 1 = aucun filtre, information de connexion disponible (acceptation automatique de n'importe quel client)  
 • 2 = demande de filtre pour acceptation du client, infos de connexion disponibles

**Conn\_Tout :** Si la valeur Conn\_Tout (secondes) est dépassée sans qu'un télégramme ne soit reçu via le canal, le canal est fermé (en fonction de la réception).  
 • 0 = aucun contrôle de temporisation  
 • x = contrôle de temporisation toutes les x secondes

**TACT↑** lorsque l'instruction « OpenUDP » a été placée dans la boîte de réception pour le module Ethernet TCP/IP.

**TACT↓** après la réponse positive (ACK) du module Ethernet TCP/IP. Une nouvelle instruction en mode Open Data peut être exécutée.

**NEXE↑** Lorsque, par exemple, le port IP local est déjà utilisé.

**Exemple :** CSF [cc] S.IP.Library ; bibliothèque  
 S.IP.OpenServerTCP ; Fonction  
 10 ; canal 10  
 5432 ; Port 5432  
 F 1010 ; base des indicateurs de  
 ; diagnostic de canal  
 R 1 ; registre de diagnostic de canal  
 0 ; aucun filtre  
 R 2 ; temporisation de la  
 ; connexion

4

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

#### 4.3.11 Close

Ferme un canal en mode IP Open Data (correspond à une « closesocket » dans la norme Berkeley).



Cette fonction peut uniquement être exécutée lorsque TACT est défini sur « L » (low)

CSF [cc] S.IP.Library ; Bibliothèque IP  
 S.IP.Close ; fonction Fermeture de canal  
 → Channel ; n° de canal (R/K)

**Exemple :** CSF [cc] S.IP.Library ; bibliothèque  
 S.IP.Close ; Fonction  
 10 ; canal 10

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

Les indicateurs de diagnostic définis pour ce canal et le registre de diagnostic ne doivent plus être utilisés après la « fermeture » (Close) jusqu'à ce qu'ils soient réinitialisés et assignés. La CSF « Close » interrompt immédiatement la communication entre les deux stations sur le PCD7.F650/1/2. Le PCD7.F655 se comporte alors de manière un peu différente. La différence est décrite dans l'exemple suivant :  
 Un client TCP se connecte à un serveur TCP et envoie un télégramme L'indicateur de diagnostic global RDATA↑ est défini sur le serveur. Le serveur ne lit momentanément pas le télégramme. Le client effectue maintenant un appel « Close », la socket est alors fermée sur le client. La socket est d'abord fermée sur le serveur lorsque le télégramme a été lu et l'indicateur de diagnostic global RDATA↑ est ainsi en position « low ». Lors de la fermeture de la connexion, aucun télégramme n'est alors perdu dans le serveur TCP.

### 4.3.12 ConnectTCP

Relie un canal TCP client en mode IP Open Data à un canal TCP serveur (correspond à « connect » dans la norme Berkeley). Après cet appel, la communication peut avoir lieu via TCP.



Cette fonction peut uniquement être exécutée lorsque TACT et TBSY sont définis sur « L » (low).

4

```
CSF [cc]  S.IP.Library           ; Bibliothèque IP
           S.IP.ConnectTCP      ; fonction Connexion au canal TCP
           → Channel            ; n° de canal (R/K)
           → RemoteIP/Node      ; adresse IP du serveur distant
                                   ; (R/K/X)*
           → Remote Port        ; Port IP du serveur distant (R/K)
           *) voir chapitre : Décodage des adresses IP
```

**TBSY↑** lorsque l'instruction « ConnectTCP » a été placée dans la boîte de réception pour le module Ethernet TCP/IP.

**TACT↑** lorsque l'instruction « ConnectTCP » a été placée dans la boîte de réception pour le module Ethernet TCP/IP.

**TBSY↓** après que le module Ethernet TCP/IP ait lu l'appel « ConnectTCP » dans la boîte de réception. Une nouvelle instruction en mode Open Data peut être exécutée.

**TACT↓** après la réponse positive (ACK) du module Ethernet TCP/IP. Une nouvelle instruction en mode Open Data peut être exécutée.

**XCON↑** lorsque l'appel « connected-Event » est retourné par le module Ethernet TCP/IP. La connexion entre le client TCP et le serveur TCP est maintenant établie.

```
Exemple : CSF [cc]  S.IP.Library           ; bibliothèque
                S.IP.ConnectTCP      ; Fonction
                10                    ; canal 10
                R5                     ; Adresse IP distante
                5555                   ; Port 5555
```

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

Tant que la CSF ConnectTCP n'est pas terminée, ses paramètres « RemoteIP/Node » et « RemotePort » ne doivent pas être modifiés.



### 4.3.13 DisconnectTCP

Déconnecte un canal TCP en mode IP Open Data. (correspond à une « closesocket » dans la norme Berkeley). Cette fonction peut être exécutée sur un canal de type client et serveur TCP.



Cette fonction peut uniquement être exécutée lorsque TACT et TBSY sont définis sur « L » (low).

CSF [cc]	S.IP.Library ; Bibliothèque IP S.IP.DisconnectTCP ; fonction Déconnexion du canal TCP → Channel ; n° de canal (R/K) → RemoteIP/Node ; adresse IP du serveur distant ; (R/K/X)* → Remote Port ; Port IP du serveur distant (R/K) *) voir chapitre : Décodage des adresses IP
<b>TBSY</b> ↑	lorsque l'instruction « DisconnectTCP » a été placée dans la boîte de réception pour le module Ethernet TCP/IP.
<b>TACT</b> ↑	lorsque l'instruction « ConnectP » a été placée dans la boîte de réception pour le module Ethernet TCP/IP.
<b>TBSY</b> ↓	après que le module Ethernet TCP/IP ait lu l'appel « ConnectTCP » dans la boîte de réception. Une nouvelle instruction en mode Open Data peut être exécutée.
<b>TACT</b> ↓	après la réponse positive (ACK) du module Ethernet TCP/IP. Une nouvelle instruction en mode Open Data peut être exécutée.
<b>XCON</b> ↑	lorsque l'appel « connected-Event » est retourné par le module Ethernet TCP/IP. La connexion entre le client TCP et le serveur TCP est maintenant établie.

**Exemple :** CSF [cc] S.IP.Library ; bibliothèque  
S.IP.DisconnectTCP ; Fonction  
10 ; canal 10  
R5 ; Adresse IP distante  
5555 ; Port 5555

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

La CSF « Disconnect » interrompt immédiatement la connexion entre les deux stations sur le PCD7.F650/1/2 et l'indicateur de diagnostic XCON ↓ indique le statut de la connexion. Le PCD7.F655 se comporte alors de manière un peu différente. La différence est décrite dans l'exemple suivant : Un client TCP se connecte à un serveur TCP et envoie un télégramme L'indicateur de diagnostic global RDATA ↑ est placé sur le serveur. Le serveur ne lit momentanément pas le télégramme. Le client effectue maintenant une déconnexion (Disconnect), l'indicateur de diagnostic XCON↓ est alors défini sur « low » sur le client. Sur le serveur, l'indicateur de diagnostic XCON↓ est d'abord défini sur « low », lorsque le télégramme a été lu et l'indicateur de diagnostic global RDATA↓ est également défini sur « low ». Lors de l'interruption de la connexion, aucun télégramme n'est alors perdu dans le serveur TCP.

#### 4.3.14 « GetConnectionTCP »

Lit les données de connexion d'un canal TCP en mode IP Open Data

**RCON**↑ Indique une modification d'un client ou serveur TCP. La modification peut être lue du serveur avec l'instruction « GetConnectionTCP ». L'information concernant le statut « Requête de connexion/connecté/déconnecté » peut uniquement être utilisée lorsque le filtre a été défini sur 1 ou 2 dans le serveur TCP préalablement défini. « GetConnectionTCP » peut uniquement être exécuté sur un canal TCP serveur. Si plusieurs canaux sont utilisés, l'accès s'effectue sur le canal qui est indiqué dans le registre du canal.

4

**CSF** [cc] S.IP.Library ; Bibliothèque IP  
 S.IP.GetconnectionTCP ; fonction Données de connexion du canal TCP  
 → Channel ; n° de canal (R/K)  
 → RemoteIP/Node ; adresse IP du serveur distant  
 ; (R/K/X)\*  
 → Remote Port ; Port IP du serveur distant (R/K)  
 Status ; statut de la connexion\*\*  
 \*) voir chapitre : Décodage des adresses IP  
 \*\*) Statut : 0 = le client demande une connexion et attend l'accord  
 du serveur  
 1 = connecté  
 2 = déconnecté

**RCON**↓ Après l'exécution de l'instruction GetconnectionTCP

**Exemple :** CSF [cc] S.IP.Library ; bibliothèque  
 S.IP.GetconnectionTCP ; Fonction  
 10 ; canal 10  
 R100 ; Adresse IP distante  
 R101 ; Port distant  
 R102 ; statut

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

### 4.3.15 AcceptConnectionTCP

Acceptation d'une connexion sur un canal TCP serveur en mode IP Open Data Cette fonction peut uniquement être utilisée en relation avec une station TCP serveur qui a été préalablement définie avec l'option de filtre de connexion 1 ou 2 dans « OpenServerTCP ».



Cette fonction peut uniquement être exécutée lorsque TACT et TBSY sont définis sur « L » (low).

4

```
CSF [cc]  S.IP.Library           ; Bibliothèque IP
           S.IP.AcceptconnectionTCP ; fonction Données de connexion du canal TCP
           → Channel             ; n° de canal (R/K)
           → Status              ; statut de la connexion*
           *) Statut :           0 = connexion acceptée
                                1 = connexion refusée
```

```
Exemple : CSF [cc]  S.IP.Library           ; bibliothèque
                  S.IP.AcceptconnectionTCP ; Fonction
                  10             ; canal 10
                  0               ; Statut = connexion
                                 ; acceptée
```

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

### 4.3.16 SendData

Envoie des données via un canal en mode IP Open Data (correspond à un appel « send » pour TCP ou « sendto » pour UDP dans la norme Berkeley).



Cette fonction peut uniquement être exécutée lorsque TACT et TBSY sont définis sur « L » (low).

CSF [cc]	S.IP.Library	; Bibliothèque IP	<b>4</b>
	S.IP.SendData	; fonction Envoi de données	
→	Channel	; n° de canal (R/K)	
→	RemoteIP/Node	; Adresse IP pour l'envoi de données ; (R/K/X)*	
→	Remote Port	; Port IP du serveur distant (R/K)	
→	Datalength	; Longueur des données à envoyer (R/K)**	
	Data	; Tampon de données (R/X/DB)	

\*) voir chapitre : Décodage des adresses IP

\*\* ) La longueur max. des données envoyées est de 1536 octets

**TACT**↑ lorsque l'instruction « SendData » a été placée dans la boîte de réception pour le module Ethernet TCP/IP.

**TBSY**↑ lorsque l'instruction « SendData » a été placée dans la boîte de réception pour le module Ethernet TCP/IP.

**TACT**↓ après la réponse positive (ACK) du module Ethernet TCP/IP.

**TBSY**↓ après que le module Ethernet TCP/IP ait lu l'appel « SendData » dans la boîte de réception. Une nouvelle instruction en mode Open Data peut être exécutée. **Exemple :** CSF [cc] S.IP.Library  
; bibliothèque

S.IP.SendData; Fonction

10 ; canal 10

R100 ; Adresse IP distante

R101 ; Port distant

100 ; Longueur des données à envoyer

R1000 ; Début du tampon d'émission

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

Le tampon de données ne doit pas être modifié avant que l'instruction « SendData » ne soit complètement terminée.

### 4.3.17 SendDataRev

Inverse l'ordre de transmission des octets via un canal en mode IP Open Data (correspond à un appel « send » pour TCP ou « sendto » pour UDP dans la norme Berkeley).

SendDataRev inverse les octets lorsque la station distante a le format « *Intel* » (voir tableau ci-après).



Cette fonction peut uniquement être exécutée lorsque TACT et TBSY sont définis sur « L » (low).

4

CSF	[cc]	S.IP.Library	; Bibliothèque IP
		S.IP.SendDataRev	; Fonction Envoi de données
→		Channel	; n° de canal (R/K)
→		RemotelIP/Node	; Adresse IP pour l'envoi de données ; (R/K/X)*
→		Remote Port	; Port IP du serveur distant (R/K)
→		Datalength	; Longueur des données à envoyer (R/K)**
→		Data	; Tampon de données (R/X/DB)
		*) voir chapitre : Décodage des adresses IP	
		**) La longueur max. des données envoyées est de 1536 octets	

**TACT**↑ lorsque l'instruction « SendData » a été placée dans la boîte de réception pour le module Ethernet TCP/IP.

**TBSY**↑ lorsque l'instruction « SendData » a été placée dans la boîte de réception pour le module Ethernet TCP/IP.

**TACT**↓ après la réponse positive (ACK) du module Ethernet TCP/IP.

**TBSY**↓ après que le module Ethernet TCP/IP a lu l'appel « SendData » dans la boîte de réception. Une nouvelle instruction en mode Open Data peut être exécutée.

**Exemple :**

CSF	[cc]	S.IP.Library	; bibliothèque
		S.IP.SendDataRev	; Fonction
		10	; canal 10
		R100R	; Adresse IP distante
		R101	; Port distant
		100	; Longueur des données à envoyer
		R1000	; Début du tampon d'émission

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

Le tampon de données ne doit pas être modifié avant que l'instruction « SendData » ne soit complètement terminée.

### 4.3.18 ReceiveData

Reçoit des données via un canal en mode IP Open Data

Cette fonction peut uniquement être exécutée lorsque RDATA est défini sur « H » (high). Si plusieurs canaux sont utilisés, le canal sur lequel les données sont reçues doit être indiqué dans le paramètre de canal (registre de canal). Le contenu du registre de canal (configuré dans InitODM) doit être lu directement avant l'appel de la CSF « ReceiveData » et sa copie doit être utilisée dans l'instruction sous forme de « Channel ».

4

CSF [cc]	S.IP.Library	; Bibliothèque IP
	S.IP.ReceiveData	; Fonction
→	Channel	; n° de canal (R/K)
→	RemoteIP/Node	; Adresse IP pour l'envoi de données ; (R/K/X)*
←	Remote Port	; Port IP du serveur distant (R/K)
→	Max_ Datalength	; Taille max. du tampon de données en octets ; (0 = aucun contrôle)**
→	Datalength	; Longueur des données à envoyer (R/K)**
→	Data	; Tampon de données (R/X/DB)

\*) voir chapitre : Décodage des adresses IP  
 \*\*) La longueur max. des données envoyées est de 1536 octets

**RDATA↓** après l'exécution de l'instruction ReceiveData

**Exemple :**

CSF [cc]	S.IP.Library	; bibliothèque
	S.IP.ReceiveData	; Fonction
	10	; canal 10
	R100R	; Adresse IP distante
	R101	; Port distant
	100	; Longueur des données
	R1	; Données reçues en octets
	R1000	; Début du tampon d'émission

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

### 4.3.19 ReceiveDataRev

Inverse l'ordre de réception des octets via un canal IP en mode « Open Data ». ReceiveDataRev inverse les octets lorsque la station distante a le format « Intel » (voir tableau ci-après).

Cette fonction peut uniquement être exécutée lorsque RDATA est défini sur « H » (high). Si plusieurs canaux sont utilisés, le canal sur lequel les données sont reçues doit être indiqué dans le paramètre de canal (registre de canal). Le contenu du registre de canal (configuré dans InitODM) doit être lu directement avant l'appel de la CSF « ReceiveData » et sa copie doit être utilisée dans l'instruction sous forme de « Channel ».

CSF [cc]	S.IP.Library	; Bibliothèque IP
	S.IP.ReceiveDataRev	; Fonction Réception de données
→	Channel	; n° de canal (R/K)
→	RemotelIP/Node	; Adresse IP pour l'envoi de données ; (R/K/X)*
←	Remote Port	; Port IP du serveur distant (R/K)
→	Max_ Datalength	; Taille max. du tampon de données en octets ; (0 = aucun contrôle)**
→	Datalength	; Longueur des données reçues (R/K)**
→	Data	; Tampon de données de réception (R/X/

DB)

\*) voir chapitre : Décodage des adresses IP

\*\* ) La longueur max. des données envoyées est de 1536 octets

**RDATA**↓ après l'exécution de l'instruction ReceiveData

**Exemple :**

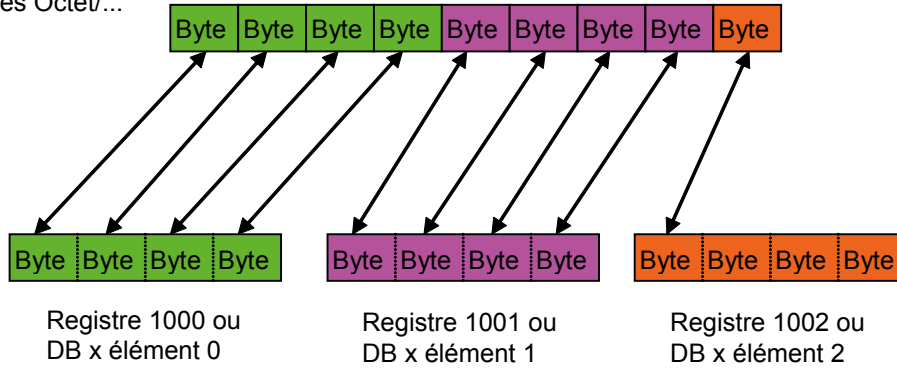
CSF [cc]	S.IP.Library	; bibliothèque
	S.IP.ReceiveDataRev	; Fonction
	10	; canal 10
	R100R	; Adresse IP distante
	R101	; Port distant
	100	; Longueur des données
	R1	; Données reçues en ; octets
	R1000	; Début du tampon de ; réception

**Indicateurs :** Si le microprogramme n'offre aucune prise en charge du mode IP Open Data, l'indicateur d'erreur (E) est défini.

### 4.3.20 Permutation d'octets

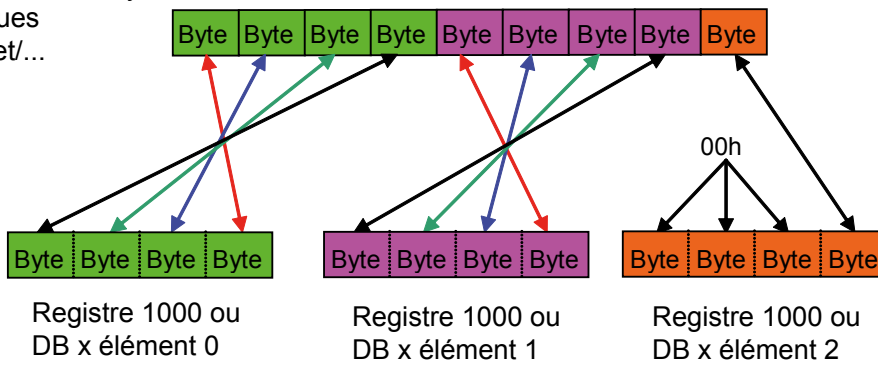
Exemple : Il y a 9 octets à envoyer/recevoir.  
 → Sans permutation des octets (SendData / ReceiveData) :

Données envoyées /reçues Octet/...



→ Avec permutation des octets (SendData / ReceiveData) :

Données envoyées /reçues Octet/...



Si le tampon contient du texte, les octets ne sont jamais inversés !



### 4.3.21 Décodage des adresses IP

L'adresse IP peut être saisie sous forme de valeur dans un texte, un registre ou une constante. Elle peut aussi désigner un nœud dans un registre ou une valeur constante. Une valeur constante peut seulement être un nœud.

#### **Adresse IP dans un texte :**

Dans ce cas, l'adresse IP est composée de 4 nombres décimaux séparés par un point, ex. : « 192.168.12.14 »

#### **Adresse IP dans un registre :**

L'adresse IP peut également être utilisée dans un registre. Si le mot de poids fort des 4 octets est nul, un nœud est alors représenté. Si le mot de poids fort des 4 octets est différent de zéro, le registre code alors une adresse et est composé de 4 nombres hexadécimaux :

aa	bb	cc	dd
----	----	----	----

L'adresse IP adopte ce format dans les informations de réception et de connexion.

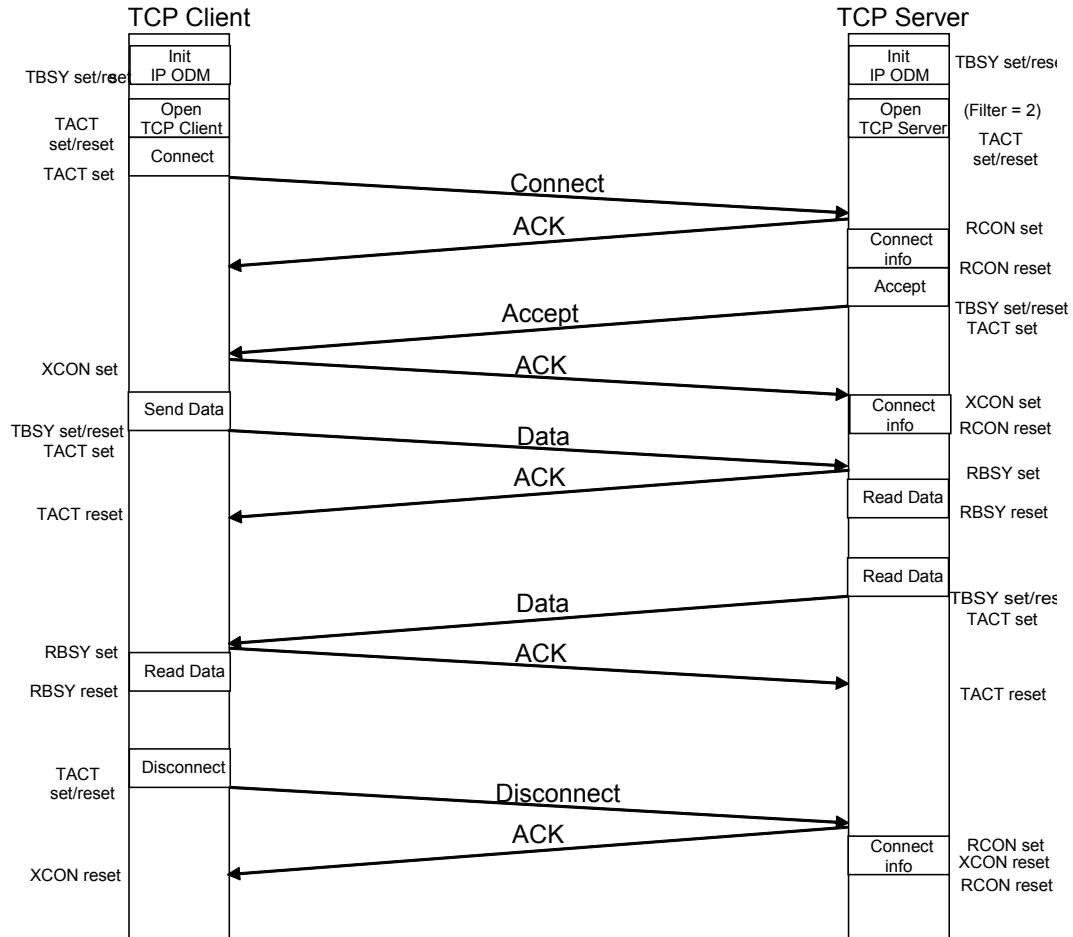
Exemple : 0C0A80C0Eh pour l'adresse IP 192.168.12.14

#### **Adresse IP dans une constante :**

Dans ce cas, l'adresse IP code toujours le numéro de nœud.

### 4.3.22 Flux de connexion TCP type

Connexion d'un client TCP (172.16.1.142) sur un serveur TCP (172.16.1.141) et envoi de deux télégrammes.



4

Ou dans l'affichage d'un analyseur Ethernet (dans ce cas, Wireshark). Seul un télégramme est envoyé.

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.1.142	172.16.1.141	TCP	5050 > 5050 [SYN] Seq=0 Ack=0 win=1840 Len=0 MSS=1460
2	0.001275	172.16.1.141	172.16.1.142	TCP	5050 > 5050 [SYN, ACK] Seq=0 Ack=1 win=720 Len=0 MSS=1460
3	0.002324	172.16.1.142	172.16.1.141	TCP	5050 > 5050 [ACK] Seq=1 Ack=1 win=720 Len=0
4	8.359572	172.16.1.142	172.16.1.141	TCP	5050 > 5050 [PSH, ACK] Seq=1 Ack=1 win=720 Len=4
5	8.733767	172.16.1.141	172.16.1.142	TCP	5050 > 5050 [ACK] Seq=1 Ack=5 win=720 Len=0
6	17.921981	172.16.1.142	172.16.1.141	TCP	5050 > 5050 [FIN, ACK] Seq=5 Ack=1 win=720 Len=0
7	17.923051	172.16.1.141	172.16.1.142	TCP	5050 > 5050 [ACK] Seq=1 Ack=6 win=720 Len=0
8	17.924448	172.16.1.141	172.16.1.142	TCP	5050 > 5050 [FIN, ACK] Seq=1 Ack=6 win=720 Len=0
9	17.925515	172.16.1.142	172.16.1.141	TCP	5050 > 5050 [ACK] Seq=6 Ack=2 win=720 Len=0

## 4.4 CSF supplémentaires

Comme déjà indiqué dans le mode Open Data, il existe pour des fonctionnalités spéciales des « Call System Functions » en relation avec le PCD7.F65x. Voici maintenant l'énumération des CSF supplémentaires.

### 4.4.1 CSF NA-Reset

Ce CSF génère une réinitialisation matérielle du PCD7.F65x. Ceci est semblable à une mise hors/sous tension du PCD7.F65x.

CSF	[cc]	S.IP.Library	; Bibliothèque IP
		S.IP.NAReset	; Fonction NAReset
	←	Status	; Statut du PCD7.F65x (R)

4

Le statut peut avoir les valeurs suivantes :

- 0 = pendant l'exécution de la réinitialisation
- 1 = la réinitialisation est exécutée et le PCD7.F65x est à nouveau prêt
- 2 = le PCD7.F65x n'est pas présent, aucune signature disponible
- 3 = le statut du registre est en dehors des limites de registre
- 4 = temporisation dans la boîte de réception
- 5 = le microprogramme est trop ancien / non compatible.

### 4.4.2 CSF SetLocalIPNode

Ce CSF configure le PCD7.F65x sur une nouvelle configuration IP

CSF	[cc]	S.IP.Library	; Bibliothèque IP
		S.IP.SetLocalIPNode	; Fonction SetLocalIPNode
	→	IP_Address	; Nouvelle adresse IP du PCD7.F65x (R)
	→	Subnet_Mask	; Nouveau masque de sous-réseau du PCD7.F65x (R)
	→	Default_Gateway	; Nouvelle passerelle par défaut du PCD7.F65x (R)

La nouvelle adresse IP est actualisée sur le PCD7.F650/1/2 sans mise hors/sous tension. Pour pouvoir travailler avec un nouveau masque de sous-réseau et une nouvelle passerelle par défaut, le PCD7.F650/1/2 doit être redémarré.

Les trois paramètres peuvent être modifiés et actualisés pendant le fonctionnement sur le PCD7.F655.

### 4.4.3 CSF IPPhyConfig

Ce CSF configure la couche physique Ethernet sur le PCD7.F65x.

CSF [cc]	S.IP.Library	; Bibliothèque IP
	S.IP.IPPhyConfig	; Fonction IPPhyConfig
→	Autonegotiate	; Si mode d'auto-négociation, TRUE/FALSE (R)
→	Speed	; Vitesse de communication, 10/100 Mb/s (R)
→	Duplex_mode	; Mode duplex, duplex intégral/semi-duplex (R)
←	Status	; Statut du message d'erreur (R)

4

Il est possible de modifier la configuration de la couche physique sur le PCD7.F65x avec un CSF. En règle générale, la couche physique fonctionne en mode auto-négociation. Cela signifie qu'elle peut fonctionner en mode 10Mb/s ou 100Mb/s, semi-duplex ou duplex intégral.

Les configurations possibles de la couche physique sont soit le mode auto-négociation ou les modes avec une vitesse de communication fixe et un mode duplex défini. Le CSF retourne un statut.

Autonegotiate :	"1"	pour définir la couche physique en mode auto-négociation
	"0"	pour ne pas travailler en mode auto-négociation
Speed :	"1"	pour définir la couche physique sur 10Mb/s
	"2"	pour définir la couche physique sur 100Mb/s
	"3"	pour que la couche physique recherche elle-même le mode de communication adaptée (10 ou 100Mb/s). Est uniquement autorisé en mode auto-négociation.
Duplex Mode	"1"	pour définir la couche physique sur le mode semi-duplex
	"2"	pour définir la couche physique sur le mode duplex intégral
	"3"	3 pour que la couche physique recherche elle-même le mode de communication approprié (semi-duplex ou duplex intégral). Est uniquement autorisé en mode auto-négociation
Error Status	"0"	aucune erreur n'est survenue
	"1"	aucune erreur n'est survenue <ul style="list-style-type: none"> <li>• en mode auto-négociation : L'auto-négociation a échoué</li> <li>• hors mode auto-négociation : il n'a pas été possible de définir la couche physique dans la configuration correspondante.</li> </ul>

**Paramètre « Autonegotiate » = 1**

Lorsque le paramètre Autonegotiate est défini, il faut signaler alors à la couche physique les vitesses et modes de communication possibles qu'il doit prendre en charge. Son « Advertisement Capability Register » est ainsi configuré.

Le contenu de la boîte indique la configuration du « Advertisement Capability Register » et définit les modes de communication qui sont autorisés		Valeur du paramètre pour la vitesse de communication		
		1	2	3
Valeur du paramètre pour le mode duplex	1	10 Mb/s, semi-duplex, fixe	100 Mb/s, semi-duplex, fixe	10 Mb/s + 100 Mb/s, semi-duplex autorisé
	2	10 Mb/s, duplex intégral, fixe	100 Mb/s, duplex intégral, fixe	10 Mb/s + 100 Mb/s, duplex intégral autorisé
	3	10 Mb/s, semi-duplex + duplex intégral autorisés	100 Mb/s, semi-duplex + duplex intégral autorisés	10 Mb/s + 100 Mb/s, semi-duplex + duplex intégral autorisés

4

Après avoir défini la couche physique par CSF en mode auto-négociation, le PCD7.F65x effectue un cycle d'auto-négociation avec la station sur l'autre extrémité du câble pour installer correctement la liaison. Si la station à l'autre extrémité du câble ne prend pas en charge l'auto-négociation, la communication n'est plus sûre et stable et des problèmes peuvent survenir. C'est pourquoi le CSF retourne alors un message d'erreur avec le « statut ». Si la station sur l'autre extrémité du câble prend cependant en charge l'auto-négociation, les deux appareils négocient alors le meilleur mode de communication possible. Après une perte de la liaison physique et une nouvelle liaison (si le PCD7.F65x est connecté par exemple à un autre port du commutateur), un nouveau cycle d'auto-négociation est redémarré.

**Paramètre « Autonegotiate » = 0**

Si le paramètre Autonegotiate n'est pas défini, la couche physique doit alors être définie dans une configuration claire et fixe. Seuls les modes 10 Mb/s ou 100Mb/s et semi-duplex **ou** duplex intégral peuvent être définis.

Le contenu de la boîte indique la configuration du « Advertisement Capability Register » et définit les modes de communication qui sont autorisés		Valeur du paramètre pour la vitesse de communication		
		1	2	3
Valeur du paramètre pour le mode duplex	1	10 Mb/s, semi-duplex, fixe	100 Mb/s, semi-duplex, fixe	non autorisé
	2	10 Mb/s, duplex intégral, fixe	100 Mb/s, duplex intégral, fixe	non autorisé
	3	non autorisé	non autorisé	non autorisé

Après avoir défini la couche physique par CSF dans un mode fixe, le PCD7.F65x n'effectue aucun cycle d'auto-négociation avec la station à l'autre extrémité du câble. Il est toutefois indispensable que la station à l'autre extrémité du câble possède la même configuration que celle de la couche physique locale. Aucun mode de communication stable ne peut alors être garanti.

#### 4.4.4 CSF SendEtherSBUS

Ce CSF équivaut à une instruction STXM (envoi de supports). Au lieu du nœud IP, l'adresse IP de la station qui doit recevoir les supports est directement saisie. Envoie à une station esclave les éléments sources du PCD (maître).

CSF [cc]	S.IP.Library	; Bibliothèque IP
	S.IP.SendEtherSBUS	; Fonction SendEtherSBUS
→	Channel	; n° de canal (R/K)
→	RemotelP	; Adresse IP distante pour l'envoi ; de données (R/K)*
→	SBUSAddr	; Adresse S-Bus distante (R)
→	Count	; Nombre d'éléments
→	Source	; Source I/O/F/R/T/C/DB/K
→	Destination	; Destination I/O/F/R/T/C/DB/K
←	Status	; Statut du message d'erreur (R)
	* voir chapitre : Décodage des adresses IP Count	1...32
	Nombre d'éléments R/T/C qui sont envoyés	1...128
	Nombre d'éléments I/O/F qui sont envoyés	
	0	Code de fonction spécial. Les consignes dans le « groupe d'instructions pour la famille PCD » ou dans l'aide PG5 doivent être observées
source	0...8191	Adresse de base des éléments I/O/F dans le PCD maître
	0...4095	Adresse de base des éléments R dans le PCD maître
	0...1599	Adresse de base des éléments R dans le PCD maître
	0...7999	Adresse de base des éléments R dans le PCD maître
	4000	K, code de fonction spécial. Les consignes dans le « groupe d'instructions pour la famille PCD » ou dans l'aide PG5 doivent être observées
Destination	0...8191	Adresse de base des éléments I/O/F dans le PCD esclave
	0...4095	Adresse de base des éléments R dans le PCD maître
	0...1599	Adresse de base des éléments R dans le PCD maître
	0...7999	Adresse de base des éléments R dans le PCD maître
	1000	K, horodatage dans le PCD esclave
	17, 18, 19	K, code de fonction spécial. Les consignes dans le « groupe d'instructions pour la famille PCD » ou dans l'aide PG5 doivent être observées
Status	0...4095	Statut du CSF dans R 0 = aucune erreur 1 = erreur d'utilisation du canal ou module IP non présent. 2 = maître non assigné 3 = erreur de conversion de l'adresse IP 4 = ce type de diffusion n'est pas autorisé 5 = erreur de support

**Exemple :**

```

$INCLUDE SNetLib.inc ; Intégration de la bibliothèque SNET utilisée
SASI 9 ; SASI maître comme pour le S-Bus traditionnel avec
; diagnostic
SASI_Master ; Texte avec indicateurs de diagnostic et registre
LD R 10 ; Adresse IP distante = 192.168.12.95
0C0A80C5CH
LD R 20 ; Adresse S-Bus distante = 95
95

CSF S.SNET.Library
S.SNET.SendEtherSBUS
9 ; canal 9
R 10 ; Adresse IP distante
R 20 ; Adresse S-Bus distante
4 ; 4 éléments
R 0 ; A partir du registre 0...3 de la station maître
R 0 ; Sur le registre 0...3 sur 192.168.12.95, S-Bus 95
R 30 ; statut
    
```



Le tableau suivant indique les éléments de la station source locale sur laquelle peuvent être copiés les éléments correspondants de la station de destination.

PCD esclave, destination								
PCD esclave, source		O	F	R	C	T	DB	Horloge
	I	x	x					
	O	x	x					
	F	x	x					
	R			x	x	x	x	x
	C			x	x	x	x	
	T			x	x	x	x	
	DB			x	x	x		

#### 4.4.5 CSF RecvEtherSBUS

Ce CSF équivaut à une instruction SRXM (réception de supports). Au lieu du nœud IP, l'adresse IP de la station à partir de laquelle les supports doivent être lus, est directement saisie. Lit à partir de la station partenaire les éléments source et les télécharge dans le PCD interne sur les éléments de destination.

CSF [cc]	S.IP.Library	; Bibliothèque IP
	S.IP.RecvEtherSBUS	; Fonction RecvEtherSBUS
→	Channel	; n° de canal (R/K)
→	RemotelIP	; Adresse IP distante pour la lecture ; de données (R/K)*
→	SBUSAddr	; Adresse S-Bus distante (R)
→	Count	; Nombre d'éléments
→	Source	; Source I/O/F/R/T/C/DB/K
→	Destination	; Destination I/O/F/R/T/C/DB/K
←	Status	; Statut du message d'erreur (R)

\* voir chapitre : Décodage des adresses IP

Count	1...32	Nombre d'éléments R/T/C qui sont envoyés
	1...128	Nombre d'éléments I/O/F qui sont envoyés
	0	Code de fonction spécial. Les consignes dans le « groupe d'instructions pour la famille PCD » ou dans l'aide PG5 doivent être observées
	R nnnn	Est utilisé pour le transfert de blocs de données
source	0...8191	Adresse de base des éléments I/O/F dans le PCD esclave
	0...4095	Adresse de base des éléments R dans le PCD maître
	0...1599	Adresse de base des éléments R dans le PCD maître
	0...7999	Adresse de base des éléments R dans le PCD maître
	6000	K, code de fonction spécial. Les consignes dans le « groupe d'instructions pour la famille PCD » ou dans l'aide PG5 doivent être observées
Destination	0...8191	Adresse de base des éléments I/O/F dans le PCD maître
	0...4095	Adresse de base des éléments R dans le PCD maître
	0...1599	Adresse de base des éléments R dans le PCD maître
	0...7999	Adresse de base des éléments R dans le PCD maître
Status	0...4095	Statut du CSF dans R
		0 = aucune erreur
		1 = erreur d'utilisation du canal ou module IP non présent.
		2 = maître non assigné
		3 = erreur de conversion de l'adresse IP
		4 = ce type de diffusion n'est pas autorisé
5 = erreur de support		

#### Exemple :

```

$INCLUDE SNetLib.inc ; Intégration de la bibliothèque SNET utilisée
  SASI 9 ; SASI maître comme pour le S-Bus traditionnel avec
; diagnostic
  SASI_Master ; Texte avec indicateurs de diagnostic et registre
  LD R 10 ; Adresse IP distante = 192.168.12.95
  0C0A80C5CH

```



```

LD   R 20           ; Adresse S-Bus distante = 95
95

CSF  S.SNET.Library
     S.SNET.RecvEtherSBUS
     9               ; canal 9
     R 10            ; Adresse IP distante
     R 20            ; Adresse S-Bus distante
     4               ; 4 éléments
     R 0             ; A partir du registre 0...3 de la station maître
     R 0             ; Sur le registre 0...3 sur 192.168.12.95, S-Bus 95
     R 30            ; statut
    
```

4

Le tableau suivant indique les éléments de la station source locale sur laquelle peuvent être copiés les éléments correspondants de la station de destination.

PCD maître, destination							
PCD esclave, source		O	F	R	C	T	DB
	I	x	x				
	O	x	x				
	F	x	x				
	R			x	x	x	x
	C			x	x	x	x
	T			x			x
	K						
	DB			x	x	x	

#### 4.5 Messages d'erreur Ethernet TCP/IP

Message	Description :	Solution
IPM NOT PRESENT	Il existe une configuration IP mais sans module IP.	Enficher le module IP dans l'emplacement configuré ou supprimer la configuration IP.
IPM DOES NOT RESTART	Le PCD a redémarré, mais le module IP ne répond pas.	Eteindre et rallumer le PCD.
IPM HAS OLD FW	Le microprogramme du module IP n'est pas compatible avec celui du PCD.	Mettre à jour le microprogramme du module IP.
IP FAIL SASITEXT	Le texte SASI est erroné	Vérifier le texte SASI.
IP FAIL SASI DBX	La configuration de la liste des nœuds (dbx) est erronée	Vérifier la configuration des nœuds
IP FAIL NO IPM	Une fonction IP a été exécutée, mais sans configuration ni module IP.	Attention : Dans la liste de nœuds, il doit y avoir au moins un nœud IP ! Configurer le module IP et l'insérer dans le PCD.

## 5 Diagnostic, dépietage des pannes et dépannage

### 5.1 Résumé

Toute démarche de dépietage des pannes et de dépannage comprend deux phases principales, la phase de diagnostic et la phase dans laquelle la solution est mise en œuvre. L'analyse du problème consiste à cerner progressivement les causes potentielles d'une erreur en posant des questions qui permettent d'y remédier. Il est important d'isoler les causes de toutes les erreurs.

### 5.2 Communication TCP/IP

La communication entre deux stations PCD dans un réseau peut être divisée en trois étapes. Si un télégramme est envoyé d'un PCD à l'autre, ces étapes se déroulent comme suit.

- Conversion de l'identificateur du nœud IP en adresse IP locale (configurée dans les paramètres matériels).
- Conversion de l'adresse IP de transmission en adresse MAC ( par la table ARP). Lors de l'envoi de messages de diffusion, l'adresse MAC est toujours utilisée au format suivant :

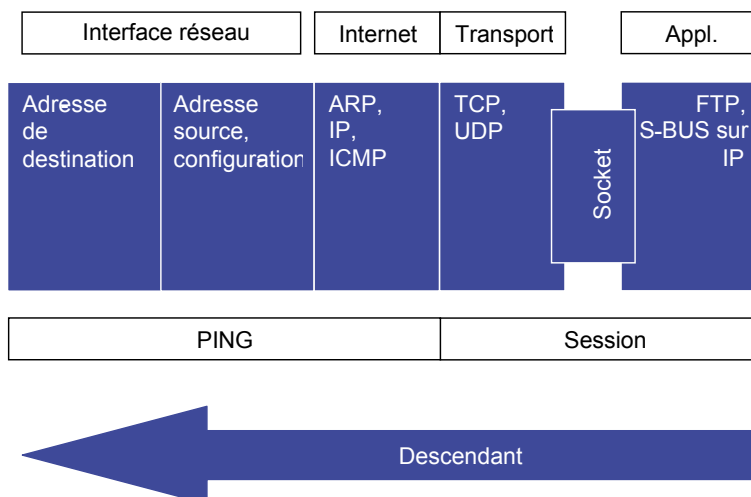
0xFF-FF-FF-FF-FF-FF

- Transmission du datagramme IP à l'adresse MAC fournie par ARP.

Chaque pile TCP/IP suit cette séquence d'un point à l'autre pour envoyer un télégramme.

### 5.3 Analyse descendante

Lors du dépietage des pannes dans TCP/IP, nous recommandons de débuter par le niveau supérieur (couche application), puis de descendre progressivement jusqu'aux couches IP inférieures. Le dépietage des pannes et le dépannage consistent à contrôler si les protocoles de chaque couche peuvent communiquer avec les couches supérieures et inférieures de cette couche.



Après avoir configuré le module TCP/IP dans les paramètres matériels du PG5 (adresse IP, masque de sous-réseau, passerelle par défaut, numéro de station (adresse S-Bus) et noeud IP), le dépiage des pannes se fait en deux étapes :

1) Etablir une communication entre un client PG5 et un esclave PCD.

Connexion du débogueur en ligne du PG5 :

- Dans les paramètres en ligne du PG5, sélectionner le type de canal IP SOCKET.
- Modifier les paramètres (si nécessaire)
- Mettre en ligne le débogueur en ligne du PG5.

Si cela fonctionne, une communication est établie entre la couche interfaces réseau et la couche application. Tous les protocoles intermédiaires nécessaires fonctionnent correctement.

Si cela ne fonctionne pas, la communication entre la couche interfaces réseau et la couche Internet doit être contrôlée.

2) Suivre l'analyse descendante et vérifier avec l'instruction PING si l'ARP fonctionne correctement, à savoir si l'adresse IP est convertie en adresse MAC pour chaque requête écho et chaque réponse à une requête écho.

5

### 5.3.1 PING

Si une instruction PING (requête écho) réussit, cela signifie que TCP/IP est correctement installé sur l'ordinateur hôte et sur le module Ethernet TCP/IP et que la liste de caches ARP contient les stations PCD.

Exemples :

PING 127.0.0.1	« Look up address » vérifie si la pile TCP/IP est correctement installée sur l'ordinateur hôte. La requête écho reste dans la pile TCP/IP locale.
PING 192.168.12.60	Adresse IP propre : Vérifie si la station a été correctement saisie dans la table des caches (utilisée pour convertir l'adresse IP en adresse MAC).
PING 192.168.12.60	Adresse IP de l'unité PCD distante : Vérifie si la pile TCP/IP (y compris le cache ARP) a été correctement installée sur la station PCD distante.
PING 192.168.12.60	L'adresse IP d'une station PCD est uniquement accessible via un routeur : Vérifie si l'adresse de passerelle par défaut configurée correspond.

PING <adresse IP> -t	Envoi en continu des requêtes écho
PING <adresse IP> -n 10	Envoie 10 requêtes écho successives
PING <adresse IP> -L 320	Envoie une requête de 320 octets de long

```
C:\WINNT\system32>ping 192.168.12.60
Pinging 192.168.12.60 with 32 bytes of data:
Reply from 192.168.12.60: bytes=32 time<10ms TTL=30
Reply from 192.168.12.60: bytes=32 time<10ms TTL=30
Reply from 192.168.12.60: bytes=32 time<10ms TTL=30
Reply from 192.168.12.60: bytes=32 time<10ms TTL=30
```

Time	Délai de réponse à la requête écho
TTL	Durée de vie (Time-to-live) : ce nombre est décrémenté à chaque passage de commutateur ou routeur.  Lorsque le nombre est nul, le télégramme n'est plus envoyé.

Si une instruction PING échoue, la liste de caches ARP doit être vérifiée pour savoir si elle contient la station PCD et si la conversion d'adresse est possible

(adresse IP en adresse MAC).

5

### 5.3.2 ARP

L'instruction ARP énumère le contenu de la table des caches ARP sur l'ordinateur hôte local. Une entrée ARP dans cette table contient les adresses IP et MAC distantes.

Si deux stations dans le même sous-réseau n'arrivent pas à échanger des instructions PING, il faut vérifier si la station figure dans la table des caches ARP.

ARP -a	Affiche toutes les entrées dans le cache ARP
ARP -d <adresse IP>	Supprime une entrée dans le cache ARP.

```
C:\WINNT\system32>arp -a

Interface: 192.168.12.46 on Interface 2
Internet Address      Physical Address      Type
192.168.12.60         00-50-c2-0c-50-24    dynamic
192.168.12.200        00-10-83-f9-1d-fe    dynamic
192.168.12.202        00-30-6e-00-03-50    dynamic
192.168.12.221        00-80-3e-7f-15-fa    dynamic
```

### 5.3.3 Autres instructions utiles pour l'ordinateur hôte

IPCONFIG Affiche les paramètres de configuration TCP/IP de l'ordinateur hôte

```
C:\WINNT\system32>ipconfig

Windows NT IP Configuration

Ethernet adapter E100B1:

    IP Address. . . . . : 192.168.12.46
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.12.221

C:\WINNT\system32>
```

```

C:\WINNT\system32>ipconfig/all

Windows NT IP Configuration

    Host Name . . . . . : ch01w234.saia-burgess.ch
    DNS Servers . . . . . : 192.168.6.204
                          192.168.6.28
    Node Type . . . . . : Hybrid
    NetBIOS Scope ID. . . . . :
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No
    NetBIOS Resolution Uses DNS : No

Ethernet adapter E100B1:

    Description . . . . . : Intel(R) PRO Adapter
    Physical Address. . . . . : 00-90-27-BC-ED-8A
    DHCP Enabled. . . . . : No
    IP Address. . . . . : 192.168.12.46
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.12.221
    Primary WINS Server . . . . . : 192.168.6.204
    Secondary WINS Server . . . . . : 192.168.6.209

```

5

**NETSTAT** Affiche les statistiques et protocoles des connexions TCP/IP disponibles sur l'ordinateur hôte

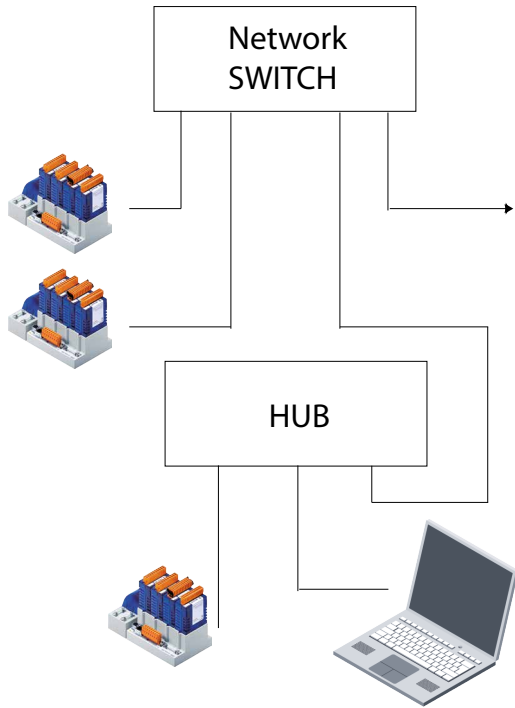
NETSTAT -a	Affiche l'ensemble des connexions et ports actifs sous forme de liste
NETSTAT -r	Affiche le contenu de la table de routage sous forme de liste
NETSTAT -p UDP	Affiche une liste avec toutes les connexions UDP actives
NETSTAT -p TCP	Affiche une liste avec toutes les connexions TCP actives
NETSTAT -p ICMP	Affiche une liste avec toutes les connexions ICMP actives

## 5.4 Analyseur de protocole Ethernet - Wireshark

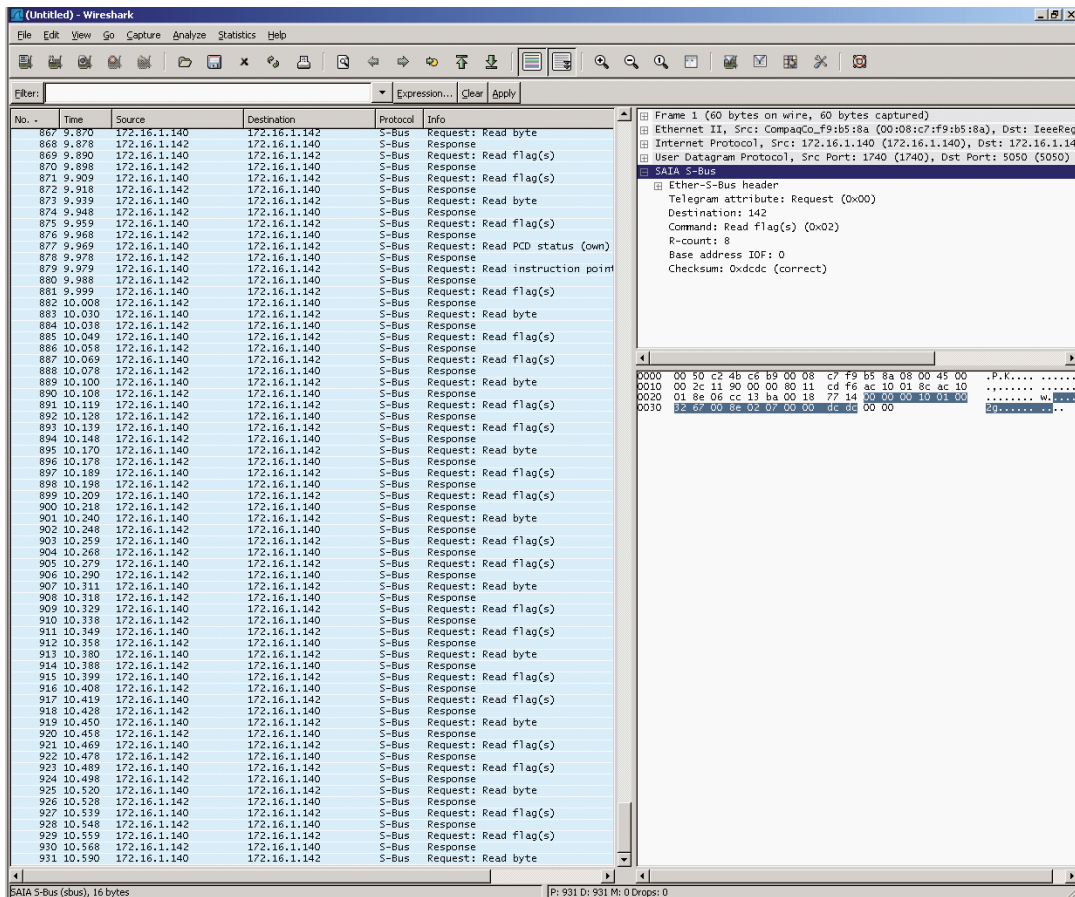
Pour analyser la communication Ethernet S-Bus propriétaire et la communication Ethernet générale, nous recommandons d'utiliser le logiciel libre Wireshark ([www.wireshark.org](http://www.wireshark.org)).

Il s'agit d'un analyseur libre avec journaliseur de cartes Ethernet, WinPcap. Nous pouvons analyser et interpréter les enregistrements Ethernet effectués avec ce programme. L'analyseur Wireshark, à partir de la version 1.1, interprète également les télégrammes Ethernet S-Bus. Pour de plus amples informations à ce sujet, voir les FAQ correspondantes #100569.

Afin d'enregistrer le plus possible toute la communication d'une station concernée, un concentrateur est installé en aval. L'analyseur accède alors à ce concentrateur. Les commutateurs canalisent la communication et ne sont pas appropriés pour les analyses et enregistrements.



Il s'agit d'un télégramme S-Bus UDP interprété (indicateurs de couleur rouge), enregistré dans Wireshark



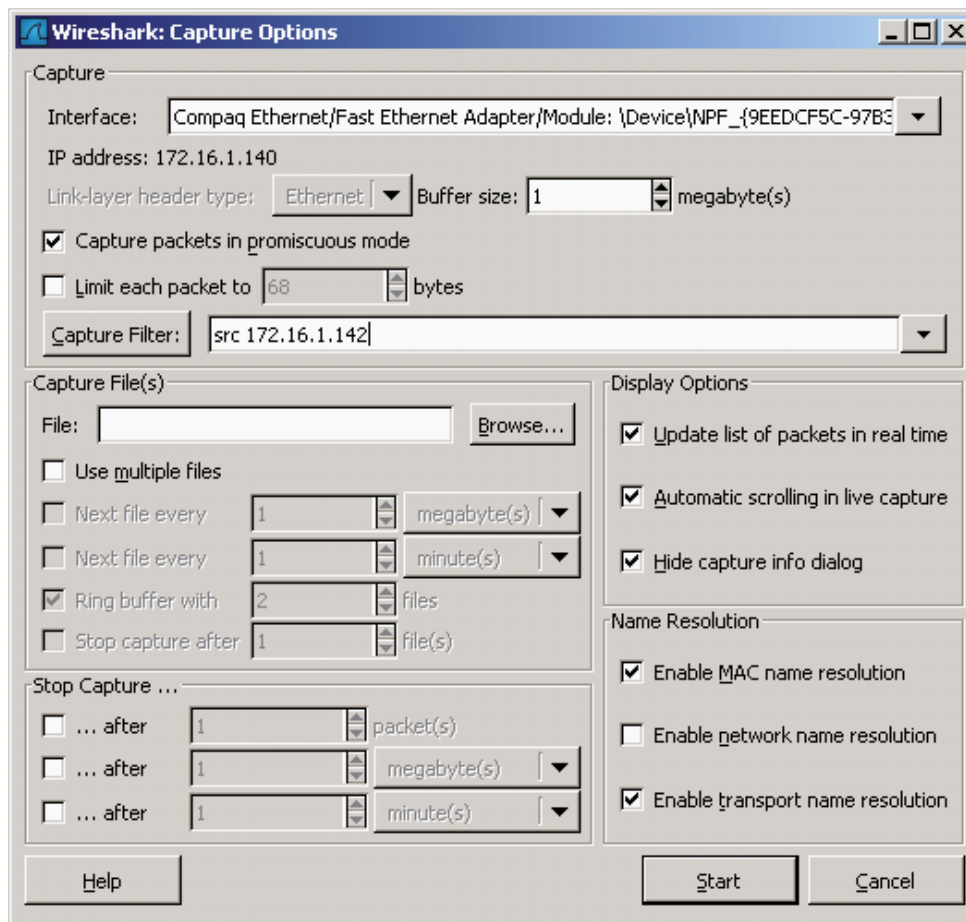
D'autres protocoles, ex. : ModBus TCP ou ModBus UDP, sont également interprétés et affichés.

Wireshark offre des fonctions de filtre très utiles. Elles peuvent être utilisées directement lors de la journalisation des données, ou ultérieurement, lors de l'analyse. Il est souvent conseillé d'enregistrer la communication sans « Capture Filter » et d'utiliser ultérieurement le filtre, comme « Display Filter », lors de l'analyse.

Voici quelques exemples :

« **Capture Filter** »-seuls les télégrammes qui remplissent la condition de filtre, sont également enregistrés. Le Capture Filter est accessible à partir de « Capture → Options... » ou avec la combinaison de touches « Ctrl+K ». Avec le paramétrage ci-après dans la figure, tous les télégrammes en partance vers ou au départ de l'adresse IP 172.16.1.142 sont enregistrés.

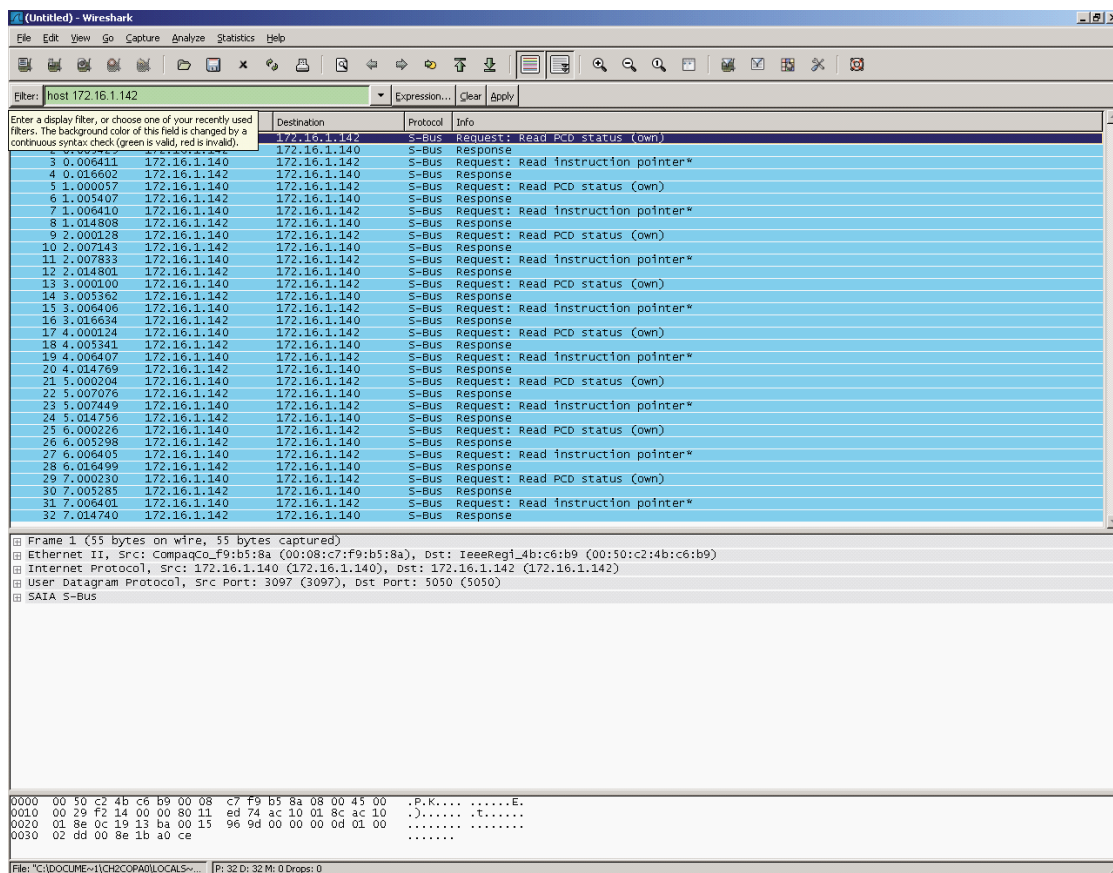
5



Exemple d'autres filtres d'enregistrement :

Hôte 172.18.5.4	enregistre uniquement le trafic au départ de ou vers l'adresse IP 172.18.5.4
Port 5050	enregistre le trafic via le port 5050
SRC 172.18.5.4	enregistre tous les télégrammes au départ de cette adresse IP
DST Port 135 et TCP Port 135	enregistre le trafic sur le port de destination 135 dans TCP

« **Display Filters** »-ce filtre est uniquement utilisé à l'écran et ne modifie pas le contenu du fichier journal. Il s'agit de filtres de tri. Le Display-Filter est directement définie dans la fenêtre principale.



5

Exemples de Display-Filter :

- Udp affiche tout le trafic UDP
- Tcp affiche tout le trafic TCP
- Host 172.18.5.4 affiche le trafic de l'adresse IP 172.18.5.4
- Host 172.18.5.4 et Port 5050 affiche le trafic de l'adresse IP 172.18.5.4 via le port 5050
- Liens avec des opérateurs de comparaison :
- tp.addr == 172.18.5.4 affiche le trafic de l'adresse IP 172.18.5.4
- tp.addr != 172.18.5.4 affiche tout le trafic hormis celui via l'adresse IP 172.18.5.4
- !(ip.addr == 172.18.5.4) affiche tout le trafic hormis celui via l'adresse IP 172.18.5.4
- ip.src == 172.18.5.4 and ip.dst == 172.18.5.5 affiche tous les télégrammes de 172.18.5.4 vers 172.18.5.5
- Frame.pkt\_len < 128 affiche tous les paquets inférieurs à 128 octets
- Tcp.port == 25 or icmp affiche tous les télégrammes via le port TCP 25 et les télégrammes ICMP
- Tcp.window\_size == 0 && tcp.flags.reset != 1 TCP Window Size est nul. Le tampon est plein



Les comparaisons suivantes sont autorisées :

eq	==	and	&&
ne	!=	or	
gt	>	xor	^^
lt	<	not	!
ge	>=		
le	<=		

Les opérations booléennes comme pour les six bits de contrôle TCP, sont également autorisées. Exemple de condition : tcp.flags.syn

- URG: Urgent Pointer field significant
- ACK: Acknowledgement field significant
- PSH: Push Function (envoi des données à partir de la pile)
- RST: Reset the connection (ex. : réponse lorsqu'un télégramme arrive sur un port qui n'est pas ouvert)

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.12.92	Broadcast	ARP	who has 192.168.12.93? Tell 192.168.12.92
2	0.000379	IeeeRegI_0c:54:0c	192.168.12.92	ARP	192.168.12.93 is at 00:50:c2:0c:54:0c
3	0.001539	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [SYN] Seq=1603176 Ack=0 win=720 Len=0 MSS=1458
4	0.002056	192.168.12.93	192.168.12.92	TCP	6000 > 6000 [RST, ACK] Seq=0 Ack=1603177 win=0 Len=0
5	1.008559	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [SYN] Seq=1795557 Ack=0 win=720 Len=0 MSS=1458
6	1.009080	192.168.12.93	192.168.12.92	TCP	6000 > 6000 [RST, ACK] Seq=0 Ack=1795558 win=0 Len=0
7	2.018454	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [SYN] Seq=1987938 Ack=0 win=720 Len=0 MSS=1458
8	2.018962	192.168.12.93	192.168.12.92	TCP	6000 > 6000 [RST, ACK] Seq=0 Ack=1987939 win=0 Len=0
9	3.028428	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [SYN] Seq=2180319 Ack=0 win=720 Len=0 MSS=1458
10	3.028932	192.168.12.93	192.168.12.92	TCP	6000 > 6000 [RST, ACK] Seq=0 Ack=2180320 win=0 Len=0
11	4.038395	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [SYN] Seq=2372700 Ack=0 win=720 Len=0 MSS=1458
12	4.038904	192.168.12.93	192.168.12.92	TCP	6000 > 6000 [RST, ACK] Seq=0 Ack=2372701 win=0 Len=0
13	5.048493	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [SYN] Seq=2565081 Ack=0 win=720 Len=0 MSS=1458
14	5.048993	192.168.12.93	192.168.12.92	TCP	6000 > 6000 [RST, ACK] Seq=0 Ack=2565082 win=0 Len=0
15	6.058333	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [SYN] Seq=2757462 Ack=0 win=720 Len=0 MSS=1458
16	6.058841	192.168.12.93	192.168.12.92	TCP	6000 > 6000 [RST, ACK] Seq=0 Ack=2757463 win=0 Len=0
17	7.068332	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [SYN] Seq=2949843 Ack=0 win=720 Len=0 MSS=1458
18	7.068844	192.168.12.93	192.168.12.92	TCP	6000 > 6000 [RST, ACK] Seq=0 Ack=2949844 win=0 Len=0

SYN: Synchronize sequence numbers (début d'une connexion TCP)

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.12.92	Broadcast	ARP	who has 192.168.12.93? Tell 192.168.12.92
2	0.000408	IeeeRegI_0c:54:0c	192.168.12.92	ARP	192.168.12.93 is at 00:50:c2:0c:54:0c
3	0.001583	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [SYN] Seq=769525 Ack=0 win=720 Len=0 MSS=1458
4	0.002373	192.168.12.93	192.168.12.92	TCP	6000 > 6000 [SYN, ACK] Seq=384001 Ack=769526 win=720 Len=0 MSS=1458
5	0.004847	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [ACK] Seq=769526 Ack=384002 win=720 Len=0

FIN: No more data from sender (fin d'une connexion TCP)

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [FIN, ACK] Seq=833653 Ack=768002 win=720 Len=0
2	0.000609	192.168.12.93	192.168.12.92	TCP	6000 > 6000 [ACK] Seq=768002 Ack=833654 win=720 Len=0
3	0.001954	192.168.12.93	192.168.12.92	TCP	6000 > 6000 [FIN, ACK] Seq=768002 Ack=833654 win=720 Len=0
4	0.003014	192.168.12.92	192.168.12.93	TCP	6000 > 6000 [ACK] Seq=833654 Ack=768003 win=720 Len=0

Les FAQ 100535 fournissent de plus amples informations sur les fonctions de filtre.

### 5.5 Débogage de pile IP du PCD7.F655 via RS-232 du Saia PCD®

La pile TCP/IP d'un PCD7.F655 peut être déboguée via l'interface série RS-232 du PCD. Les statistiques du trafic Ethernet peuvent, par exemple, être analysées, la table ARP du PCD7.F655 peut être consultée ou les sockets ouvertes peuvent être étudiées. Les détails relatifs aux différentes fonctions sont énumérés ci-après.

Le PCD7.F655 peut être débogué via les ports PCD suivants :

PCD...	PCD7.F655 sur...	Débogage via...
PCD2.M150	Slot B1	Port 2 (Pin 30...34)
PCD2.M170	Slot B2	Port 4 (Pin 40...44)
PCD4.M170	Slot B2	Port 4 (Pin 40...44)
PCD2.M480	Slot B1, Slot B2	Port 2 (Pin 30...34) Port 4 (Pin 40...44)

5



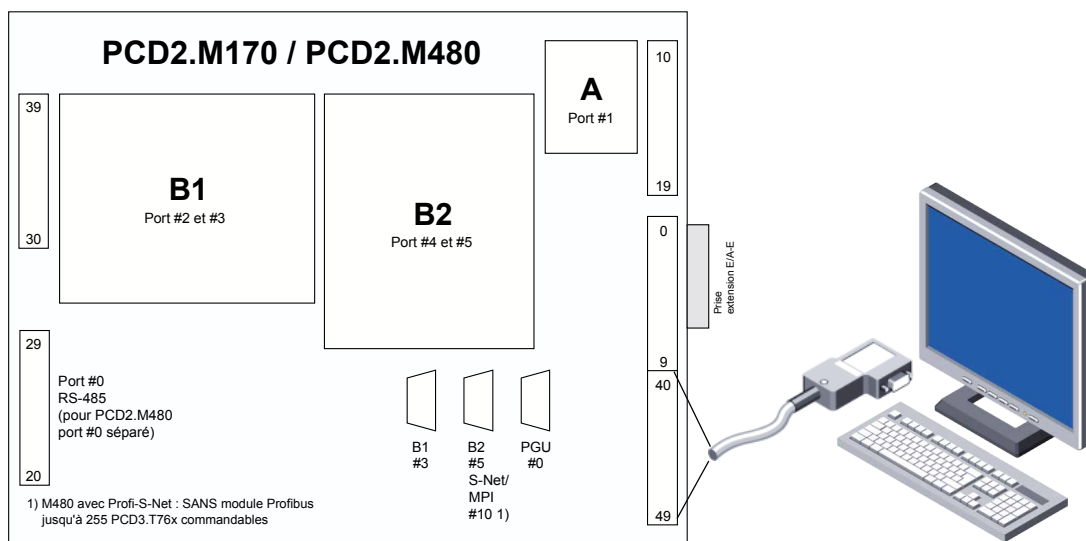
Le PCD7.F655 sur un PCD1.M13x ne peut pas être débogué en raison de l'absence d'une interface série.

Connectez l'interface série du PCD à un câble simulateur de modem (câbles utilisés uniquement : GND, TxD et RxD) sur votre PC et ouvrez, par exemple, une liaison hyperterminal avec les paramètres suivants :

- Bits per second: 115'200
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Flow Control: None

Après avoir validé en appuyant sur la touche Enter, l'invite « > » du shell apparaît. Vous pouvez maintenant débiter les instructions de débogage via le shell.

Dans l'exemple ci-après, le PCD7.F655 est situé dans le logement B2 d'un PCD2.M170. Reliez également le port 4 (broches 40...44) du PCD2.M170 à votre PC pour démarrer le shell de communication.



Au début du shell, toutes les fonctions prises en charge pour le débogage de la pile TCP/IP peuvent être énumérées avec « help ».

```

NexGen - Hyper Terminal
File Edit View Call Transfer Help
>help
Supported commands:
help          ver          exit          lsmmod
date          debug        devstat       netstat
ifconfig      arp          route         ping
sap
>_
Connected 00:00:06  Auto detect  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo

```

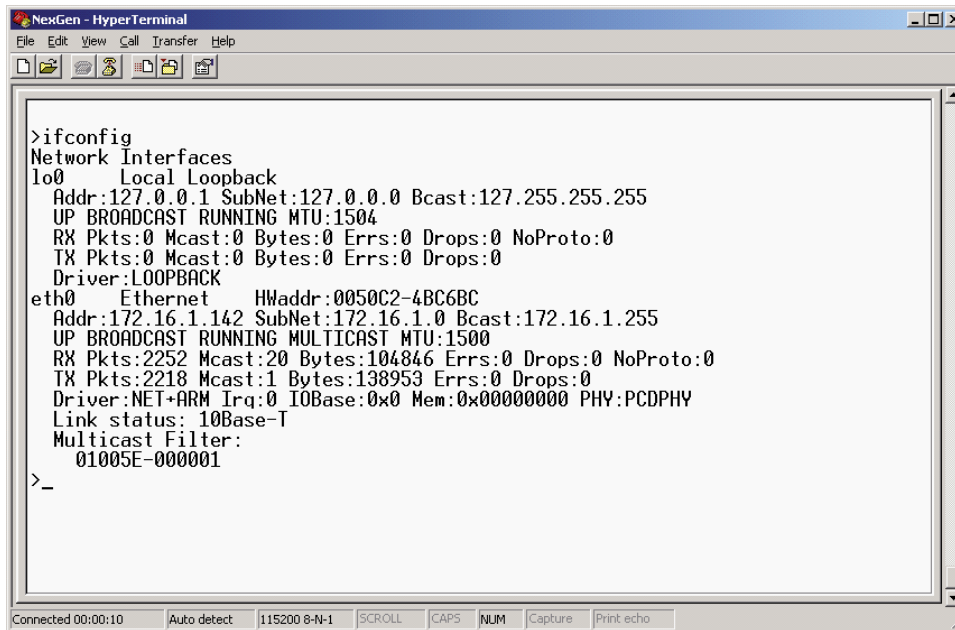
5

- « **ver** »      Version de la pile
- « **netstat** »    Connexions réseau actives
  - a
  - s      Statistique précise via les télégrammes IP, ICMP, IGMP, TCP et UDP. Les télégrammes manquants peuvent être définis ici.
  - b      Gestion de la mémoire disponible sur le PCD7.F655
- « **ifconfig** »    Interface Ethernet de la station
- « **arp** »        Table ARP de la station
- « **ping** »        Envoi d'un ping de la station distante
- « **sap** »        sockets qui sont ouvertes sur le module
  - PCD1 : **toujours** Sap 16 pour le port S-Bus UDP 5050
  - Pour tous les autres PCD compatibles IP, toujours Sap 32 pour le port S-Bus UDP 5050
- help <Instruction>** Affiche les arguments d'appel possibles de la fonction.
  - ex. : « >help netstat »

Ci-après, quelques exemples d'utilisation de la fonction de débogage pendant une communication en cours.

L'instruction shell « **ifconfig** » affiche l'interface Ethernet du PCD7.F655. L'adresse IP est dans ce cas 172.16.1.142, le sous-réseau 172.16.1.0 et l'adresse de diffusion est 172.16.1.255 pour le réseau configuré.

L'adresse MAC est également visible (00 50 C2 4B C6 BC) de même qu'une statistique approximative du nombre de télégrammes entrés et sortis.



```

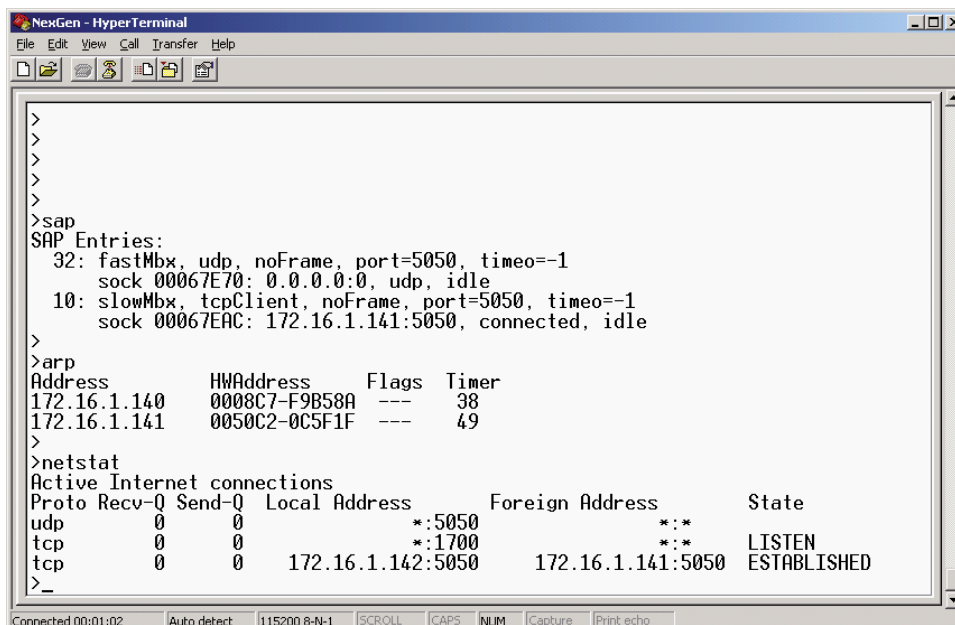
NexGen - HyperTerminal
File Edit View Call Transfer Help
>ifconfig
Network Interfaces
lo0    Local Loopback
  Addr:127.0.0.1 SubNet:127.0.0.0 Bcast:127.255.255.255
  UP BROADCAST RUNNING MTU:1504
  RX Pkts:0 Mcast:0 Bytes:0 Errs:0 Drops:0 NoProto:0
  TX Pkts:0 Mcast:0 Bytes:0 Errs:0 Drops:0
  Driver:LOOPBACK
eth0   Ethernet      HWaddr:0050C2-4BC6BC
  Addr:172.16.1.142 SubNet:172.16.1.0 Bcast:172.16.1.255
  UP BROADCAST RUNNING MULTICAST MTU:1500
  RX Pkts:2252 Mcast:20 Bytes:104846 Errs:0 Drops:0 NoProto:0
  TX Pkts:2218 Mcast:1 Bytes:138953 Errs:0 Drops:0
  Driver:NET-ARM Irq:0 IOBase:0x0 Mem:0x00000000 PHY:PCDPHY
  Link status: 10Base-T
  Multicast Filter:
    01005E-000001
>_
Connected 00:00:10  Auto detect  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo

```

L'instruction shell « **sap** » affiche les noeuds disponibles (Saps) qui sont ouverts sur le PCD7.F655. Le port UDP 5050 (S-Bus UDP9 est toujours le Sap 32 (sauf PCD1. M13x, Sap 16). Sur le Sap 10, une connexion client TCP est ouverte sur la station 172.16.1.141 sur le port le plus distant 5050. Le statut est « connecté/connected ». Ceci indique également que la communication UDP sur le port 5050 et la communication TCP sur le port 5050 fonctionnent simultanément.

L'instruction shell « **arp** » affiche la table ARP interne de la station locale. Dans l'exemple sont illustrés le PC sur lequel le débogueur en ligne du PG5 fonctionne (172.16.1.140) et la station distante (172.16.1.141) avec laquelle une connexion TCP existe sur le port 5050.

L'instruction shell « **netstat** » affiche les différentes sockets qui sont ouvertes sur la station locale. Il s'agit **toujours** de la socket UDP sur le port 5050 (S-Bus UDP) et de la socket TCP sur le port 1700 (socket du serveur FTP pour la mise à jour gratuite du PCD7.F655). Dans l'exemple, la socket client TCP préalablement déjà consultée est encore ouverte.



```

NexGen - HyperTerminal
File Edit View Call Transfer Help
>
>
>
>
>
>
>sap
SAP Entries:
  32: fastMbx, udp, noFrame, port=5050, timeo=-1
     sock 00067E70: 0.0.0.0:0, udp, idle
  10: slowMbx, tcpClient, noFrame, port=5050, timeo=-1
     sock 00067EAC: 172.16.1.141:5050, connected, idle
>
>arp
Address      HWAddress    Flags  Timer
172.16.1.140 0008C7-F9B58A ---    38
172.16.1.141 0050C2-0C5F1F ---    49
>
>netstat
Active Internet connections
Proto Recv-Q Send-Q Local Address      Foreign Address    State
udp      0      0          *:5050             *:.*
tcp      0      0          *:1700             *:.*               LISTEN
tcp      0      0 172.16.1.142:5050  172.16.1.141:5050 ESTABLISHED
>_
Connected 00:01:02  Auto detect  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo

```

L'instruction shell « **netstat -s** » affiche des statistiques précises sur les télégrammes IP, ICMP, IGMP, TCP et UDP. Les télégrammes manquants peuvent être définis ici.

```
>netstat -s
```

```
Ip:
```

```
8424 total packets received
0 with invalid header
0 with invalid address
0 forwarded
0 with unknown protocol
0 incoming packets discarded
8424 incoming packets delivered
8390 requests sent out
0 outgoing packets dropped
0 dropped because of missing route
0 reassemblies required
0 packets reassembled ok
0 packets reassembles failed
0 packets fragmented ok
0 fragments failed
0 fragments created
```

```
Icmp:
```

```
0 messages received
0 messages received with error
Input histogram:
0 messages sent
0 messages not sent
Output histogram:
```

```
Igmp:
```

```
0 messages received
0 messages received with error
0 membership queries received
0 membership reports received
0 membership reports received for our groups
0 membership reports sent
```

```
Tcp:
```

```
1 active opens
0 passive opens
0 embryonic connections dropped
0 established connections dropped
10 packets sent (0 retransmitted)
0 RESET packets sent
9 packets received
0 packets received with errors
0 duplicate ACKs
0 out-of-order packets
```

```
Udp:
```

```
8400 packets received
35 packets to unkown port received
0 packets received with errors
8380 packets sent
```

Exemple d'une application réelle composée de trois PCD (local 192.100.100.106 et distants 192.100.100.104 et 192.100.100.100) qui communiquent entre eux en mode S-Bus UDP (port 5050). Durée de l'enregistrement : env. une semaine.

```
>sap
SAP Entries:
    32:   fastMbx, udp, noFrame, port=5050, timeo=-1
        sock 000680C8: 0.0.0.0:0, udp, idle

>arp
Address          HWAddress          Flags          Timer
192.100.100.104  0050C2-4BC21D     ---           119
192.100.100.100  000D56-941547     ---           120

>ifconfig
Network Interfaces
loO  Local Loopback
    Addr:127.0.0.1 SubNet:127.0.0.0 Bcast:127.255.255.255
    UP BROADCAST RUNNING MTU:1504
    RX Pkts:0 Mcast:0 Bytes:0 Errs:0 Drops:0 NoProto:0
    TX Pkts:0 Mcast:0 Bytes:0 Errs:0 Drops:0
    Driver:LOOPBACK

ethO      Ethernet          HWaddr:0050C2-4BC6B4
    Addr:192.100.100.106 SubNet:192.100.100.0
    Bcast:192.100.100.255
    UP BROADCAST RUNNING MULTICAST MTU:1500
    RX Pkts:52921868 Mcast:81423 Bytes:2723134990 Errs:39 Drops:0
    NoProto:0
    TX Pkts:52846356 Mcast:15481 Bytes:2016432650 Errs:2 Drops:0
    Driver:NET+ARM lrq:0 IOBase:0x0 Mem:0x00000000 PHY:PCDPHY
    Link Status:  loOBase-TX
    Multicast Filter:
        01005E-000001

>netstat -b
Network message buffers
total:    126 free:    116, 15560/196056 bytes in use
Socket control blocks
total:    200 free:    198, 336/33600 bytes in use
TCP control blocks
total:    200 free:    199, 212/42400 bytes in use

>netstat -s
Ip:
52851798 total packets received
28 with invalid header
207 with invalid address
0 forwarded
0 with unknown protocol
0 incoming packets discarded
52851563 incoming packets delivered
53295668 requests sent out
0 outgoing packets dropped
0 dropped because of missing route
0 reassemblies required
0 packets reassembled ok
0 packets reassembles failed
0 packets fragmented ok
0 fragments failed
0 fragments created
```

```
Icmp:
  191684 messages received
  0 messages received with error
  Input histogram:
    echo reply: 4
    echo request: 191680
  191684 messages sent
  0 messages not sent
  Output histogram:
    echo reply: 191680
    echo request: 4

Icmp:
  0 messages received
  0 messages received with error
  0 membership queries received
  0 membership reports received
  0 membership reports received for our groups
  0 membership reports sent

Tcp:
  0 active opens
  0 passive opens
  0 embryonic connections dropped
  0 established connections dropped
  0 packets sent (0 retransmitted)
  0 RESET packets sent
  0 packets received
  0 packets received with errors
  0 duplicate ACKs
  0 out-of-order packets

Udp:
  52659652 packets received
  25391 packets to unknown port received
  0 packets received with errors
  53103984 packets sent
```

## 6 Exemples de programmation

### 6.1 Référence à des liens Ethernet

On a renoncé ici à des références relevant de la bibliographie Ethernet car il en existe à ce sujet une multitude mais elles ne sont pas toutes valables. De plus, Internet regorge également d'un nombre très important d'informations.

**IANA** (International Assigned Numbers Authority). Recherche plein texte pour la collecte RFC, etc. <http://www.iana.org>.

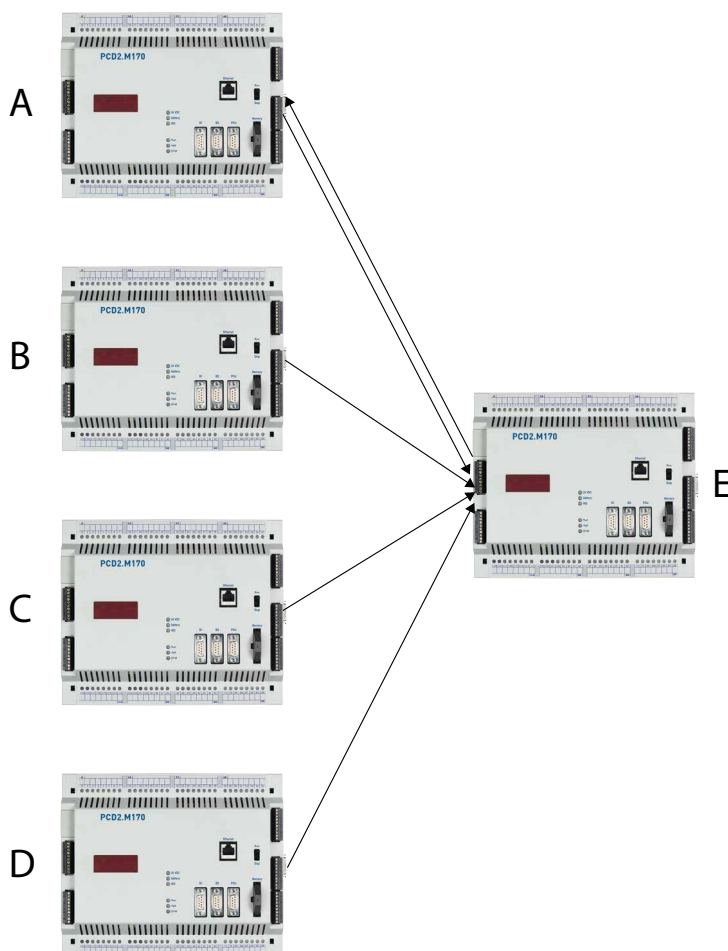
**Hirschmann** : Pour nos essais internes, nous utilisons généralement du matériel industriel de la société Hirschmann dont nous sommes satisfait. Pour leur page d'accueil avec une recherche texte plein fonctionnelle et des bons manuels, voir leur site <http://www.hirschmann.com>.

Un **analyseur Ethernet** gratuit est disponible à l'adresse <http://www.wireshark.org>. Pour de plus amples informations, voir les FAQ #100569.

6

### 6.2 Mesures de performances

Les résultats sont basés sur des mesures effectuées dans un réseau local séparé avec concentrateur. Les PCD2.M170 ont été utilisés et le PCD7.F65x a été équipé du microprogramme version 041. La communication a été testée avec S-Bus UDP.





**Saia PCD® E arrêté**

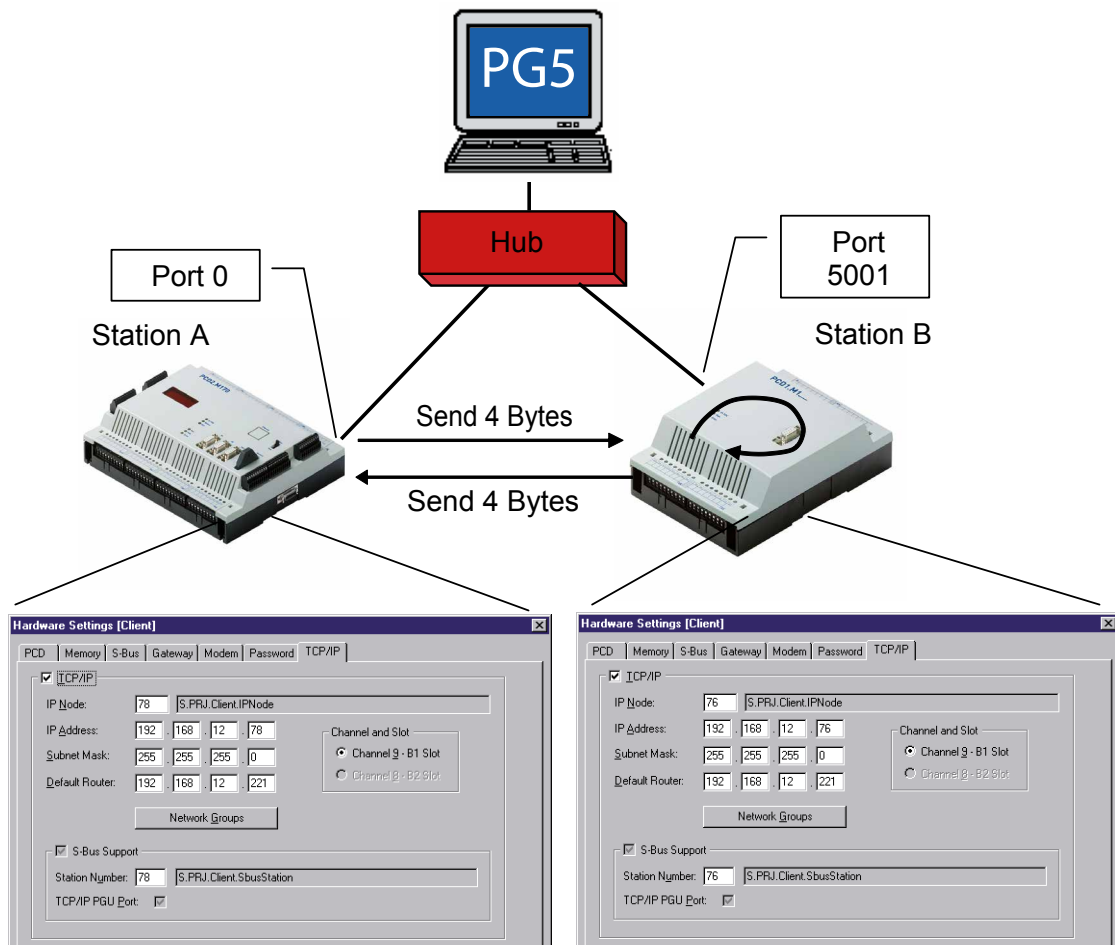
A → E	A: 124 S-Bus Tlgs/s ce qui signifie qu'1 télégramme S-Bus nécessite 8 ms
A → E + B → E	A + B: 112 + 112 = 224 S-Bus Tlgs/s
A → E + B → E + C → E	A + B + C: 103 + 103 + 103 = 309 S-Bus Tlgs/s
A → E + B → E + C → E + D → E	A + B + C + D: 86 + 85 + 85 + 86 = 342 S-Bus Tlgs/s Dans cette constellation, la largeur de bande est à peu près atteinte

**Saia PCD® E envoie des télégrammes S-Bus sur Saia PCD® A**

A → E + B → E + C → E + E → A	E: 79 S-Bus Tlgs/s
A → E + B → E + C → E + D → E + E → A	E: 63 S-Bus Tlgs/s

**6.3 Exemple de programmation : Mode Open Data TCP/IP**

Un exemple programmé de connexion TCP/IP en mode Open Data est illustré ci-après. Il implémente un serveur écho client dans TCP/IP. Un registre préalablement initialisé est envoyé au serveur qui le retourne. Le client incrémente le registre et envoie la requête suivante.

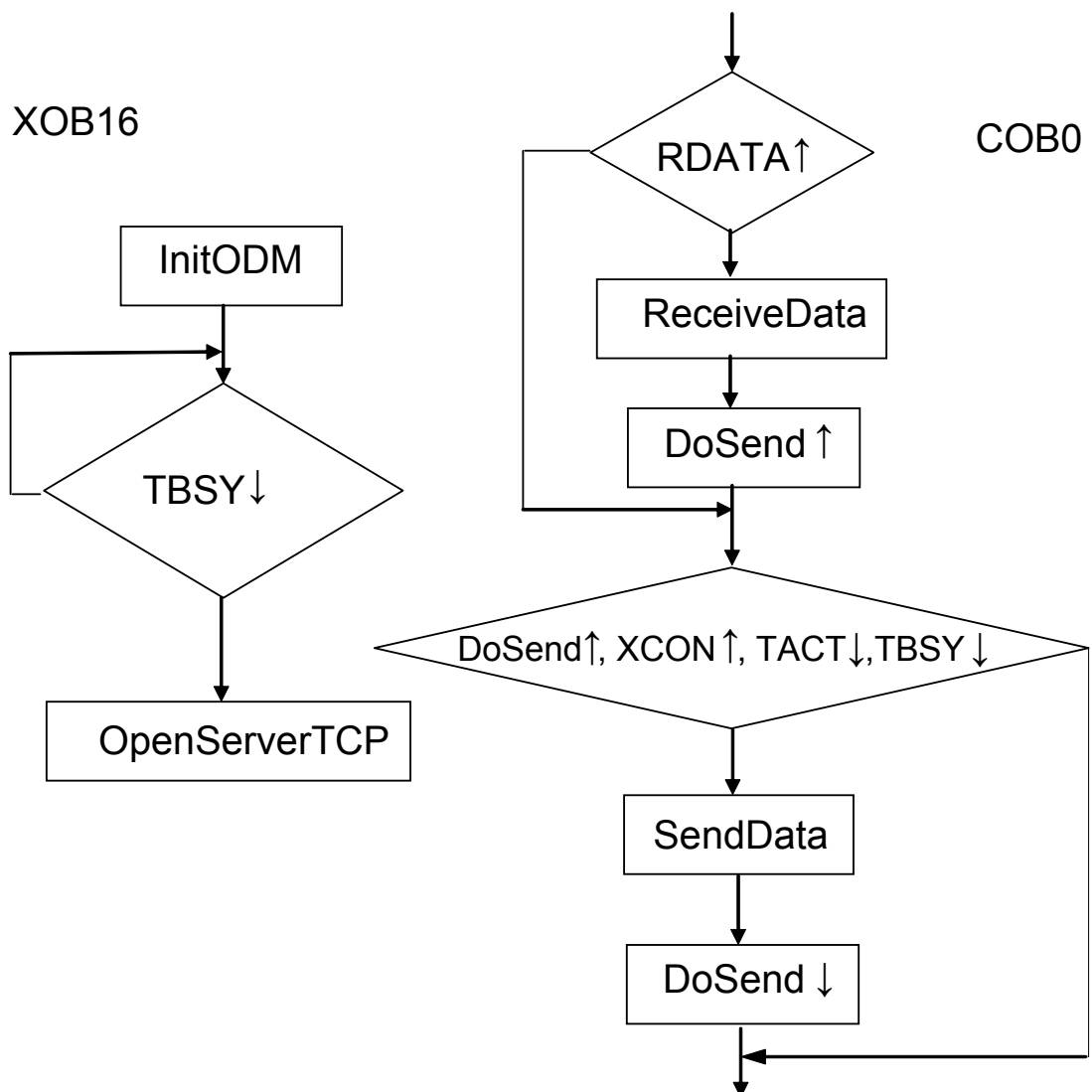


6.3.1 Serveur

Organigramme du serveur

- XOB 16 Initialise le mode Open Data  
Ouvre une socket serveur TCP sur le port 5001  
Commute en mode « accord client automatique ».  
Attend la connexion avec le client.
- COB 0 Reçoit des données  
Retourne les données reçues

Pour faciliter la représentation, pour les questions OUI/NON, les étapes « JR -1 » ont été utilisées.



**Liste de codes serveur (IL)**

Pour faciliter la représentation, les étapes « JR-1 » ont été utilisées.

```

$include "IPLib.inc"
$init
    LD    Conn_Tout          ;temporisation sur illimité
        0

    CSF   S.IP.Library      ; initialise le mode Open Data
        S.IP.InitODM
        GlobalFlags        ;indicateurs de diag. global
        ChannelReg         ;pour plusieurs canaux :
                            ;ici figure le numéro du canal
                            ;dans lequel les données reçues
                            ;ont été écrites
        0                   ;Temporisation 0 signifie : reste illimité
                            ;dans boîte de réception

    STL   GlobalFlags[TBSY] ;vérifie si l'émetteur est libre
    JR    L-1

                                ;ouvre le port serveur TCP

    CSF   S.IP.Library
        S.IP.OpenServerTCP
        Channel            ;canal virtuel sur lequel les données
                            ;sont envoyées et reçues
        Local_Port        ; port IP local via lequel les données
                            ;sont envoyées et reçues
        DiagFlag          ;Indicateurs de diagnostic de canal
        DiagReg           ;registre de diagnostic de canal
        Filter            ;si acceptation avec ou sans filtre
        Conn_Tout         ;temporisation de la connexion : 0
                            ;pour néant,
                            ;sinon x secondes

    RES   DoSend           ;Déclencheur d'envoi
    RES   DoDisconnect

$endinit

```

```

        COB    0
            0
read: STH    GlobalFlags[RDATA]    ;est défini lorsqu'un télégramme
                                       ;arrive
JR          L    nodata

        CSF   H S.IP.Library        ;Réception de données
            S.IP.ReceiveData
            Channel                  ;Saisie du numéro du canal de réception,
                                       ;sur lequel les données sont
                                       ;reçues. Pour plusieurs canaux :
                                       ;Valeur du registre de canal.
            RecvIP                  ;Données reçues à partir de cette
                                       ;adresse IP
            RecvPort                ;Données reçues via ce port
            4                       ;Longueur de données max. acceptée
                                       ;(tampon)
            RecvLnth                ;Longueur de données reçues effective
            RecvData                ;Données reçues

        SET   DoSend

nodata STH   DoSend
        ANH   DiagFlag[XCON]        ;Port connecté (canal)
        ANL   DiagFlag[TACT]        ;et émetteur inactif (canal)
        ANL   GlobalFlags[TBSY]     ;et émetteur libre (global)
        JR    L    nosend            ;Envoi impossible
        CSF   H S.IP.Library        ;Envoi de données
            S.IP.SendData
            Channel                  ;Canal virtuel
            RecvIP                  ;Adresse IP partenaire
            RecvPort                ;Port partenaire
            4                       ;Envoi de 4 octets
            RecvData                ;Retour ds 4 octets préalablement
                                       ;reçus

        RES   DoSend



























        STH   DoDisconnect
        JR    L    nosend

        CSF   S.IP.Library
            S.IP.DisconnectTCP
            Channel
            RecvIP
            RecvPort
        RES   DoDisconnect

nosend: ECOB

```

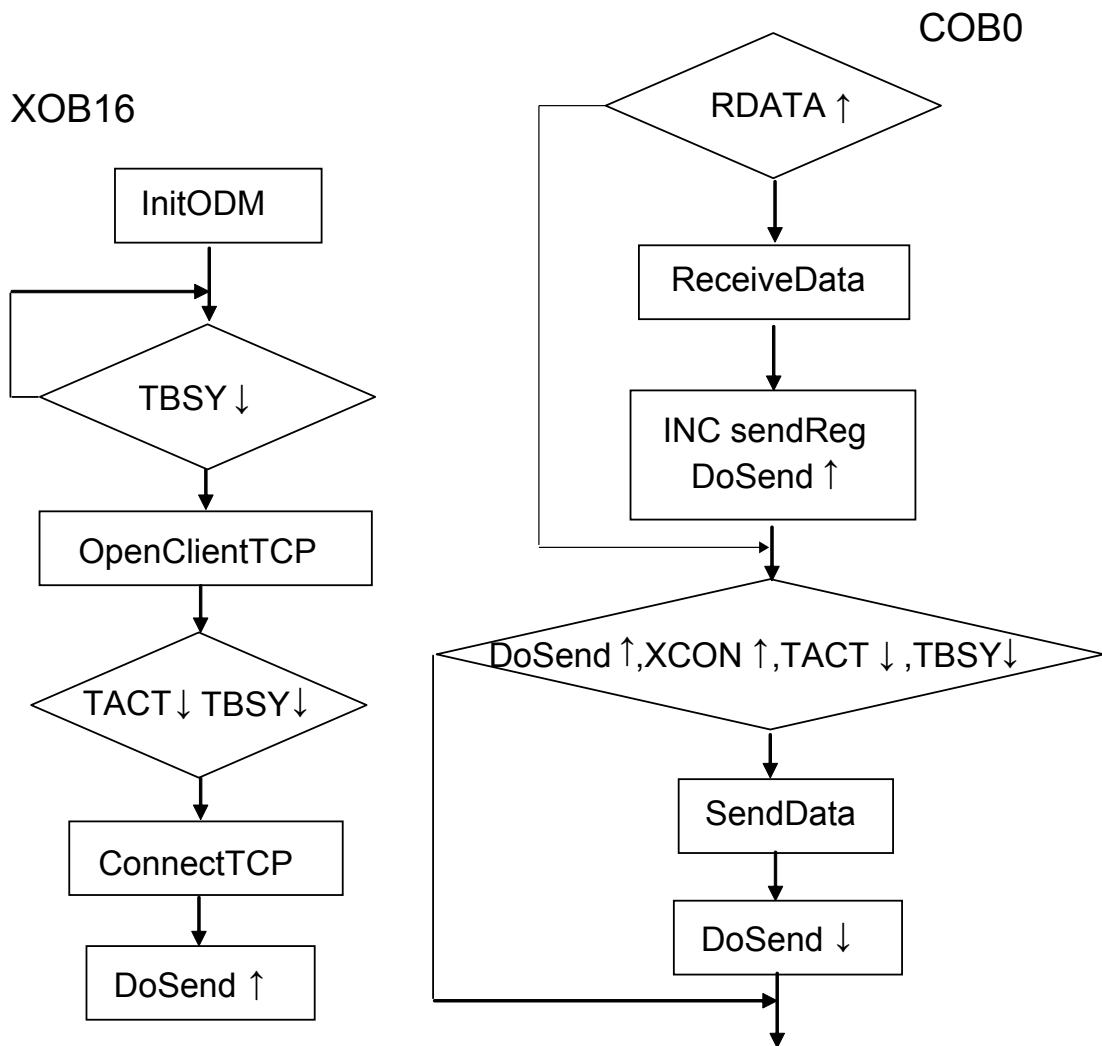
**Code serveur : Déclaration et initialisation des variables**

Group/Symbol	Type ▲	Address/Value	Comment
[-] 			
[-]  RDIA		0	
[-]  TBSY		0	
[-]  RDATA		1	
[-]  TACT		1	
[-]  RCON		2	
[-]  TDIA		2	
[-]  XCON		3	
[-]  NEXE		4	
[-]  RemotePort		65535	
[-]  GlobalFlags	F	0 [3]	
[-]  DoSend	F	5	
[-]  DiagFlag	F	10 [5]	
[-]  DoDisconnect	F	100	
[-]  Filter	K Constant	0	
[-]  Channel	K Constant	10	
[-]  Local_Port	K Constant	5001	
[-]  Conn_Tout	R		
[-]  ChannelReg	R	0	
[-]  DiagReg	R	1	
[-]  SendReg	R	2	
[-]  RecvPort	R	3	
[-]  RecvLnth	R	4	
[-]  RecvData	R	5	
[-]  RecvIP	Text RAM		
[-]  RemoteIP	Text RAM		

### 6.3.2 Client

#### Organigramme du client

- XOB16 Initialise le mode Open Data  
Ouvre une socket client TCP sur le port 0  
Etablit la communication avec le serveur sur le port 5001
- COB0 Contrôle les données  
Reçoit les données  
Incrémente le registre (compteur)  
Envoie les données



**Liste de codes client (IL)**

Pour faciliter la représentation, les étapes « JR-1 » et « JR-2 » ont été utilisées.

```

$include "IPLib.inc"
$init
    LD    SendReg          ;initialise le registre d'envoi
        0
    LD    Conn_Tout       ;temporisation sur illimité
        0

    CSF   S.IP.Library    ; initialise le mode Open Data
        S.IP.InitODM
        GlobalFlags      ;indicateurs de diag. global
        ChannelReg       ;pour plusieurs canaux :
                        ;ici figure le numéro du canal
                        ;dans lequel les données reçues
                        ;ont été écrites
        0                ;Temporisation 0 signifie : reste illimité
                        ;dans boîte de réception

    STL   GlobalFlags[TBSY] ;vérifie si l'émetteur est libre
    JR    L -1

                                ;ouvre le port serveur TCP

    CSF   S.IP.Library
        S.IP.OpenClientTCP
        Channel          ;canal virtuel sur lequel les données
                        ;sont envoyées et reçues
        Port             ; port IP local via lequel les données
                        ;sont envoyées et reçues
        DiagFlag        ;Indicateurs de diagnostic de canal
        DiagReg         ;registre de diagnostic de canal
        Filter          ;si acceptation avec ou sans filtre
        Conn_Tout       ;temporisation de la connexion : 0
                        ;pour néant,
                        ;sinon x secondes

    STL   DiagFlag[TACT]
    ANL   GlobalFlags[TBSY]
    JR    L -2

    CSF   S.IP.Library    ;Etablissement de la communication
                        ;avec le serveur TCP
        S.IP.ConnectTCP
        Channel          ;Canal virtuel
        RemoteIP        ;Adresse IP serveur
        RemotePort      ;Port serveur

    SET   DoSend

    RES   DoDisconnect

$endinit

```

```

COB  0
      0
read: STH GlobalFlags[RDATA] ;est défini lorsqu'un télégramme
      ;arrive

JR   L nodata

CSF  H S.IP.Library ;Réception de données
      S.IP.ReceiveData
      Channel ;Saisie du numéro du canal de réception,
              ;sur lequel les données sont
              ;reçues. Pour plusieurs canaux :
              ;Valeur du registre de canal.
      RecvIP ;Données reçues à partir de cette
              ;adresse IP
      RecvPort ;Données reçues via ce port
      4 ;Longueur de données max. acceptée
              ;(tampon)
      RecvLnth ;Longueur de données reçues effective
      RecvData ;Données reçues

SET  DoSend

nodata STH DoSend
      ANH DiagFlag[XCON] ;Port connecté (canal)
      ANL DiagFlag[TACT] ;et émetteur inactif (canal)
      ANL GlobalFlags[TBSY] ;et émetteur libre (global)
      JR L nosend ;Envoi impossible
      CSF H S.IP.Library ;Envoi de données
          S.IP.SendData
          Channel ;Canal virtuel
          RecvIP ;Adresse IP partenaire
          RecvPort ;Port partenaire
          4 ;Envoi de 4 octets
          RecvData ;Retour ds 4 octets préalablement
                  ;reçus

RES  DoSend

STH  DoDisconnect
JR   L nosend


CSF  S.IP.Library ;Interruption de la communication
      S.IP.DisconnectTCP
      Channel
      RecvIP
      RecvPort
RES  DoDisconnect

nosend: ECOB

```








**Code client : Déclaration et initialisation des variables**

Group/Symbol	Type ▲	Address/Value	Comment
[-] 			
[-] <input type="checkbox"/> RDIA		0	
[-] <input type="checkbox"/> TBSY		0	
[-] <input type="checkbox"/> RDATA		1	
[-] <input type="checkbox"/> TACT		1	
[-] <input type="checkbox"/> RCON		2	
[-] <input type="checkbox"/> TDIA		2	
[-] <input type="checkbox"/> XCON		3	
[-] <input type="checkbox"/> NEXE		4	
[-] <input type="checkbox"/> RemotePort		5001	
[-] <input type="checkbox"/> GlobalFlags	F	0 [8]	
[-] <input type="checkbox"/> DiagFlag	F	8 [8]	
[-] <input type="checkbox"/> DoSend	F	16	
[-] <input type="checkbox"/> DoDisconnect	F	100	
[-] <input type="checkbox"/> Channel	K Constant	10	
[-] <input type="checkbox"/> Port	K Constant	5005	
[-] <input type="checkbox"/> Conn_Tout	R		
[-] <input type="checkbox"/> ChannelReg	R	0	
[-] <input type="checkbox"/> DiagReg	R	1	
[-] <input type="checkbox"/> SendReg	R	2	
[-] <input type="checkbox"/> RecvPort	R	3	
[-] <input type="checkbox"/> RecvLnth	R	4	
[-] <input type="checkbox"/> RecvData	R	5	
[-] <input type="checkbox"/> RecvIP	Text RAM		
[-] <input type="checkbox"/> RemoteIP	Text RAM		

## A Annexe

### A.1 Icônes

	Ce symbole indique que des informations supplémentaires sur ce thème existent dans ce manuel ou dans un autre, ou encore dans des documents techniques. Il n'existe aucun renvoi direct à de tels documents.
	Ce symbole avertit le lecteur que des composants peuvent être endommagés par une décharge électrostatique au contact. Recommandation : touchez au minimum le pôle moins du système (boîtier de la fiche PGU) avant d'entrer en contact avec des pièces électroniques. Il est encore mieux de porter au poignet un bracelet mis à la terre, relié au pôle moins du système.
	Ce symbole désigne des instructions qui doivent être strictement suivies.
	Les explications à côté de ce symbole ne s'appliquent qu'à la série Saia PCD® Classic.
	Les explications à côté de ce symbole ne s'appliquent qu'à la série Saia PCD® xx7.

## A.2 Adresses

### Saia-Burgess Controls AG

Bahnhofstrasse 18  
3280 Murten / Suisse

Téléphone : +41 26 580 30 00

Télécopie : +41 26 580 34 99

E-mail : [support@saia-pcd.com](mailto:support@saia-pcd.com)

Page d'accueil : [www.saia-pcd.com](http://www.saia-pcd.com)

Assistance: [www.sbc-support.com](http://www.sbc-support.com)

Entreprises de distribution international &

Représentants SBC : [www.saia-pcd.com/contact](http://www.saia-pcd.com/contact)

### Adresse postale pour les retours de produits par les clients de "Vente Suisse" :

#### Saia-Burgess Controls AG

Service Après-Vente  
Rue de la Gare 18  
3280 Morat / Suisse