

Modbus Utilisateur Interface

| | | |
|----------|--|------|
| 0 | Sommaire | |
| 0.1 | Historique du document..... | 0-3 |
| 0.2 | Marques déposées | 0-3 |
| 1 | Le protocole Modbus | |
| 1.1 | Modèle ISO / OSI..... | 1-1 |
| 1.2 | Communication Modbus | 1-2 |
| 1.3 | Supports Modbus | 1-4 |
| 2 | SBC Modbus | |
| 2.1 | Fonctionnalités de SBC Modbus | 2-1 |
| 2.2 | Diagnostic..... | 2-3 |
| 2.2.1 | Drapeaux de diagnostic client / serveur | 2-3 |
| 2.2.2 | Registre Diagnostic | 2-4 |
| 3 | Serveur Modbus SBC | |
| 3.1 | Instance serveur | 3-1 |
| 3.2 | UID serveur..... | 3-2 |
| 3.3 | Mappage des supports | 3-2 |
| 3.4 | Mappage par défaut | 3-5 |
| 4 | Client Modbus SBC | |
| 4.1 | Canal client..... | 4-1 |
| 4.2 | Requêtes Modbus | 4-2 |
| 5 | Bibliothèque FBox Modbus | |
| 6 | Bibliothèque des fonctions système Modbus | |
| 6.1 | Introduction | 6-1 |
| 6.2 | Paramètres | 6-2 |
| 6.3 | Codes d'erreur | 6-6 |
| 6.3.1 | Codes d'erreur CSF | 6-6 |
| 6.3.2 | Codes d'erreur ModbusShell | 6-7 |
| 6.3.3 | Codes d'erreur ModbusDriver | 6-7 |
| 6.4 | Codes de fonction | 6-8 |
| 6.5 | Protocoles..... | 6-8 |
| 6.6 | Types de données Modbus | 6-8 |
| 6.7 | Types de zones Modbus (serveur) | 6-8 |
| 6.8 | Types d'accès aux zones Modbus (serveur) | 6-9 |
| 6.9 | Types de traitement (client) | 6-9 |
| 6.10 | Types de supports Saia PCD® | 6-9 |
| 6.11 | Codes d'exception | 6-10 |

| | | |
|----------|---|-----|
| 7 | Spécification CSF Modbus | |
| 7.1 | S.SF.Modbus.InitSerialPort (Client / Server) | 7-1 |
| 7.2 | S.SF.Modbus.OpenChannel (Client) | 7-2 |
| 7.3 | S.SF.Modbus.SendReadRequest / S.SF.Modbus.SendWriteRequest (Client)..... | 7-3 |
| 7.4 | S.SF.Modbus.InitServer (Server)..... | 7-4 |
| 7.5 | S.SF.Modbus.InitUID (Server) | 7-5 |
| 7.6 | S.SF.Modbus.InitMap (Server) | 7-6 |
| A | Annexe | |
| A.1 | Icônes | A-1 |
| A.2 | Liste d'abréviations | A-2 |
| A.3 | Adresses..... | A-3 |

0.1 Historique du document

0

| Date | Version | Changements | Remarques |
|------------|---------|-------------------------|--|
| 2009-06-18 | FR01 | - | Nouvelle édition |
| 2012-05-31 | FR01 | - | Texte dans la « Figure 7 » |
| 2011-05-24 | FR02 | chapitre 7.2 point 5 | Close Timeout, modification de S.Modbus à S.SF.Modbus |
| 2013-10-23 | FR03 | - | Nouveau logo et nouveau nom de l'entre- prise. |
| 2019-01-09 | FRA04 | Chapitre 6.4 | «Codes de fonction»: valeurs modifiées |

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, 2019. © Tous droits réservés.

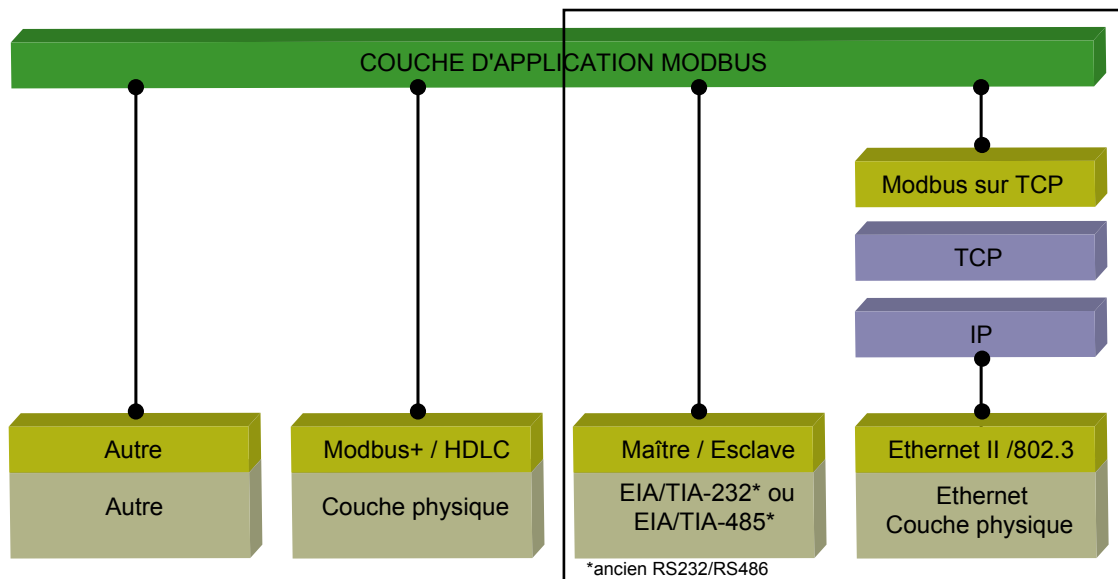
Publié en Suisse.

1 Le protocole Modbus

1.1 Modèle ISO / OSI

1

Modbus est un protocole de messagerie qui permet la communication entre un maître et plusieurs esclaves. Il peut être appliqué sur les liaisons série (RS-485, RS-232, RS-422) en modes RTU et ASCII ou sur Ethernet via TCP et UDP. Dans le cas des protocoles TCP / UDP, le port public 502 est réservé pour Modbus mais d'autres ports peuvent être définis.



*ancien RS232/RS486

Couches physiques supportées par les automates Saia PCD®

1.2 Communication Modbus

Modbus est un protocole de requête / réponse qui offre des services définis par codes de fonction. 127 codes de fonction publics et définis par l'utilisateur sont disponibles et peuvent être utilisés pour l'accès aux données (lecture, écriture), le diagnostic et autres services.

1

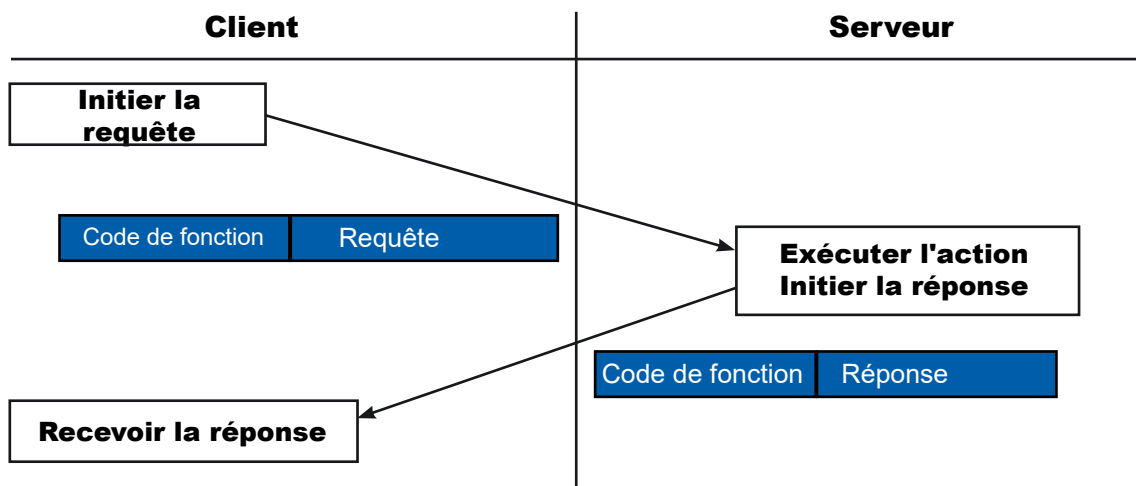


Figure 2 : Transaction Modbus (sans erreurs)

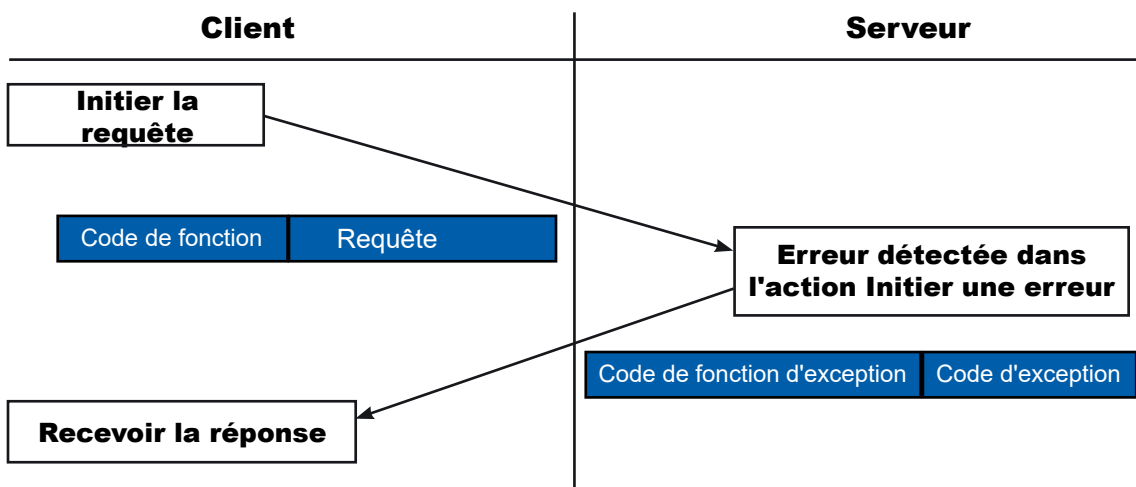


Figure 3 : Transaction Modbus (réponse négative)

En mode série, au maximum 247 esclaves peuvent être présents sur le bus, chacun ayant au minimum une adresse unique appelée identifiant d'unité (IDU).

RTU est le mode par défaut de Modbus sur liaisons série. Ce mode est à temps contrôlé : les temps intercaractère et intertrame doivent être respectés, sinon le protocole va être refusé.

Le mode ASCII peut également être utilisé pour Modbus sur liaison série. Dans ce cas, la communication est contrôlée par des caractères de début et de fin spécifiques. Mais, comme un octet de données est envoyé à l'aide de 2 caractères, deux fois plus de données que RTU sont transmises.

La communication Modbus est basée sur des trames spécifiques qui incluent au moins un code de fonction et les données utiles. Pour Modbus sur une liaison série, un CCR est joint ainsi qu'une adresse. L'adresse peut également être utilisée pour Modbus sur IP.

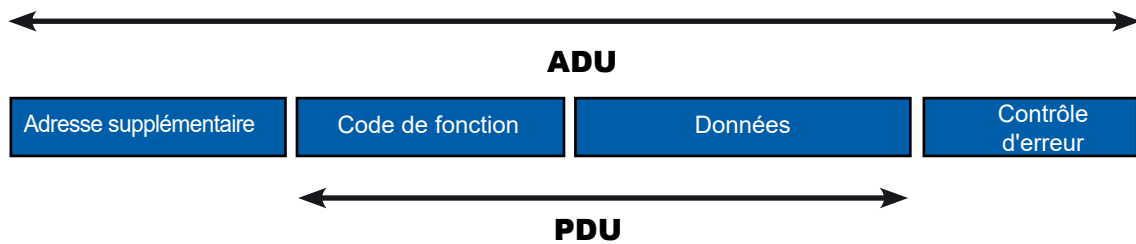


Figure 4 : Trame Modbus générale

2 types de requêtes peuvent être envoyés :

- Requêtes de lecture : Données = adresse Modbus + nombre d'éléments à lire
→ Réponse : Données = nombre d'octets de données + données
- Requêtes d'écriture : Données = adresse Modbus + nombre d'éléments à écrire + données à écrire
→ Réponse : Données = Données de requête
- Si une erreur associée à la fonction Modbus demandée se produit, une réponse négative est envoyée par le serveur, avec un code de fonction d'exception :
→ Code de fonction d'exception = Code de fonction de requête + 0x80,
données = Code d'exception

1.3 Supports Modbus

Modbus définit 4 types de support :

- Registres de maintien (HR) : 16 bits - lecture / écriture
- Registres d'entrée (IR) : 16 bits - lecture uniquement
- Bobines : 1 bit - lecture / écriture
- Entrées discrètes (DI) : 1 bit - lecture uniquement

1

Dans chaque périphérique, ces supports Modbus correspondent aux supports spécifiques des périphériques. Chaque périphérique peut avoir sa propre organisation de données ; par conséquent, il est nécessaire que l'utilisateur consulte la documentation du périphérique avant d'utiliser Modbus. Deux exemples d'organisation des données sont illustrés ci-après.

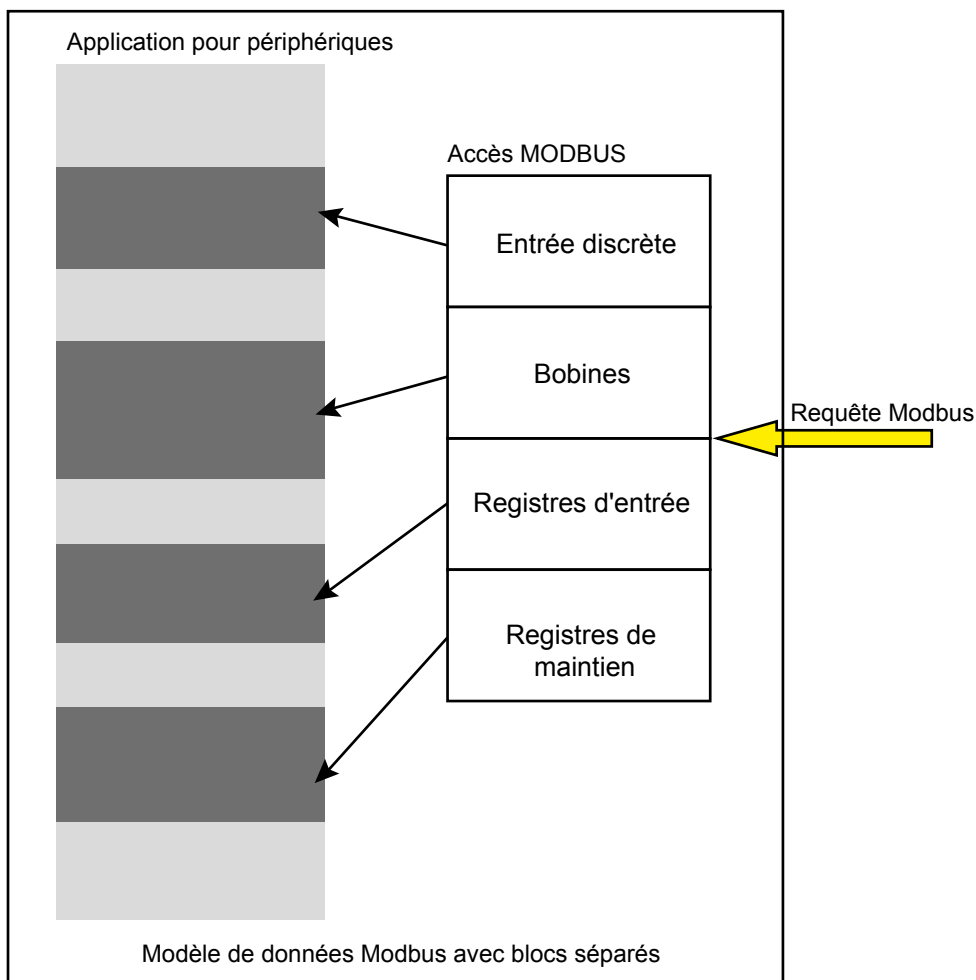


Figure 5 : Modèle de données Modbus avec blocs séparés

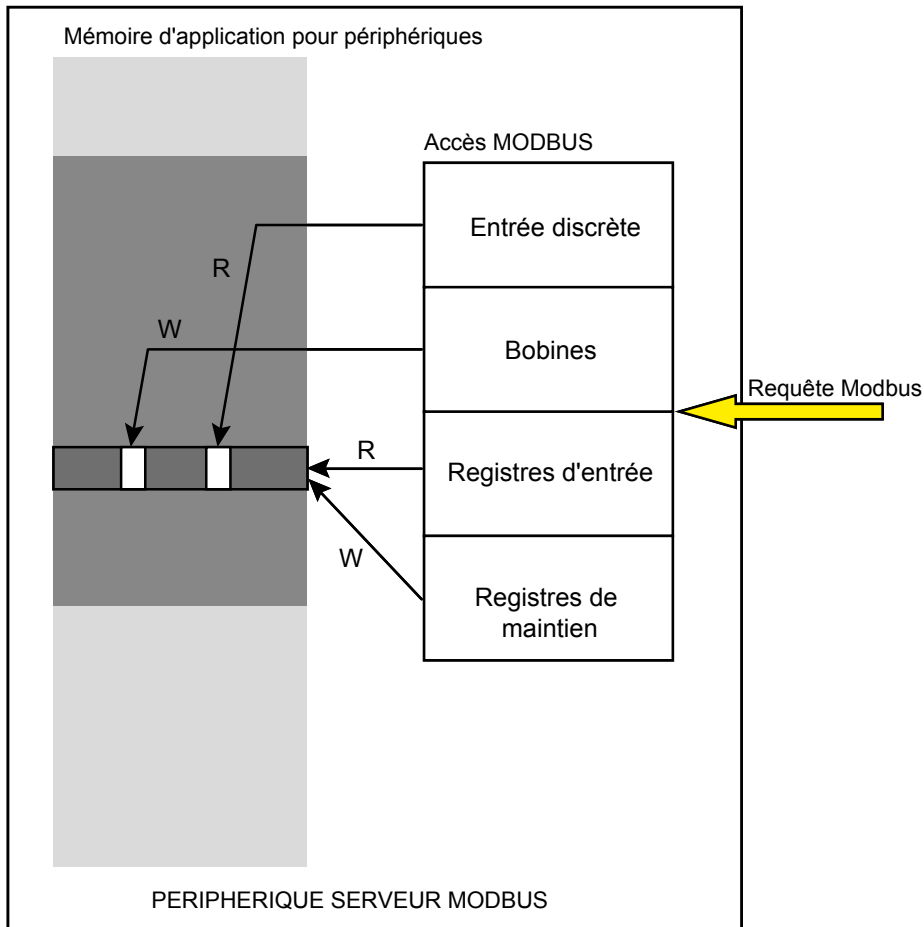


Figure 6 : Modèle de données Modbus avec 1 seul bloc

Dans une trame Modbus, chaque donnée peut être adressée entre 0 et 65535, tandis que le modèle de données Modbus adresse chaque élément dans un bloc de données entre 1 et n. Le modèle de données Modbus doit alors être lié à l'application pour périphériques. Le mappage entre le modèle de données Modbus et l'application pour périphériques est spécifique du fournisseur.

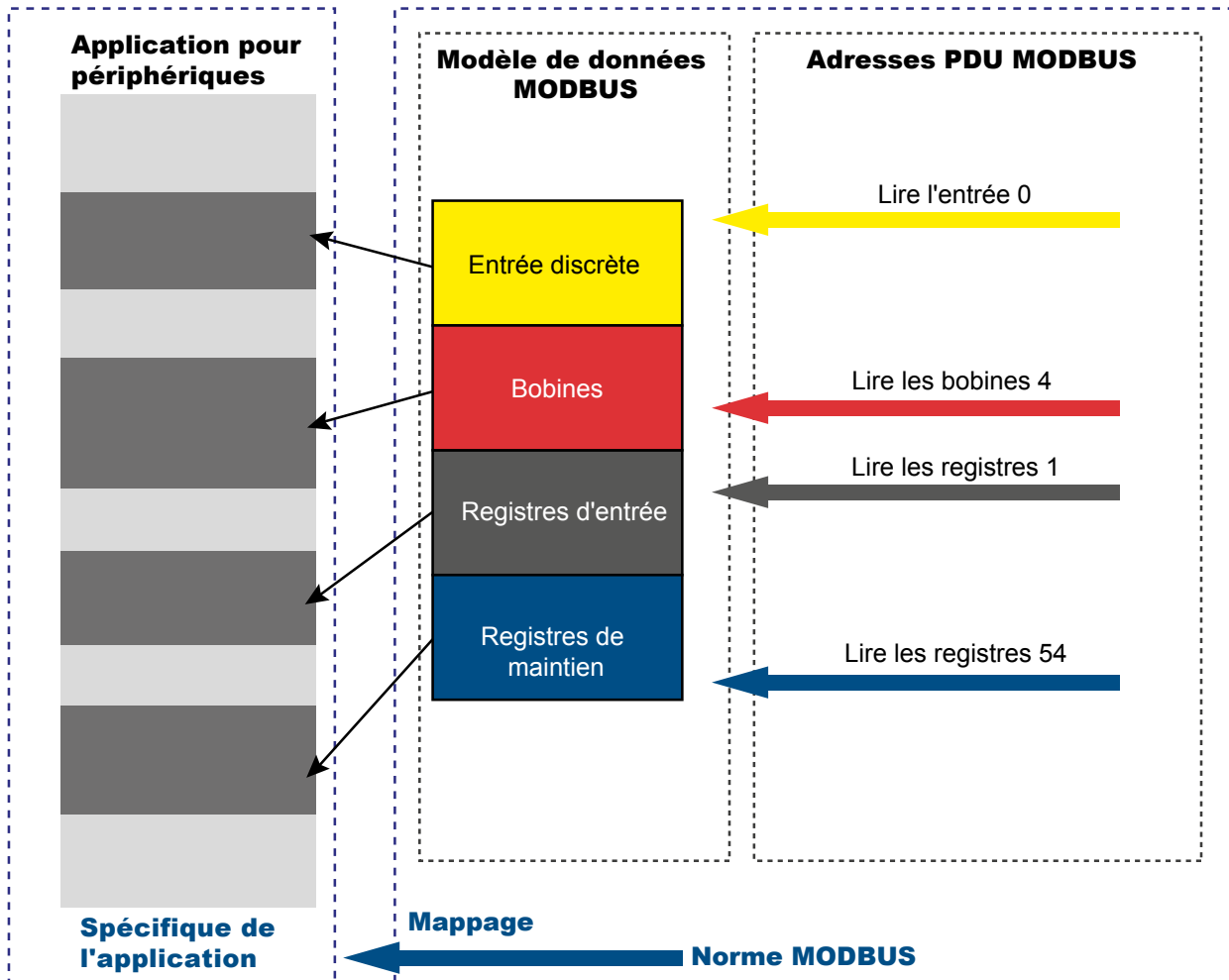


Figure 7 : Modèle d'adressage Modbus

Pour plus d'informations sur Modbus, consultez le site Internet <http://www.modbus.org>.

2 SBC Modbus

Cette partie décrit l'application du protocole Modbus sur Saia PCD®. Dans cette partie, les termes Client et Serveur sont utilisés, soit pour TCP / UDP soit pour RTU / ASCII à la place de Maître et Esclave.

2

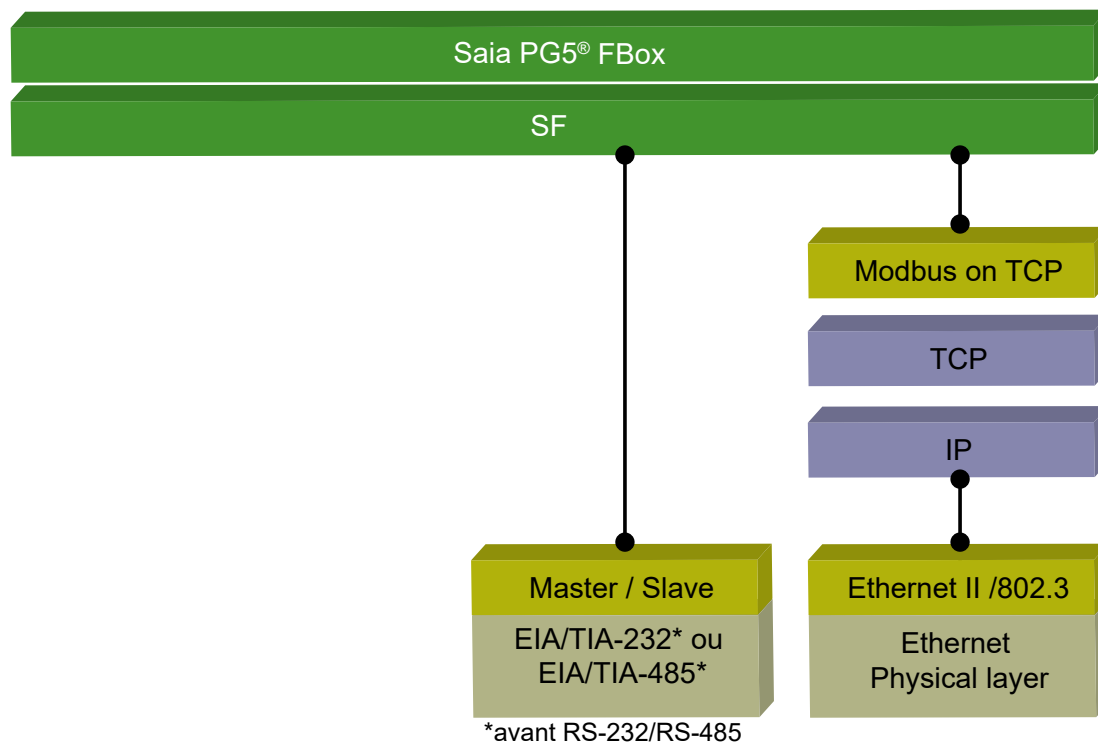


Figure 8 : Pile de protocoles de communication SBC Modbus

L'application de Modbus par SBC permet à l'utilisateur de configurer et d'utiliser des clients et serveurs Modbus. Ceci peut avoir lieu à l'aide d'une interface dédiée qui consiste en Saia PG5® FBox et CSF.

2.1 Fonctionnalités de SBC Modbus

- **Fonctionnalités client** et serveur
- **Modes** : IP (TCP et UDP) et série (RTU et ASCII)
- **Modbus série** : Disponible sur tous les ports série, entre 1200 Bd et 115,2 kBd Pour utiliser Modbus sur un module PCD2.F2xxx ou PCD3.F2xxx, la version minimale du FW est 1.14.23 et le module F2xx doit avoir une version matérielle D ou plus récente.

- **8 codes de fonction :**
 - Lire les bobines
 - Lire les entrées discrètes
 - Lire les registres de maintien
 - Lire les registres d'entrée
 - Ecrire une sortie (coil - bobine)
 - Ecrire plusieurs sorties (coil - bobines)
 - Ecrire un seul registre de maintien
 - Ecrire plusieurs registres de maintien

- **Mappage des supports :** définissable par l'utilisateur

- **Traitements des données :** nombreux traitements disponibles pour garantir la compatibilité avec d'autres périphériques que SBC, pour permettre l'accès 32 bits aux registres, le format de virgule flottante, etc.

- **Serveurs :** maximum 4 sur un système Saia PCD® (un serveur est défini par une paire {port / protocole} unique) – Exemple : 1 serveur TCP port 502, 1 serveur TCP port 503, 1 serveur UDP port 502 et 1 serveur RTU sur le port série 2.

- **ID d'unité (UID) :** maximum 10 sur un système Saia PCD® : 10 = 9 + IDU 255 (les IDU entre 0 et 254 sont autorisés ; l'IDU 255 existe par défaut sur le serveur mais il doit être configuré avant tout accès).

- **Zones de mappage :** maximum 10 par IDU – Si plus de 10 zones de mappage sont nécessaires, un IDU supplémentaire peut être défini (exemple : définir un IDU pour bobines/entrées discrètes et un autre pour registres de maintien/d'entrée).

- **Canaux :** maximum 10 sur un système Saia PCD® (un canal est défini par une paire {port / protocole} unique) - Exemple, 1 canal TCP sur le port serveur 502, 1 canal TCP sur le port serveur 503, 1 canal UDP sur le port serveur 502 et 1 canal RTU utilisant le port 1.

- **Connexions :** 26 connexions peuvent être ouvertes simultanément. Sur ces 26, maximum 10 connexions peuvent être ouvertes côté client. Les connexions restantes peuvent être utilisées par le serveur. Remarque : une connexion est définie par un triplet unique {Port - Protocole - Adresse IP} dans TCP / UDP et une seule paire Port / Protocole) en mode série). Les connexions sont ouvertes à la première requête Modbus. Dans TCP/UDP, les connexions sont automatiquement fermées au-delà d'un délai défini par l'utilisateur.

2.2 Diagnostic

Des drapeaux et registres de diagnostic sont utilisés pour signaler l'état des UID (UID de diagnostic) et des canaux (diagnostic de canal). Ils sont également utilisés en mode série pour signaler les erreurs qui peuvent se produire sur la liaison série (diagnostic serveur et diagnostic canal).

Dans le registre diagnostic, tous les bits sauf « Recommencer le comptage » et « Délai de réponse » sont cumulés (c'est à l'utilisateur de les effacer).

2

2.2.1 Drapeaux de diagnostic client / serveur

| Adresse | Nom | Description |
|----------|---------|---|
| Fxxx | RBSY | Canal : non utilisé UID : UID occupé Serveur : non utilisé |
| Fxxx + 1 | RFUL | Canal : non utilisé UID : une requête avec un UID valide a été reçue et traitée Serveur : non utilisé |
| Fxxx + 2 | RDIA | Diagnostic de réception (voir registre Diagnostic) |
| Fxxx + 3 | TBSY | Canal : Canal occupé (si le canal est occupé, aucune requête ne peut être envoyée dessus) UID : non utilisé Serveur : non utilisé |
| Fxxx + 4 | Réservé | - |
| Fxxx + 5 | TDIA | Diagnostic de transmission (voir registre Diagnostic) |
| Fxxx + 6 | Réservé | - |
| Fxxx + 7 | NEXE | Client : la requête n'a pas été traitée avec succès UID : non utilisé Serveur : non utilisé |

2.2.2 Registre Diagnostic

2

| | Bit | Désignation | Description | Protocole | Utilisé pour |
|--------------------------------|-----|--------------------|--|-----------|---------------|
| R E C E P T I O N | 0 | Overrun Error | Surcharge du tampon de réception interne | Série | Canal/Serveur |
| | 1 | Parity Error | Erreur de parité du caractère reçu | Série | Canal/Serveur |
| | 2 | Framing Error | Débit en bauds incorrect | Série | Canal/Serveur |
| | 3 | Break Error | Coupure sur la ligne série | Série | Canal/Serveur |
| | 4 | CRC Error | CRC incorrect | Série | Canal/Serveur |
| | 5 | - | - | | |
| | 6 | - | - | | |
| | 7 | - | - | | |
| | 8 | Length Error | Télégramme avec longueur invalide reçue | Série | Canal/Serveur |
| | 9 | Media Access Error | Erreur lors de l'accès au support Saia PCD® | Tous | Canal/UID |
| | 10 | - | - | | |
| | 11 | Server Start Error | Le serveur n'a pas pu démarrer | Tous | Serveur |
| | 12 | Range Error | Erreur dans l'adresse ou le numéro des éléments Modbus auxquels il faut accéder (voir ci-après). | Tous | UID |
| | 13 | Value Error | Une valeur dans le télégramme est invalide (type de support, etc.) | Tous | UID |
| | 14 | Area access error | Zone non accessible (UID occupé, accès la zone non autorisé) | Tous | UID |
| 15 | - | - | | | |
| T R A N S M I S S I O N | 16 | Retry count | Nombre de répétitions effectuées | Tous | Canal |
| | 17 | | | | |
| | 18 | | | | |
| | 19 | | | | |
| | 20 | Response exception | Exception de réponse reçue | Tous | Canal |
| | 21 | Response Timeout | Aucune réponse reçue après un délai donné | Tous | Canal |
| | 22 | Response Error | Réponse invalide reçue | Tous | Canal |
| | 23 | - | - | | |
| | 24 | Send error | La requête n'a pas pu être envoyée | Tous | Canal |
| | 25 | - | - | | |
| | 26 | Client Start Error | Le client n'a pas pu être démarré | Tous | Canal |
| | 27 | - | - | | |
| | 28 | Media Error | Erreur de paramètre de support Saia PCD® | Tous | Canal |
| | 29 | - | - | | |
| | 30 | Status Error | Erreur de statut client (erreur interne) | Tous | Canal |
| | 31 | Program Error | Erreur de programme utilisateur (ex. : appel de CSF lorsque TBSY ou RBSY est défini) | Tous | Canal, UID |

Une « **Erreur de plage** » peut avoir différentes causes :

- Aucun mappage correspondant trouvé : aucune zone de mappage correspondant à l'adresse de démarrage des données et au nombre d'éléments Modbus n'a pu être trouvée dans les zones de mappage définies par l'utilisateur (et également dans les zones de mappage par défaut si DEFMAP = 1).
- Le nombre de supports Saia PCD® auxquels il faut accéder dépasse la plage Saia PCD® de la zone : par exemple, lors de la tentative d'accès à 8 registres Saia PCD® avec décalage de 2 dans une zone contenant 8 registres Saia PCD®. L'attention doit être portée sur le nombre de registres Saia PCD® nécessaires lors de l'utilisation de trous.
- Nombre impair ou décalage de HR lors de l'accès aux registres 32 bits : dans une zone 32 bits 1 registre Saia PCD® (32 bits) correspond à 2 registres de maintien Modbus. Il est par conséquent interdit d'accéder aux registres Saia PCD® avec un décalage impair ou avec un nombre impair de registres de maintien.

(voir Figure 12 : Mappage des registres de maintien (16 bits) sur les registres Saia PCD® (32 bits) dans une zone 32 bits)

Une « **Erreur de démarrage du client** » peut avoir différentes causes :

- TCP/UDP : impossible d'établir une connexion avec le serveur
- RTU/ASCII : impossible de démarrer l'écoute sur la ligne série (numéro de port invalide → erreur interne)

Une « **Erreur de support** » peut se produire si le nombre de supports Saia PCD® auxquels il faut accéder dépasse les limites définies dans 7.2 (Paramètre « Comp-tage ») ou si une erreur se produit lors de l'accès au support Saia PCD® (support Saia PCD® invalide, adresse du support Saia PCD® invalide, etc.).

3 Serveur Modbus SBC

Du côté serveur, les principales étapes de communication sont les suivantes :

1. Création d'une instance serveur
2. Traitement de la requête reçue
3. Envoi de la réponse

Le serveur Modbus SBC est accessible via TCP/UDP ou une ligne série (RTU/ASCII).

3

Saia PCD®

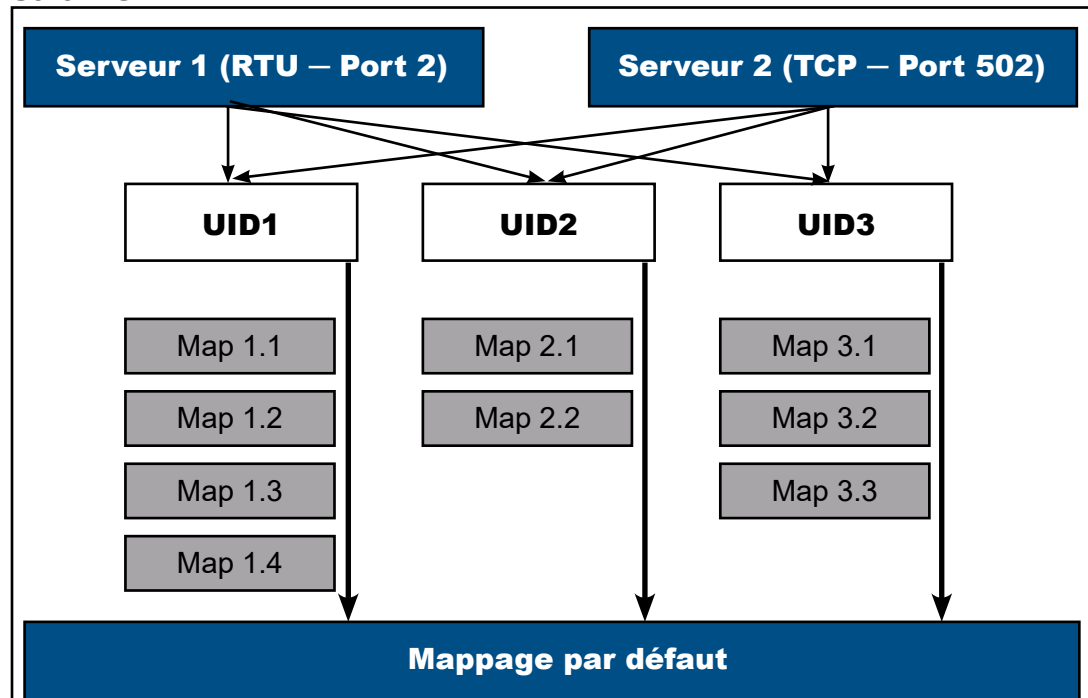


Figure 9 : Architecture Modbus générale d'une station serveur Saia PCD®

3.1 Instance serveur

Plusieurs instances serveur peuvent être définies sur un Saia PCD®, ce qui permet plusieurs accès via ligne série ou IP.

Un serveur est défini en tant que paire {Port – Protocole}. En mode série, il n'est pas possible d'utiliser plusieurs serveurs sur une interface matérielle (p.e. RTU sur le port 1, ASCII sur le port 2, etc.). Pour TCP / UDP, le port est un port logique ; plusieurs serveurs peuvent être définis sur le même port ou sur le même protocole (TCP sur le port 502, UDP sur le port 502, TCP sur le port 503, etc.). De plus, il n'est pas possible de définir 2 instances de serveur identiques.

Les serveurs sont accessibles via des connexions. En mode série et UDP, une connexion par instance serveur est utilisée. En mode TCP, une connexion par client connecté est nécessaire.

Des drapeaux et registres de diagnostic sont disponibles pour chaque serveur (uniquement utilisé en mode série).

3.2 UID serveur

Le serveur Modbus SBC est également caractérisé par un ou plusieurs identifiants d'unité (UID). Un UID sert de référence à toutes les instances serveur de la station. Un UID est également associé à un mappage de support spécifique et à un traitement de données spécifique. Il est possible de configurer différents UID sur un Saia PCD®. Un UID peut également être considéré comme un numéro de station. Le serveur Modbus traite toutes les requêtes adressées à un de ses UID.

En mode série, le serveur répond uniquement à ses UID définis. En cas d'UID de diffusion (UID 0), le serveur traite la requête pour tous les UID définis. Il est interdit de définir 2 fois le même UID sur un bus série (sauf 0).

En mode TCP/UDP, le serveur répond à toutes les requêtes, avec un code d'exception ou avec une réponse correcte. Il n'y a pas d'UID de diffusion.

Via les multiples UID, il est possible d'utiliser différentes configurations de mappage de support avec différents types de traitement ou de développer le mappage par défaut existant.

Des drapeaux et registres de diagnostic sont disponibles pour chaque UID.

3.3 Mappage des supports

Modbus définit 4 types de support : Bobines, Entrées discrètes, Registres de maintien, Registres d'entrée.

Pour qu'un support Saia PCD® soit accessible sur un serveur Modbus, ces supports Modbus doivent être mappés sur des supports Saia PCD® ; les zones de mappage et les UID permettent à l'utilisateur de configurer le mappage des supports Modbus sur des supports Saia PCD®. **Au moins, un UID doit être défini.**

Les bobines et les entrées discrètes peuvent être mappées uniquement sur des drapeaux ou des entrées/sorties sans aucun traitement.

Les registres de maintien et les registres d'entrée peuvent être uniquement mappés sur les minuteries/compteurs/registres/blocs de données mais l'attention doit être portée au traitement des données. De plus, côté client, les registres peuvent être lus à partir du serveur et enregistrés dans un texte ou un texte peut être lu et écrit dans les registres de serveurs. Ceci a lieu avec les codes de fonction relatifs aux registres de maintien ou d'entrée.

Différents types de traitement peuvent être utilisés côté client ou serveur pour être compatibles avec les périphériques autres que SBC : 16 bits/32bits, 16 bits signés / 16 bits non signés, 32 bits FFP / 32 bits IEEE-754, décalage et, pour les registres 32 bits, permutation et trous.

| | Supports Modbus | | | Supports Saia PCD® | | | |
|--------------|-----------------|-------------------|-------|--------------------|-------------------|-------|--------------|
| Type d'accès | Type de support | Adresse de départ | Plage | Type de support | Adresse de départ | Plage | Type de zone |

Figure 10 : Définition d'une zone de mappage

Le **type de zone** définit le traitement qui sera effectué sur les données reçues :

- Bobines sans traitement spécial
- 16 bits signés (requêtes d'écriture uniquement) : Si valeur HR / IR < 0, définir MSW de Saia PCD® registre sur 0xFFFF
Exemple : HR = 0x8001 à registre Saia PCD® = 0xFFFF 8001
- 16 bits non signés à MSW de registre Saia PCD® défini sur 0x0000
Exemple : HR = 0x8001 à registre Saia PCD® = 0x0000 8001
- 32 bits : 2 HR / IR = 1 registre Saia PCD®, aucun traitement
- 32 bits IEEE : Conversion FFP - IEEE effectuée avant l'envoi de données sur le bus (uniquement utilisé si l'utilisateur a des valeurs FFP, utilisation de 32 bits), conversion IEEE - FFP effectuée avant d'écrire les données dans le support Saia PCD® (uniquement utilisé si l'utilisateur veut la valeur dans FFP, utilisation de 32 bits).

Le schéma suivant montre comment les données sont traitées entre Saia PCD® et Modbus Fram

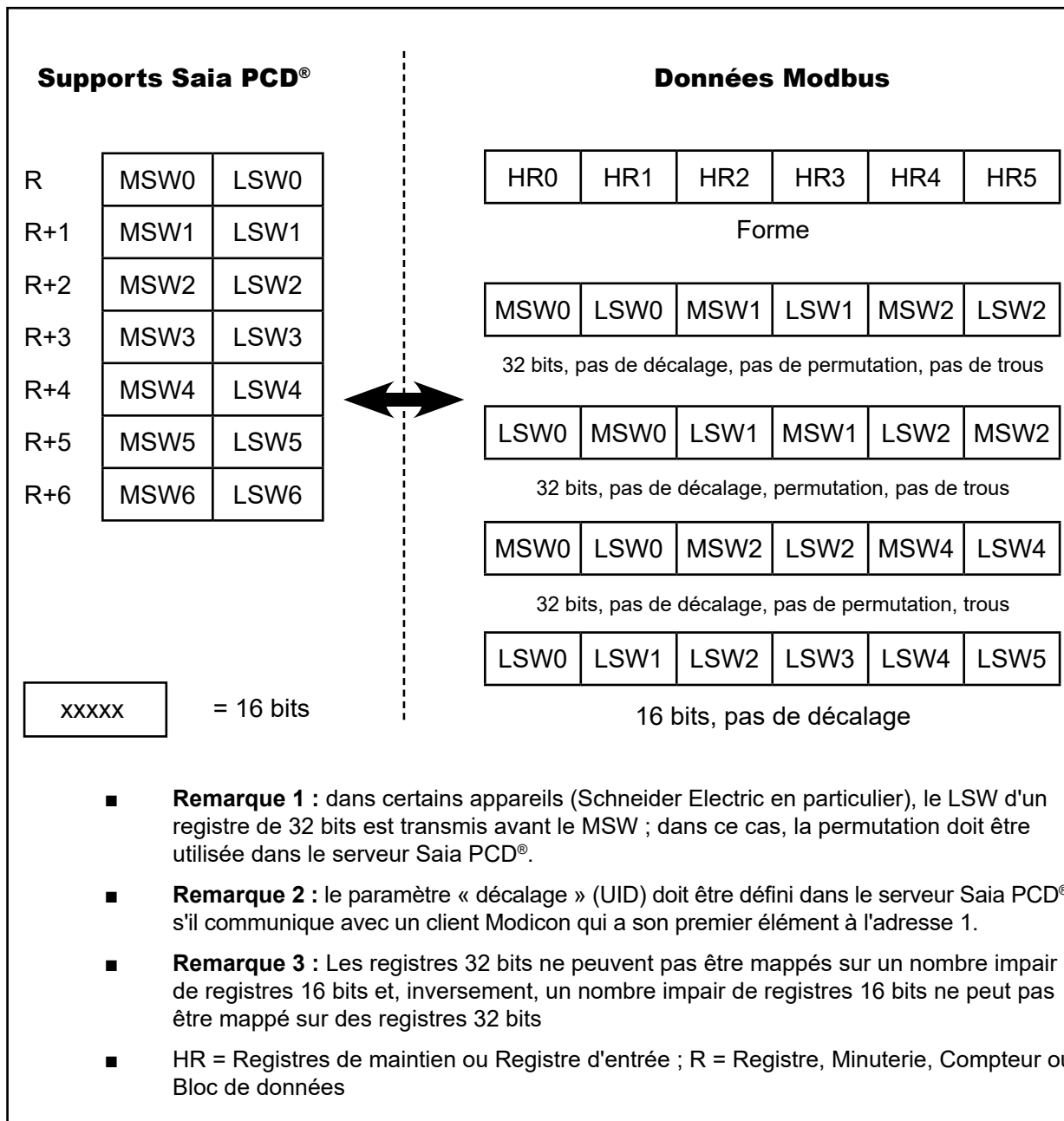


Figure 11 : différents types de traitement pour mapper les registres de maintien Modbus sur les registres Saia PCD® (on suppose que Modbus HR à l'adresse 0 est mappé sur le registre 0 Saia PCD®)

3.4 Mappage par défaut

| Type d'accès | Requête Modbus | | | Saia PCD® | | | Type de zone |
|--------------|----------------------|----------------|-------|-----------------|----------------|-------|-----------------------|
| | Type de support | Adr. de départ | Plage | Type de support | Adr. de départ | Plage | |
| ReadWrite | MB_HOLDING_REG_MEDIA | 1 | 10000 | PCD_REG_MEDIA | 0 | 10000 | MB_AREA_16BITS_SIGNED |
| ReadWrite | MB_HOLDING_REG_MEDIA | 10001 | 10000 | PCD_REG_MEDIA | 0 | 10000 | MB_AREA_32BITS |
| ReadWrite | MB_HOLDING_REG_MEDIA | 20001 | 10000 | PCD_REG_MEDIA | 0 | 10000 | MB_AREA_32BITS_IEEE |
| ReadOnly | MB_INPUT_REG_MEDIA | 1 | 10000 | PCD_TC_MEDIA | 0 | 10000 | MB_AREA_16BITS_SIGNED |
| ReadOnly | MB_INPUT_REG_MEDIA | 10001 | 10000 | PCD_TC_MEDIA | 0 | 10000 | MB_AREA_32BITS |
| ReadWrite | MB_COILS_MEDIA | 1 | 10000 | PCD_FLAG_MEDIA | 0 | 10000 | MB_AREA_COILS |
| ReadOnly | MB_DISCR_INPUT_MEDIA | 1 | 10000 | PCD_IO_MEDIA | 0 | 10000 | MB_AREA_COILS |

4 Client Modbus SBC

La communication Modbus est initialisée par le client Modbus. Sur un Saia PCD®, ceci a lieu à l'aide des canaux Modbus.

Côté client, les principales étapes de communication sont les suivantes :

1. Création d'une instance client
2. Etablissement d'une connexion avec le serveur (TCP)
3. Envoi de la requête
4. Traitement de la réponse

4

Le client Modbus SBC Permet à l'utilisateur d'envoyer des requêtes de lecture et d'écriture Modbus sur TCP/UDP ou sur une liaison série (RTU/ASCII).



Il est interdit d'utiliser 2 clients Modbus sur le même réseau série. Le comportement correct du réseau dans le cas où 2 clients communiquent simultanément sur le bus ne peut pas être garanti.

4.1 Canal client

Plusieurs canaux Modbus peuvent être définis sur un Saia PCD®, ce qui permet une communication via une liaison série ou IP vers différents serveurs Modbus.

Un canal est défini en tant que paire {Port – Protocole}. Le port est soit le numéro de port local en mode série ou le port serveur distant dans TCP / UDP. Par exemple : {RTU – Port 2}, {TCP – Port 502}.

Des drapeaux et registres de diagnostic sont disponibles pour chaque canal. De plus, il est possible de définir 2 délais : un délai de réponse qui représente le délai dans lequel est attendue une réponse du serveur, et un délai de fermeture, qui représente le délai au-delà duquel une connexion doit être fermée si aucune activité n'a lieu.

Le canal est occupé entre l'envoi d'une requête et la réception de la réponse (ou le délai de réponse). Dans cet intervalle de temps, il n'est pas possible d'utiliser le canal pour une autre requête.

Des drapeaux et registres de diagnostic sont disponibles pour chaque canal.

4.2 Requêtes Modbus

Une requête Modbus est envoyée sur un canal spécifié et est adressée à un UID spécifique (et à une adresse IP pour TCP/UDP). Il est également possible de définir le traitement des données qui doit s'appliquer aux données envoyées ou reçues.

Une réponse du serveur est attendue dans un intervalle de temps défini. Si aucune réponse n'est reçue dans cet intervalle de temps, le serveur n'existe pas (ou l'UID n'est pas défini en mode série). En cas d'UID de diffusion (UID 0) sur une liaison série, aucune réponse n'est attendue du client. Elle doit seulement être attendue pour le délai défini avant l'envoi de la requête suivante.

Lors de l'envoi de la requête, il faut définir le traitement qui sera effectué sur les données (avant de les envoyer ou après les avoir reçues) :

- Coils_DI_Processing sans traitement spécial
- 16 bits signés (requêtes de lecture uniquement) : Si valeur HR / IR < 0, définir MSW de Saia PCD® registre sur 0xFFFF

Exemple : HR = 0x8001 à registre Saia PCD® = 0xFFFF 8000

- 16 bits non signés à MSW de registre Saia PCD® défini sur 0x0000

Ex. : HR = 0x8001 à registre Saia PCD® = 0x0000 8001

- 32 bits binaires à 2 HR / IR = 1 registre Saia PCD®, sans traitement
- permutation 32 bits à 2 HR / IR par mot permuté = 1 registre Saia PCD®

Ex. : HR0 = 0x1234, HR1 = 0x5678

Registre Saia PCD® = 0x1234 5678 (32 bits binaires)

Registre Saia PCD® = 0x5678 1234 (permutation 32 bits)

- 32 bits IEEE2FFP - 32 bits FFP2IEEE

Conversion FFP - IEEE effectuée avant l'envoi de données sur le bus (uniquement utilisé si l'utilisateur a des valeurs FFP, utilisation de 32 bits)

Conversion IEEE - FFP effectuée avant d'écrire les données sur le support Saia PCD® (uniquement utilisé si l'utilisateur veut la valeur dans FFP, utilisation de 32 bits)

- Permutation + 32 bits IEEE2FFP / 32 bits FFP2IEEE : permutation par mot effectuée juste avant l'envoi des données / après avoir reçu les données

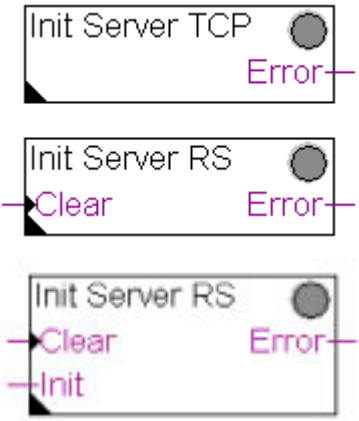
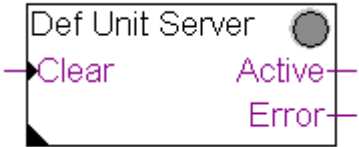
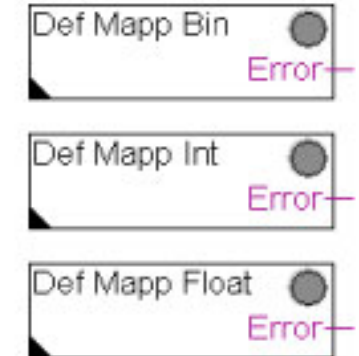
5 Bibliothèque FBox Modbus

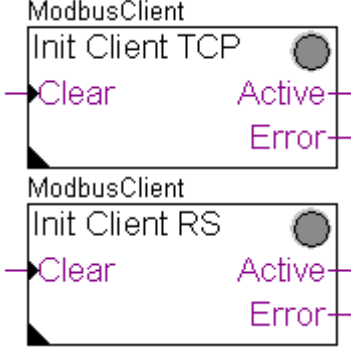
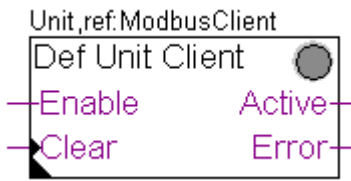
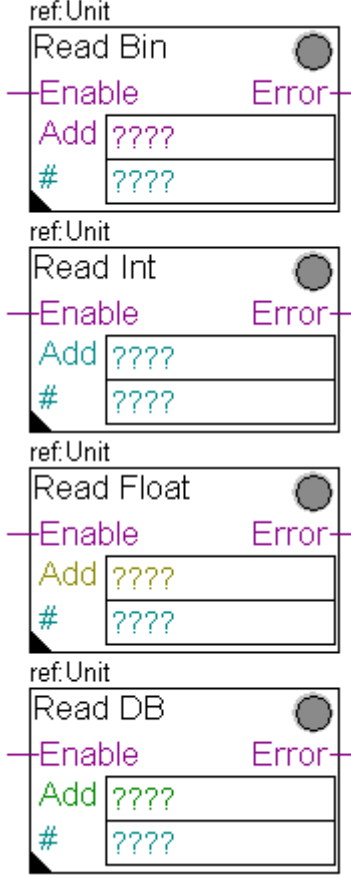
La bibliothèque FBox Modbus permet d'utiliser le protocole Modbus sur un Saia PCD® par des blocs de fonction spécifiques.

La bibliothèque inclut les fonctionnalités suivantes :

- Initialisation du serveur
- Définition d'UID (serveur)
- Définition de mappages (serveur)
- Définition de canaux (client)
- Envoi de requêtes de lecture et d'écriture à partir d'un client

5

| | |
|---|--|
|  | <p>Initialise l'instance serveur :</p> <p>Numéro de port (+ paramètres de port pour liaison série)</p> <p>Mode : TCP, UDP, RTU, ASCII</p> <p>Une autre FBox Extended InitServerRS est disponible pour permettre de détruire une instance serveur série et de fermer le port série associé.</p> |
|  | <p>Définition de l'UID avec des paramètres spécifiques :</p> <p>Décalage, Trous, Mappage, Permutation</p> |
|  | <p>Définit les zones de mappage :</p> <p>UID concerné par le mappage</p> <p>Supports et adresses Modbus mappés</p> <p>Supports et adresses Saia PCD® correspondant aux données Modbus</p> <p>Type de zone</p> |

| | |
|--|---|
|  | <p>Initialise l'instance canal:</p> <ul style="list-style-type: none"> Numéro de canal Numéro de port (+ paramètres de port pour ligne série) Mode : TCP, UDP, RTU, ASCII Délai de fermeture Délai de réponse Essais <p>Une autre FBox Extended InitClientRS est disponible pour permettre de détruire un canal série et de fermer le port série associé.</p> |
|  | <p>Définit le serveur, le client communique avec:</p> <ul style="list-style-type: none"> l'adresse IP pour TCP / UDP UID |
|  | <p>Envoie une requête de lecture (Lire les bobines, les entrées discrètes, les registres de maintien ou les registres d'entrée), traite les données reçues et les stocke dans les supports Saia PCD® définis. Il existe également des FBox supplémentaires pour l'adressage indirect des supports Saia PCD®.</p> |

| | |
|--|--|
| <p>ref.Unit Write Bin Enable Error Add ???? # ???? ref.Unit Write Int Enable Error Add ???? # ???? ref.Unit Write Float Enable Error Add ???? # ???? ref.Unit Write DB Enable Error Add ???? # ???? ref.Unit</p> | <p>Lit les supports Saia PCD® spécifiés, traite les données et les envoie sous forme de requête d'écriture (écrire les registres de maintien ou les bobines). Il existe également des FBox supplémentaires pour l'adressage indirect des supports Saia PCD®.</p> |
|--|--|

Pour plus de détails, voir l'aide en ligne des Saia PG5® FBox.

Si les CSFs sont déplacés à partir du sélecteur de fonction, il n'est pas nécessaire d'inclure un fichier manuellement avec \$INCLUDE; ceci se passe automatiquement en arrière-plan.

Remarquez que les noms de mots-clés ont changé en S.SF.Modbus dans PG5 2.0.

6 Bibliothèque des fonctions système Modbus

6.1 Introduction

Les fonctions système dans un programme utilisateur sont accessibles avec la commande CSF (Call System Function). Le format général de l'appel CSF est le suivant:

```
CSF [cc]    lib_number    ;library number 0-4095/4096-8191
           func_number    ;function number to call
           [parms...]     ;optional parameters
```

PG5.x4

Les valeurs « lib_number » et « function_number » sont requises pour exécuter les commandes CSF. Les mots-clés peuvent être utilisés en incluant le fichier « Modbus.inc » dans PG5.4.

```
$INCLUDE "Modbus.inc"
COB 0
    0 ...
```

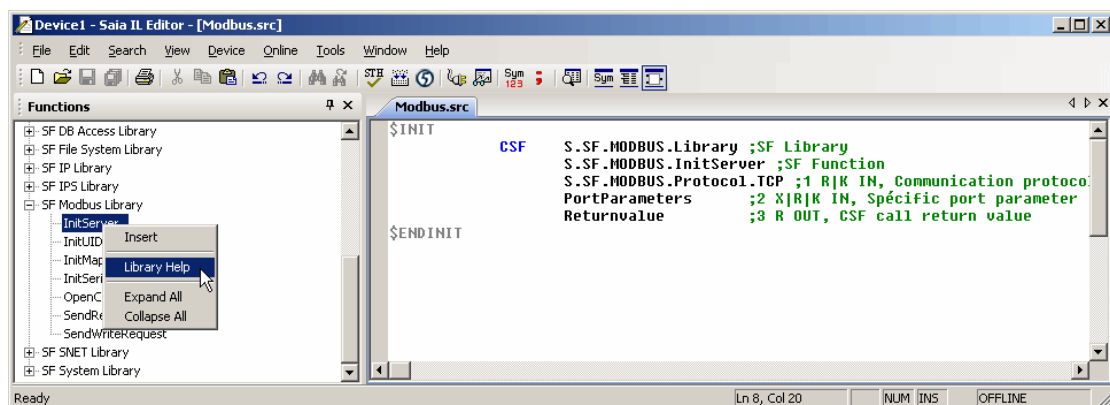
Les appels de la bibliothèque Modbus sont accessibles par le mot-clé de la bibliothèque:

S.Modbus.Library

Les fonctions disponibles sont expliquées dans les chapitres suivants.

PG5 2.0

Dans PG5 2.0, les CFSs peuvent être ajoutés au fichier IL par la fonction glisser-déplacer à partir des sélecteurs de fonction.



6.2 Paramètres

| Nom | Utilisé | Description |
|---------------------|--|---|
| AccessType | InitMap | Définit le moyen d'accéder à la zone de mappage (lecture/écriture). |
| AreaType | InitMap | Définit le traitement par défaut à effectuer sur les données de cette zone avant (écriture) ou après (lecture) l'accès aux supports Saia PCD®. |
| ChannelID | OpenChannel SendReadRequest SendWriteRequest | Une station client peut appliquer un ou plusieurs canaux logiques. Le ChannelID est utilisé pour créer ou modifier un canal ou pour envoyer une requête à un partenaire distant. Plage autorisée: 1-10 |
| CloseTimeout | OpenChannel | Délai pour fermer une connexion après un délai d'inactivité (en secondes) - Uniquement utilisé avec Modbus-TCP. |
| Count | SendReadRequest SendWriteRequest | Nombre d'éléments à lire/écrire (PAS les éléments Modbus : Si requête de lecture d'un registre 32 bits sur un Saia PCD®, Count = 1) Nombre max. : - Lire les bobines / DI : 2000 - Ecrire les bobines : 1968 - Lire HR / IR : 125 - Ecrire HR : 123 en cas de requête 16 bits (respectivement 62 et 61 en cas d'accès 32 bits) |
| Diag | OpenChannel InitUID | Ce paramètre définit l'emplacement des diagnostics et des registres de diagnostic. Format : « DIAG:Fxxx,Ryyy » Fxxx = adresse de base de 8 drapeaux consécutifs Ryyy = adresse du registre de diagnostic Il est recommandé d'effacer tous les supports de diagnostic utilisés au démarrage du programme (XOB16). |
| ExceptCode | SendReadRequest SendWriteRequest | Emplacement où stocker le code d'exception retourné dans la réponse du partenaire. |
| Mode | InitUID | Définit le mode de fonctionnement : 0 = auto : le traitement des requêtes et leur réponse sont automatiques 1 = transparent : le traitement des requêtes et leur réponse sont effectués par l'application utilisateur (pas encore implémentée) |
| PartnerRange | InitMap | Nombre d' éléments Modbus appartenant à la zone de mappage. |
| PartnerStart | SendReadRequest SendWriteRequest InitMap | Première adresse Modbus de la zone de mappage (InitMap) ou première adresse Modbus des données à lire/écrire (SendRequest). Prêter attention aux trous de paramètres (UID) pour accéder au registre de droite. |
| PartnerType | InitMap | Ce paramètre définit le type de données Modbus auxquelles le client a eu accès. |
| Saia PCD® Media | SendReadRequest SendWriteRequest | Requête de lecture : définit à quel emplacement sur le Saia PCD® les données demandées seront copiées. Requête d'écriture : Définit à partir de quel emplacement dans le Saia PCD® les données à envoyer sont prélevées |
| Saia PCD® MediaType | InitMap | Type de support Saia PCD® sur lequel les données Modbus de type « PartnerType » seront mappées. |

| | | |
|-----------------|------------------------------|--|
| Saia PCD® Range | InitMap | Nombre d'éléments Saia PCD® dans la zone de mappage. En cas de blocs de données, nombre d'éléments de ce bloc de données dans la zone ; l'adresse du 1er élément est 0. |
| Saia PCD® Start | InitMap | Adresse de démarrage du Saia PCD® correspondant au démarrage du partenaire. En cas de blocs de données, nombre de blocs de données concernés par la zone. |
| Port | OpenChannel | TCP/UDP : Numéro de port logique auquel la requête sera envoyée RTU/ASCII : numéro de port série utilisé pour l'envoi |
| PortParam | InitServer InitSerialPort | TCP/UDP : numéro de port logique utilisé par le serveur RTU/ASCII : texte contenant les paramètres de port série. Format : «<i>PORT:portnumber; UART:baudrate,data,parity,stobits;LINE:linetype</i>» - PortNumber est le numéro du port série - Baudrate est le débit en bauds utilisé sur ce port série - Data, Parity, Stobits ; Data doit être égal à 8 pour répondre à la norme Modbus en mode RTU et à 7 en mode ASCII. Les bits de parité et d'arrêt doivent également être configurés de sorte que le nombre total de bits dans un caractère soit égal à 11 en mode RTU et à 10 en mode ASCII (à savoir N,2 ou E,1 ou O,1) - LineType peut être : - SL0 : RS-232 - SL1 : RS-422 - SL2 : RS-232, pas d'établissement de liaison - SL3 : RS-422, pas d'établissement de liaison - SL4 : RS-485, RTS auto Pour InitServer : ce texte doit être amélioré avec « DIAG:Fxxx,Ryyy » pour indiquer à quel endroit stocker les diagnostics associés à l'instance serveur. Il est recommandé d'effacer les drapeaux et registres de diagnostic au démarrage du programme. Pour fermer un port : Format : «<i>PORT:portnumber;OFF</i> » |

| | | |
|---------------|--|---|
| Process | InitUID | <p>Définit le traitement à effectuer sur les données</p> <p>Format : « OFFSET=0, SWAP=1, DEFMAP=1, HOLES=0 »</p> <p>OFFSET =1 pour la communication avec un client Modicon (premier élément à l'adresse 1) ; OFFSET =0 pour les autres périphériques avec le premier registre à l'adresse 0 (Saia PCD®, etc.).</p> <p>SWAP=1 si une permutation par mot doit être effectuée sur les données 32 bits (cf. Figure 11)</p> <p>DEFMAP : définit dans quelle tables de mappage rechercher les adresses Modbus</p> <p>DEFMAP=0 : seul le mappage défini par l'utilisateur doit être utilisé.</p> <p>DEFMAP=1 : première recherche dans le mappage défini par l'utilisateur. Si la recherche n'a rien donné, rechercher dans le mappage par défaut (voir DefaultMapping).</p> <p>HOLES : définit comment les registres Saia PCD® sont mappés sur HR/IR.</p> <p>HOLES=0 : pour l'accès 32 bits, la requête HR0 met à jour HR0 et HR1 avec le contenu de Reg0, la requête HR2 met à jour HR2 et HR3 avec le contenu de Reg1.</p> <p>HOLES=1 : pour l'accès 32 bits, la requête HR0 met à jour HR0 et HR1 avec le contenu de Reg0, la requête HR2 met à jour HR2 et HR3 avec le contenu de Reg2.</p> <p>(cf. Figure 11)</p> |
| Processing | SendReadRequest SendWriteRequest | <p>Requête de lecture : définit le traitement à effectuer sur les données reçues avant qu'elles ne soient écrites sur le support Saia PCD®</p> <p>Requête d'écriture : définit le traitement à effectuer sur les données avant qu'elles ne soient copiées dans la requête</p> |
| Protocol | InitServer OpenChannel | Protocole utilisé pour la communication Modbus. |
| RemotelIPAddr | SendReadRequest SendWriteRequest | Adresse IP du serveur à laquelle une requête est envoyée (TCP/UDP) |
| ReqType | SendReadRequest SendWriteRequest | Code de fonction Modbus. Ce paramètre définit le type de requête (Lecture/Ecriture, Bobines/Registres de maintien, etc.). |
| RespTimeout | OpenChannel | Délai dans lequel une réponse doit être reçue (en millisecc.) |
| Retries | OpenChannel | Nombre d'essais en cas de délai de réponse (1er essai inclus) |
| Status | InitMap InitServer InitUID OpenChannel SendReadRequest SendWriteRequest | <p>Etat après avoir exécuté le CSF.</p> <p>Si la valeur est inférieure à 0, une erreur s'est produite pendant l'appel. Voir la liste des codes d'erreur</p> <p>Lorsque la fonction a été exécutée avec succès, le statut est >= 0</p> |

| | | |
|-----|---|--|
| UID | SendReadRequest SendWriteRequest InitUID InitMap | <p>L'identifiant d'unité (UID) est inséré dans la requête. Lorsqu'une requête est reçue, l'UID est utilisé pour trouver quel mode doit être utilisé et quel traitement sera effectué sur les données. Plusieurs UID peuvent être définis sur une station serveur. Les UID sont désignés par toutes les instances serveur sur la station. Il est recommandé d'utiliser un seul UID sur une station en mode série.</p> <p>En mode série, si une requête avec l'UID 0 (UID de diffusion) est reçue, tous les UID définis réagissent à cette requête avec leur propre traitement et mappage.</p> <p>En mode TCP/UDP, l'UID 0 et l'UID 255 correspondent au même UID. Pour accéder à cet UID sur un serveur TCP/UDP, l'utilisateur doit configurer ses paramètres. Ceci peut être effectué à l'aide du CSF InitUID.</p> <p>Il est recommandé d'éviter d'accéder à un serveur TCP/UDP avec l'UID 0.</p> |
|-----|---|--|

6.3 Codes d'erreur

En cas de traitement réussi, tous les CSF retournent à 0 (zéro) ou une valeur positive. Une valeur négative indique une erreur. Voici une liste des codes d'erreur :

6.3.1 Codes d'erreur CSF

La plupart de ces erreurs peuvent apparaître lorsqu'un ou plusieurs paramètres CSF sont incorrects.

| Code | Désignation | Description |
|-------|---------------------------|---|
| 0 | NO_ERROR | Pas d'erreur |
| -4200 | INVALID_MEDIATYPE | Le type de support PCD est incorrect (requête d'envoi) |
| -4199 | DIAGTEXT_ERROR | Impossible de lire le texte de diagnostic (canal/UID) (interne) |
| -4198 | INVALID_DIAG_SYNTAX | Erreur de syntaxe dans le texte de diagnostic (canal/UID) |
| -4197 | INVALID_PROCESSING_PARAM | Au moins 1 paramètre dans le texte de traitement est incorrect (UID) |
| -4196 | INVALID_PROCESSING_SYNTAX | Erreur de syntaxe dans le texte de traitement (UID) |
| -4195 | PROCESSINGTEXT_ERROR | Impossible de lire le texte de traitement (UID) (interne) |
| -4194 | CHANNEL_NOT_DEFINED | Le canal n'est pas défini (requête d'envoi) |
| -4193 | CHANNEL_BUSY | Le canal est déjà occupé (requête d'envoi) |
| -4192 | SERVER_NOT_CREATED | Impossible de créer le serveur (InitServer) : nombre max. de serveurs déjà atteint ou serveur déjà défini. |
| -4191 | SERVER_NOT_STARTED | Impossible de démarrer le serveur (Init Server) |
| -4190 | INVALID_FUNCTION_CODE | Le code de fonction ne correspond pas au type de requête (lecture/écriture) |
| -4189 | INVALID_PROCESSING | Le traitement ne correspond pas au code de fonction ou au type de requête (lecture/écriture) |
| -4188 | INVALID_PROTOCOL | Le protocole est incorrect (serveur/canal) |
| -4187 | INVALID_PARTNER_TYPE | Le support Modbus est incorrect (Init Map) |
| -4186 | INVALID_PCDMEDIA_TYPE | Le support PCD est incorrect (Init Map) |
| -4185 | INVALID_AREA_TYPE | Le type de zone est incorrect (Init Map) |
| -4184 | INVALID_PCD_MEDIA_ADDR | L'adresse du support PCD est incorrecte (Init Map) |
| -4183 | INVALID_RANGE | Plage de support incorrecte (Init Map) : Plage de partenaire < plage PCD ou plage = 0. |
| -4182 | INVALID_AREA_ACCESS_TYPE | Le type d'accès est incorrect (Init Map) : n'existe pas ou ne correspond pas au support Modbus. |
| -4181 | INVALID_NB_OF_MEDIA | Le nombre de supports ne correspond pas au code de fonction (requête d'envoi) : > 1 pour écrire un seule bobine ou un seul registre. |
| -4180 | INVALID_UID | UID en dehors de la plage 0-0xFF (Init UID, Send Request) ou UID = 0 pour une requête de lecture en mode série |
| -4179 | INVALID_CHANNEL | Canal en dehors de la plage 1-10 (Open Channel, Send Request) |
| -4178 | INCOMPATIBLE_PCDMEDIA | Le support PCD est incompatible avec le support Modbus (Init Map) ou le code de fonction (Send Request) |
| -4177 | INVALID_PORT_CONF | La configuration du port (texte ou numéro) ne correspond pas au protocole (Init Server) |
| -4176 | PORTCONFIGTEXT_ERROR | Impossible de lire le texte de configuration du port en mode série (Init Server, Init SR Port) (interne) |
| -4175 | PORTCONFIGPARAM_ERROR | Erreur dans le texte de configuration du port en mode série (Init Server, Init SR Port) |
| -4174 | SRPORT_START_ERROR | Impossible d'ouvrir le port SR (Init Server, Init SR Port). En série, le port est peut-être déjà utilisé ou les paramètres de port sont incorrects. |
| -4173 | INVALID_PORT_NUMBER | Le numéro de port SR est incorrect (Open Channel) |
| -4172 | INVALID_RETRY_NB | Nombre d'essais incorrect (doit être < 16) |
| -4171 | NO_PORT_TO_CLOSE | Le port série à fermer n'a pas été ouvert |

6.3.2 Codes d'erreur ModbusShell

Ces erreurs apparaissent lors du traitement CSF.

| Code | Désignation | Description |
|-------|-------------------------|---|
| -4100 | UID_TABLE_FULL | Le nombre maximum d'UID est déjà atteint (Init UID) |
| -4099 | UID_NOT_DEFINED | L'UID n'est pas défini (Init Map) |
| -4098 | UID_BUSY | L'UID est utilisé (Init UID), impossible de le modifier |
| -4097 | CHANNEL_TABLE_FULL | Le tableau des canaux est plein ; impossible d'ajouter le canal |
| -4096 | CHANNEL_ALREADY_DEFINED | Un canal avec le même port / protocole est déjà défini (Open Channel) |
| -4095 | CHANNEL_BUSY | Ce canal est utilisé, impossible de le modifier |
| -4094 | MAP_TABLE_FULL | Nombre maximum de mappages atteint pour l'UID (Init Map) |
| -4093 | MAP_AREA_OVERLAP | Au moins 2 zones de mappage se chevauchent (Init Map) |
| -4092 | CONNECTION_TABLE_FULL | Le nombre maximum de « connexions » est atteint (Requête d'envoi). TCP/UDP : connexion = {Port-Protocole-Adresse IP} Série : connexion = {Port-Protocole} |
| -4091 | CONNECTION_NOT_READY | La connexion n'est pas prête pour l'envoi (requête d'envoi) : adresse IP non définie, etc. |

6

6.3.3 Codes d'erreur ModbusDriver

La plupart des erreurs entre -3980 et -4000 sont internes. Si une erreur non listée ci-après se produit, contacter Saia Burgess Controls.

| Code | Désignation | Description |
|-------|-----------------------|--|
| -3990 | CLII_DRIVER_NOT_READY | Le pilote n'est pas prêt (requête d'envoi) |
| -3987 | QUANTITY_ERROR | Erreur dans la quantité de supports à lire / écrire (requête d'envoi) : dépasse la limite ou < 0 |
| -3983 | CLIENT_NO_RESSOURCE | Aucune ressource disponible (requête d'envoi) |
| -3982 | PORT_ID_ERROR | Port déjà initialisé ou utilisé |
| -3981 | INVALID_PORT_STATE | Port pas prêt (série) (requête d'envoi) |
| -3980 | UNKNOWN_ERROR | Erreur inconnue |

6.4 Codes de fonction

| PG5 2.0 | PG5 1.4 | EQU |
|---------------------------|------------------------|-------|
| S.SF.Modbus.FunctionCode. | S.Modbus.FunctionCode. | [hex] |
| ReadCoils | ReadCoils | 0x01 |
| ReadDiscreteInput | ReadDiscreteInput | 0x02 |
| ReadHoldingReg | ReadHoldingReg | 0x03 |
| ReadInputReg | ReadInputReg | 0x04 |
| WriteSingleCoil | WriteSingleCoil | 0x05 |
| WriteSingleReg | WriteSingleReg | 0x06 |
| WriteMultipleCoils | WriteMultipleCoils | 0x0F |
| WriteMultipleRegs | WriteMultipleRegs | 0x10 |

6

6.5 Protocoles

| PG5 2.0 | PG5 1.4 | EQU |
|-----------------------|--------------------|-----|
| S.SF.Modbus.Protocol. | S.Modbus.Protocol. | |
| TCP | TCP | 0 |
| UDP | UDP | 1 |
| RTU | RTU | 2 |
| ASCII | ASCII | 3 |

6.6 Types de données Modbus

| PG5 2.0 | PG5 1.4 | EQU |
|----------------------|-------------------|-----|
| S.SF.Modbus.MBMedia. | S.Modbus.MBMedia. | |
| NoMedia | NoMedia | 0 |
| Coils | Coils | 1 |
| DiscrInput | DiscrInput | 2 |
| HoldingReg | HoldingReg | 3 |
| InputReg | InputReg | 4 |

6.7 Types de zones Modbus (serveur)

| PG5 2.0 | PG5 1.4 | EQU |
|-----------------------|---------------------|-----|
| S.SF.Modbus.AreaType. | S.Modbus.AreaType. | |
| Coils | Coils | 0 |
| Reg_16bits_signed | Reg_16bits_signed | 1 |
| Reg_16bits_unsigned | Reg_16bits_unsigned | 2 |
| Reg_32bits | Reg_32bits | 3 |
| Reg_32bits_IEEE | Reg_32bits_IEEE | 4 |

6.8 Types d'accès aux zones Modbus (serveur)

| PG5 2.0 | PG5 1.4 | EQU |
|--------------------------------|-----------------------------|-----|
| S.SF.Modbus.AreaAccess. | S.Modbus.AreaAccess. | |
| ReadWrite | ReadWrite | 0 |
| ReadOnly | ReadOnly | 1 |
| WriteOnly | WriteOnly | 2 |
| NoAccess | NoAccess | 3 |

6.9 Types de traitement (client)

| PG5 2.0 | PG5 1.4 | EQU |
|--------------------------------------|-----------------------------------|-----|
| S.SF.Modbus.ClientProcessing. | S.Modbus.ClientProcessing. | |
| Coils_DI_Processing | Coils_DI_Processing | 0 |
| R_16bits_signed | R_16bits_signed | 1 |
| RW_16bits_unsigned | RW_16bits_unsigned | 2 |
| RW_32bits_binary | RW_32bits_binary | 3 |
| RW_32bits_swap | RW_32bits_swap | 4 |
| R_32bits_IEEE2FFP | R_32bits_IEEE2FFP | 5 |
| R_32bits_swap_IEEE2FFP | R_32bits_swap_IEEE2FFP | 6 |
| W_32bits_FFP2IEEE | W_32bits_FFP2IEEE | 7 |
| W_32bits_swap_FFP2IEEE | W_32bits_swap_FFP2IEEE | 8 |

6

6.10 Types de supports Saia PCD®

| PG5 2.0 | PG5 1.4 | EQU |
|------------------------------|---------------------------|-----|
| S.SF.Modbus.PCDMedia. | S.Modbus.PCDMedia. | |
| NoMedia | NoMedia | 0 |
| IO | IO | 1 |
| Input | Input | 2 |
| Output | Output | 3 |
| Flag | Flag | 4 |
| TC | TC | 5 |
| Reg | Reg | 6 |
| DB_Media | DB_Media | 9 |

6.11 Codes d'exception

Codes renvoyés par un serveurs dans le cas où la commande ne puet pas être exécutée.

| | | |
|------------------------|----|---|
| ILLEGAL_FUNCTION_CODE | 1 | Function code not valid or does not correspond to Modbus media type |
| ILLEGAL_DATA_ADDRESS | 2 | Modbus media at received address cannot be accessed (check UID mapping/ see diagnostic) |
| ILLEGAL_DATA_VALUE | 3 | Number of data in the request is invalid |
| SERVER_FAILURE | 4 | PCD Media could not be accessed (see diagnostic) or UID not configured (in particular UID 255 in TCP/UDP) |
| ACKNOWLEDGE | 5 | |
| SERVER_BUSY | 6 | Server is already busy when trying to access it |
| GATEWAY_PATH_NOT_FOUND | 10 | UID not defined. |
| GATEWAY_TARGET_PROBLEM | 11 | |

7 Spécification CSF Modbus

7.1 S.SF.Modbus.InitSerialPort (Client / Server)

| | | | | |
|----------------------------|-----------|--|-----------------------------------|-----------------------------------|
| S.SF.Modbus.InitSerialPort | | <p>Cette fonction ouvre un port série côté client. Côté serveur, elle n'est pas nécessaire car l'initialisation du port a lieu lors de la création du serveur.</p> <p>Pour le client et le serveur, cette fonction peut également être utilisée pour fermer un port série et toutes les instances associées (canaux et serveurs). Pour l'envoi/la réception sur le port série qui a été fermé, les fonctions d'initialisation appropriées doivent être appelées (InitServer, InitChannel, InitSerialPort) en premier à nouveau.</p> <p>L'usage multiple du même port série est interdit.</p> | | |
| Paramètres | | 8 | | |
| 1 | PortParam | X | IN | Paramètres du port série à ouvrir |
| 2 | Status | R | OUT | Valeur retournée (voir ci-après) |
| Valeur retournée | | | | |
| | Success | = 0 | Canal créé ou modifié avec succès | |
| | Error | < 0 | voir chapitre 6.3 | |

7

| | |
|----------------|---|
| Exemple | Initialization of a serial port |
| CSF | <pre>S.SF.Modbus.Library S.SF.Modbus.InitSerialPort ConfigPort2 ; Text: "PORT:2; ART:9600,8,N,2;LINE:SL4" R_StatusOpenPort ; R 10101</pre> |
| Exemple | Fermeture d'un port série |
| CSF | <pre>S.SF.Modbus.Library S.SF.Modbus.InitSerialPort ConfigPort2 ; Text: "PORT:2; OFF" R_StatusOpenPort ; R 10101</pre> |

7.2 S.SF.Modbus.OpenChannel (Client)

| | | | | |
|-------------------------|--------------|---|-----------------------------------|---|
| S.SF.Modbus.OpenChannel | | <p>Cette fonction crée un nouveau canal ou modifie un canal existant (s'il n'est pas actuellement occupé). Un canal est utilisé pour envoyer une requête à un partenaire et pour traiter les données reçues. Un canal peut être utilisé sur n'importe quelle interface donnée (série/Ethernet).</p> <p>Il est interdit d'ouvrir 2 canaux avec la même paire Port/Protocole. Lors de la tentative d'ouvrir un canal avec un ID de canal déjà défini, l'« ancien » canal est écrasé (s'il n'est pas actuellement utilisé, voir TBSY).</p> <p>Il est interdit d'utiliser 2 clients Modbus sur le même réseau série. Le comportement correct du réseau dans le cas où 2 clients communiquent simultanément sur le bus ne peut pas être garanti.</p> | | |
| Paramètres | | 8 | | |
| 1 | ChannelID | R K | IN | Identifiant du canal à créer ou modifier. Plage autorisée : 1-10 |
| 2 | Port | R K | IN | TCP/UDP : Port distant auquel la requête sera envoyée RTU/ASCII : Numéro de port série utilisé pour l'envoi |
| 3 | Protocol | R K | IN | Protocole utilisé pour la communication sur ce canal |
| 4 | Diag | X | IN | Définition du diagnostic. « DIAG:Fxxxx,Ryyyy » |
| 5 | CloseTimeout | R K | IN | Délai pour fermer une connexion après un délai d'inactivité. Ce paramètre définit le temps après lequel une liaison TCP se referme, si aucune donnée n'est transmise. Si la valeur est nulle, la liaison est automatiquement fermée après chaque demande (Firmware 1.16.xx ou supérieur requis). Ceci peut être utilisé si plus de 10 serveurs Modbus TCP doivent être reliés ensemble. Le nombre des cadres Ethernet générés est augmenté, en raison des liaisons créées. |
| 6 | RespTimeout | R K | IN | Délai dans lequel une réponse doit être reçue - Cette valeur est également utilisée comme « délai de diffusion » |
| 7 | Retries | R K | IN | Nombre d'essais autorisés en cas de délai de réponse sur ce canal |
| 8 | Status | R | OUT | Valeur retournée (voir ci-après) |
| Valeur retournée | | | | |
| | Success | = 0 | Canal créé ou modifié avec succès | |
| | Error | < 0 | voir chapitre 6.3 | |

7

| Exemple | |
|---------|-------------------------------------|
| CSF | S.SF.Modbus.Library |
| | S.SF.Modbus.OpenChannel |
| | 2 ; Channel ID |
| | 502 ; Remote Port |
| | S.SF.Modbus.Protocol.TCP ; Protocol |
| | Diagnostic ; «DIAG:F1000,R1000» |
| | 2000 ; CloseTimeout |
| | 1000 ; ResponseTimeout |
| | 0 ; Number of retries |
| | R_StatusOpenCh ; R 10105 |

7.3 S.SF.Modbus.SendReadRequest / S.SF.Modbus.SendWrite-Request (Client)

| | | | | |
|------------------------------|-----------------|---|-----|--|
| S.SF.Modbus.SendReadRequest | | Cette fonction envoie une requête de lecture Modbus sur un canal donné. | | |
| S.SF.Modbus.SendWriteRequest | | Cette fonction envoie une requête d'écriture Modbus sur un canal donné. | | |
| | | Pour les requêtes d'envoi, le drapeau TBSY du canal doit être bas pour envoyer la requête ; le cas échéant, le CSF retourne une erreur. | | |
| Paramètres | | 12 | | |
| 1 | ChannelID | R K | IN | Identifiant de canal (un des canaux déjà ouverts) : indique sur quel canal est envoyée la requête |
| 2 | RemoteIPAddr | R K | IN | Adresse IP du serveur à laquelle la requête est envoyée (uniquement utilisée en mode TCP/UDP) |
| 3 | UID | R K | IN | Identifiant d'unité : utilisé pour définir quel traitement et quel mode doit être utilisé par le serveur |
| 4 | ReqType | R K | IN | Code de fonction |
| 5 | ExceptCode | R | OUT | Emplacement où est copié le code d'exception de la réponse du partenaire |
| 6 | Saia PCD® Media | R T C I O F X D B | IN | Requête de lecture : définit sur quel support Saia PCD® et à quel emplacement les données demandées seront copiées Requête d'écriture : définit à partir de quel support Saia PCD® et à quel emplacement les requêtes seront prélevées |
| 7 | PartnerStart | R K | IN | Définit à partir de quel support de serveur les données seront écrites/lues (dépend du mappage du serveur) |
| 8 | Count | R K | IN | Nombre d'éléments lus/écrits de/vers le partenaire (PAS les éléments Modbus : si les registres 32 bits sont demandés, nombre de registres 32 bits) |
| 9 | Processing | R K | IN | Requête de lecture : définit quel traitement sera effectué sur les données reçues avant qu'elles ne soient écrites sur le support Saia PCD®. Requête d'écriture : définit le traitement à effectuer sur les données avant qu'elles ne soient copiées dans la requête. |
| 10 | Status | R | OUT | Valeur retournée (voir ci-après) |
| Valeur retournée | | | | |
| | Success | = 0 | | Requête envoyée avec succès |
| | Error | < 0 | | voir chapitre 6.3 |

| Exemple | |
|---------|--|
| CSF | <pre> S.SF.Modbus.Library S.SF.Modbus.SendReadRequest ; Function = Send Read Request 2 ; ChannelID RegisterContainingIP ; IP address = 172.23.2.129 = 0AC170281H 1 ; Remote UnitID S.SF.Modbus.FunctionCode.ReadCoils R 100 ; Location to store Exception Code F 0 ; PCD Local Media 10000 ; PartnerStartAddress 10 ; Number of Media S.SF.Modbus.ClientProcessing.Coils_DI_Processing R 101 ; Location to store CSF Status </pre> |

7.4 S.SF.Modbus.InitServer (Server)

7

| | | | | |
|------------------------|-----------|---|-----|--|
| S.SF.Modbus.InitServer | | Cette fonction permet de créer une instance serveur. L'usage multiple du serveur avec la même paire Port/Protocole est interdit en mode TCP/UDP (uniquement le même port en série) | | |
| Paramètres | | 3 | | |
| 1 | Protocol | R K | IN | Protocole de communication utilisé par l'instance serveur |
| 2 | PortParam | R K X | IN | TCP/UDP : numéro de port de communication logique utilisé par l'instance serveur RTU/ASCII : Texte contenant les paramètres de port série |
| 3 | Status | R | OUT | Valeur retournée (voir ci-après) |
| Valeur retournée | | | | |
| | Success | = 0 | | Serveur démarré avec succès |
| | Error | < 0 | | voir chapitre 6.3 |

| Exemple (TCP/UDP) | |
|-------------------|--|
| CSF | <pre> S.SF.Modbus.Library S.SF.Modbus.InitServer S.SF.Modbus.Protocol.UDP 502 ; Port R 102 ; Location to store CSF status </pre> |

| Exemple (SERIE) | |
|-----------------|--|
| CSF | <pre> S.SF.Modbus.Library S.SF.Modbus.InitServer S.SF.Modbus.Protocol.RTU ConfigPort1 ; Port configuration: ; "PORT:1; UART:9600,8,N,2;LINE:SL4 ;DIAG:F800,R800" R 102 ; Location to store CSF status </pre> |

7.5 S.SF.Modbus.InitUID (Server)

| | | | | |
|---------------------|---------|--|-----|---|
| S.SF.Modbus.InitUID | | <p>Cette fonction crée ou modifie un identifiant d'unité qui est désigné par toutes les instances serveur. Lorsqu'une requête est reçue, l'UID est utilisé pour trouver quel mode doit être utilisé et quel traitement sera effectué sur les données. Plusieurs UID peuvent être définis sur une station serveur.</p> <p>Lors de la tentative de créer un ID avec un numéro d'UID déjà défini, l'« ancien » est écrasé (s'il n'est pas actuellement utilisé, voir RBSY).</p> | | |
| Paramètres | | 5 | | |
| 1 | UID | R K | IN | Identifiant d'unité |
| 2 | Mode | R K | IN | Mode serveur : 0 = auto, 1 = transparent (non implémenté actuellement) |
| 3 | Diag | X | IN | Définition du diagnostic. Format : « DIAG:Fxxx,Ryyy » |
| 4 | Process | X | IN | Définition du traitement des données. Format : « OFFSET=0, SWAP=1, DEFMAP=1, HOLES=0 » |
| 5 | Status | R | OUT | Valeur retournée (voir ci-après) |
| Valeur retournée | | | | |
| | Success | = 0 | | UID ajouté avec succès |
| | Error | < 0 | | voir chapitre 6.3 |



| Exemple | | |
|---------|-------------------------|---|
| CSF | S.SF.Modbus.Library | |
| | S.SF.Modbus.InitUID | |
| | 1 | ; UID |
| | S.SF.Modbus.ServerMode. | |
| | Auto | |
| | DiagnosticUID | ; "DIAG:F2000,R2000" |
| | Processing | ; "OFFSET=0, SWAP=0, DEFMAP=1, HOLES=0" |
| | R 103 | ; Location to store CSF status |

7.6 S.SF.Modbus.InitMap (Server)

| | | | | |
|---------------------|------------------------|--|-----|--|
| S.SF.Modbus.InitMap | | <p>Cette fonction permet de définir une zone de mappage ; ainsi, un ensemble de supports Modbus à l'adresse spécifiée est mappé sur un ensemble de supports Saia PCD® à une adresse spécifiée</p> <p>Il est interdit de créer une zone qui en chevauche une autre (même type de support Modbus et plages d'adresses Modbus se chevauchant)</p> | | |
| Paramètres | | 10 | | |
| 1 | PartnerType | R K | IN | Type de support Modbus qui sera mappé sur le support Saia PCD® |
| 2 | PartnerStart | R K | IN | Adresse de départ du partenaire de la zone de mappage |
| 3 | PartnerRange | R K | IN | Nombre d'éléments de la zone de mappage (ne peut pas être égal à 0) |
| 4 | Saia PCD® MediaType | R K | IN | Type de support Saia PCD® sur lequel le support Modbus sera mappé |
| 5 | Saia PCD® Start | R K | IN | <p>Adresse de départ du support Saia PCD® correspondant à l'adresse de départ du partenaire.</p> <p>En cas de mappage sur un bloc de données, nombre de blocs de données sur lesquels les données seront mappées.</p> |
| 6 | Saia PCD® Range | R K | IN | <p>Nombre d'éléments Saia PCD® dans la zone de mappage (doit correspondre au nombre d'éléments de partenaire, tenir compte du type de zone utilisé).</p> <p>En cas de blocs de données, nombre d'éléments de ce bloc de données appartenant à la zone ; le 1er élément de la zone est l'élément 0 du bloc de données.</p> <p>Saia PCD® Range ne peut pas être égal à 0.</p> <p>Pour éviter les erreurs lors de l'accès à la zone, noter que, dans une zone 32 bits, 1 registre Saia PCD® correspond à 2 registres de maintien (voir Figure 12 ci-après).</p> |
| 7 | AreaType | R K | IN | Type de zone de mappage |
| 8 | AccessType | R K | IN | Type d'accès à la zone de mappage |
| 9 | UID | R K | IN | UID concerné par cette zone de mappage |
| 10 | Status | R | OUT | Valeur retournée (voir ci-après) |
| Valeur retournée | | | | |
| | Success | = 0 | | Zone de mappage ajoutée avec succès |
| | Error | < 0 | | voir chapitre 6.3 |

| Exemple | | |
|---------|----------------------------------|---------------------------------|
| CSF | S.SF.Modbus.Library | |
| | S.SF.Modbus.InitMap | |
| | S.SF.Modbus.ModbusType.Coils | |
| | 1000 | ; Partner Start Address |
| | 2000 | ; Partner Range |
| | S.SF.Modbus.PCDMedia.Flag | |
| | 100 | ; PCD Start Address |
| | 2000 | ; PCD Range |
| | S.SF.Modbus.AreaType.Coils | |
| | S.SF.Modbus.AreaAccess.ReadWrite | |
| | 1 | ; UID concerned by this mapping |
| | R 104 | ; Location to store CSF status |

Dans cet exemple, les bobines avec l'adresse Modbus comprise entre 2000 et 3999 sont respectivement mappées sur les drapeaux Saia PCD® 100 à 2099 (pour l'UID 1).

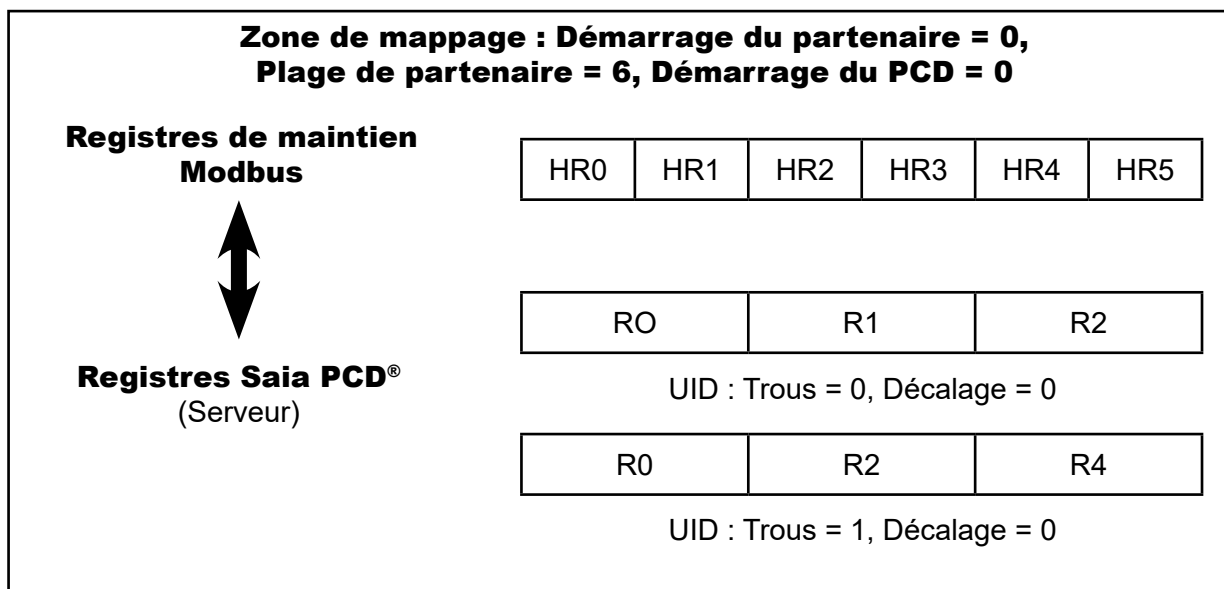


Figure 12 : Mappage des registres de maintien (16 bits) sur des registres Saia PCD® (32 bits) dans une zone 32 bits






Dans la **zone 32 bits** décrite ci-dessus, les éléments Modbus 1, 3 et 5 ne sont pas accessibles : une erreur sera retournée. L'accès à HR2 avec 3 registres Saia PCD® (soit 6 HR) retourne également une erreur.

Pour accéder à HR 0 avec 3 registres Saia PCD® (= 6 HR), la zone de mappage doit être définie avec au moins 6 HR commençant à l'adresse 0 correspondante avec :

- au moins 3 registres Saia PCD® commençant à l'adresse 0 (1er cas)
- au moins 5 registres Saia PCD® commençant à l'adresse 0 (2ème cas)

A Annexe

A.1 Icônes

| | |
|--|--|
|  | <p>Dans les manuels, ce symbole indique au lecteur des informations supplémentaires qui sont contenues dans ce manuel ou dans d'autres manuels ou documents techniques.</p> <p>En règle générale, il n'existe pas de lien direct avec ces documents.</p> |
|  | <p>Ce symbole informe le lecteur du risque de décharges électrostatiques en cas de contact avec les composants.</p> <p>Recommandation : toucher au moins la borne négative du système (armoire du connecteur PGU) avant d'entrer en contact avec les composants électroniques.</p> <p>Il est préférable d'utiliser un bracelet antistatique de terre avec le câble relié à la borne négative du système.</p> |
|  | <p>Ce signe accompagne les instructions qui doivent impérativement être observées.</p> |
|  | <p>Les explications jointes à ce signe ne concernent que pour la série Saia PCD® Classic.</p> |
|  | <p>Les explications jointes à ce signe ne concernent que pour la série Saia PCD® xx7.</p> |

A.2 Liste d'abréviations

| | |
|----------|--|
| UID | Identifiant unique |
| HR | Registres de maintien |
| IR | Registres d'entrée |
| DI | Entrées numériques |
| TCP | Protocole TCP (Transmission Control Protocol) |
| UDP | Protocole UDP (User Datagram Protocol) |
| FFP | Virgule flottante rapide |
| IEEE | Institute of Electrical and Electronics Engineers |
| IEEE 754 | 32 bit floating point format, utilisé par divers non-Saia controllers/system |
| LSW | Mot de poids faible |
| MSW | Mot de poids fort |
| DB | Bloc de données |
| I/O | Entrées/Sorties |
| SR | Série |

A.3 Adresses

Saia-Burgess Controls AG

Bahnhofstrasse 18
3280 Murten, Suisse

Téléphone standard +41 26 580 30 00

Téléphone support SBC..... +41 26 580 31 00

Fax : +41 26 580 34 99

E-mail assistance : support@saia-pcd.com

Page d'assistance : www.sbc-support.com

Page d'accueil SBC : www.saia-pcd.com

Représentations internationales et

succursales SBC : www.saia-pcd.com/contact

Adresse postale pour les retours effectués par les clients pour les ventes en Suisse**Saia-Burgess Controls AG**

Service Après-Vente
Bahnhofstrasse 18
3280 Murten, Suisse

A