

# Description de l'application

## Communication "Modbus RTU" API Siemens – Appareils Studer Innotec



## Table des matières

|       |  |    |
|-------|--|----|
| 1     | Introduction.....  | 1  |
| 2     | Matériel nécessaire .....  | 2  |
| 2.1.1 | Automate programmable Siemens .....  | 2  |
| 2.1.2 | Interface Homme-Machine Siemens.....   | 2  |
| 2.2   | Appareils Studer Innotec.....  | 2  |
| 3     | Logiciels nécessaires Siemens .....  | 2  |
| 4     | Paramètres de communication Modbus RTU .....                                       | 3  |
| 5     | Configuration de l'interface Xcom-485i .....                                       | 4  |
| 6     | Configuration du module CM PtP RS422/485 HF.....                                   | 5  |
| 7     | Câble de liaison entre le module CM PtP RS422/485 HF et l'interface Xcom-485i..... | 6  |
| 8     | Paramètres de communication Profinet .....   | 6  |
| 9     | Programme de l'automate programmable .....   | 7  |
| 9.1   | Blocs de programme .....   | 7  |
| 9.2   | Structure du programme.....  | 7  |
| 9.2.1 | Gestion des opérations de lecture et d'écriture .....                              | 8  |
| 9.2.2 | Séquence de lecture des paramètres depuis l'Xtender et BSP .....                   | 8  |
| 9.2.3 | Séquence d'écriture des paramètres vers l'Xtender .....                            | 8  |
| 9.2.4 | Séquence de lecture des messages depuis l'Xcom-485i .....                          | 8  |
| 9.2.5 | Fonctions de lecture et d'écriture Modbus RTU .....                                | 10 |
| 9.3   | Structure des données .....  | 13 |
| 10    | Programme de l'Interface Homme-Machine (IHM) .....                                 | 15 |

## 1 Introduction

L'objectif de ce document est d'expliquer le fonctionnement de l'application "Modbus\_RTU\_Master" qui permet d'établir une communication via le protocole Modbus RTU (Remote Terminal Unit) entre un automate programmable de Siemens et une interface Xcom-485i qui communique avec un onduleur-chargeur Xtender et un module de batterie BSP de Studer Innotec.

L'ensemble des données qui sont transférées entre l'automate programmable et les appareils de Studer Innotec sont affichées sur une Interface Homme-Machine (IHM) de Siemens.

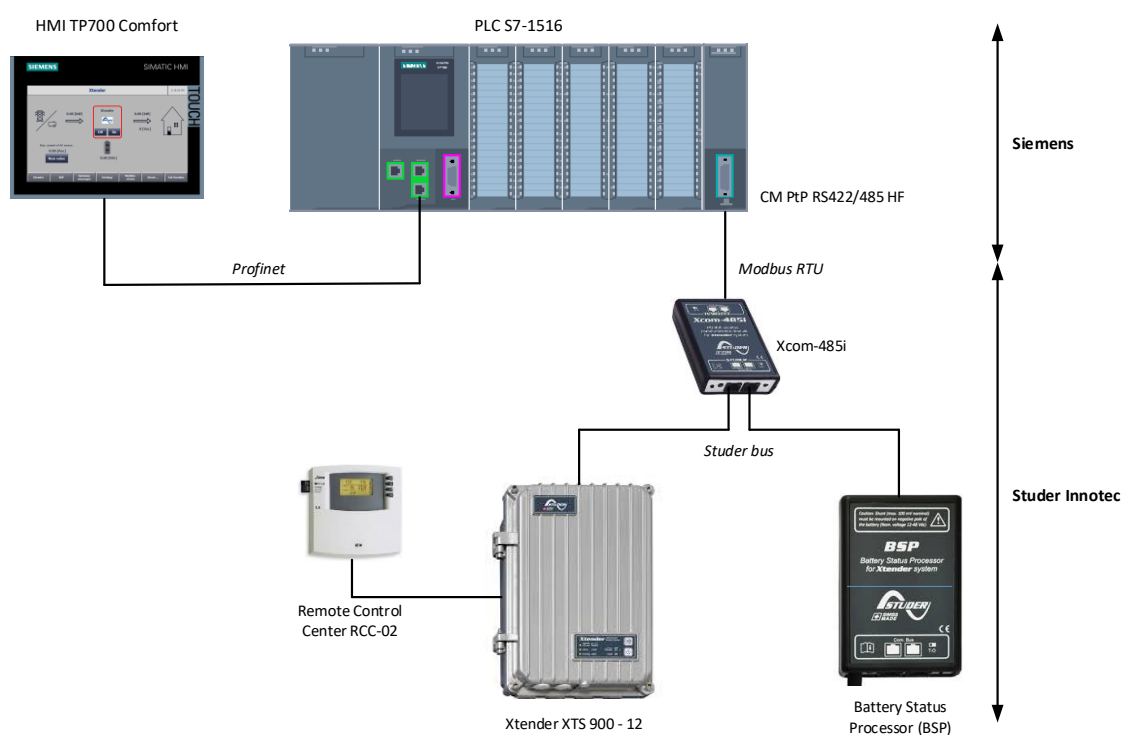


Figure 1 : Vue d'ensemble

Le protocole Modbus RTU utilise un réseau maître/esclave où toutes les communications sont déclenchées par le maître. Dans notre application, les appareils de Studer Innotec sont les esclaves et ils répondent aux requêtes envoyées par l'automate programmable qui est le maître.

Le protocole Profinet permet la communication entre l'automate programmable et l'Interface Homme-Machine (IHM).

## 2 Matériel nécessaire

### 2.1.1 Automate programmable Siemens

| Composant           | Numéro d'article    | Remarque   |
|---------------------|---------------------|--|
| PM 190W 120/230VAC  | 6EP1333-4BA00       |  |
| CPU 1516-3 PN/DP    | 6ES7 516-3AN01-0AB0 | Firmware : V2.5                                  |
| DI 32x24VDC HF      | 6ES7 521-1BL00-0AB0 | <i>Module pas utilisé dans cette application</i> |
| DQ 32x24VDC/0.5A HF | 6ES7 522-1BL01-0AB0 | <i>Module pas utilisé dans cette application</i> |
| AI 8xU/I/RTD/TC ST  | 6ES7 531-7KF00-0AB0 | <i>Module pas utilisé dans cette application</i> |
| AQ 8xU/I HS         | 6ES7 532-5HF00-0AB0 | <i>Module pas utilisé dans cette application</i> |
| TM Count 2x24V      | 6ES7 550-1AA00-0AB0 | <i>Module pas utilisé dans cette application</i> |
| CM PtP RS422/485 HF | 6ES7 541-1AB00-0AB0 |  |

### 2.1.2 Interface Homme-Machine Siemens

| Composant                        | Numéro d'article    | Remarque         |
|----------------------------------|---------------------|------------------|
| TP700 Comfort                    | 6AV2 124-0GC01-0AX0 | Version 15.0.0.0 |
| Mémoire micro SD                 | 8 GB                |                  |
| Adaptateur pour mémoire micro SD |                     |                  |

## 2.2 Appareils Studer Innotec

| Composant                               | Numéro d'article               | Remarque |
|---|--------------------------------|----------|
| Interface RS 485 <-> bus Studer Innotec | Xcom-485i                      |          |
| Onduleur-Chargeur                       | Xtender XTS 900 - 12           |          |
| Module de batterie                      | Battery Status Processor (BSP) |          |
| Centre de contrôle à distance           | Remote Control Center RCC-02   |          |

## 3 Logiciels nécessaires Siemens

| Logiciel                             | Version       |
|--------------------------------------|---------------|
| Totally Integrated Automation Portal | V15, update 4 |
| STEP 7 Professional                  | V15, update 4 |
| WinCC Advanced                       | V15, update 4 |

## 4 Paramètres de communication Modbus RTU

Les paramètres de configuration du réseau Modbus RTU ainsi que les différentes adresses utilisées par les appareils Studer Innotec sont décrits ci-dessous.

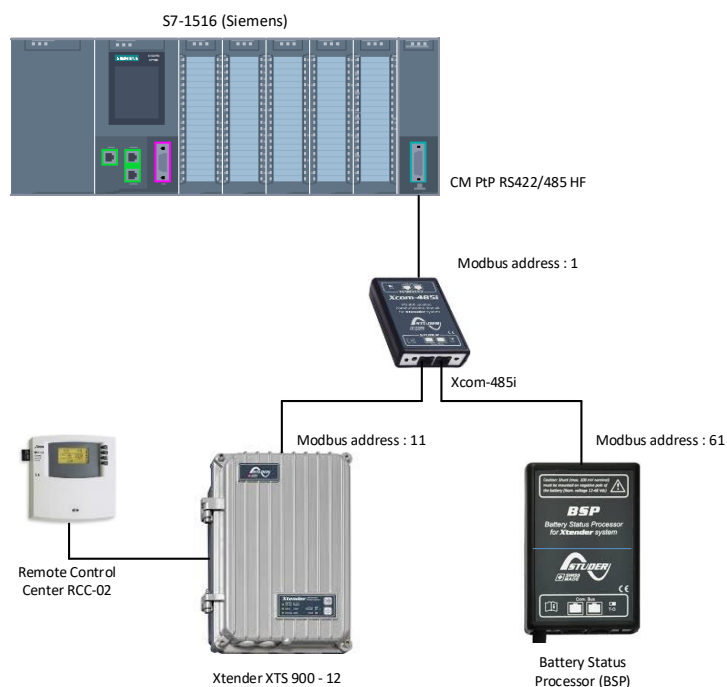


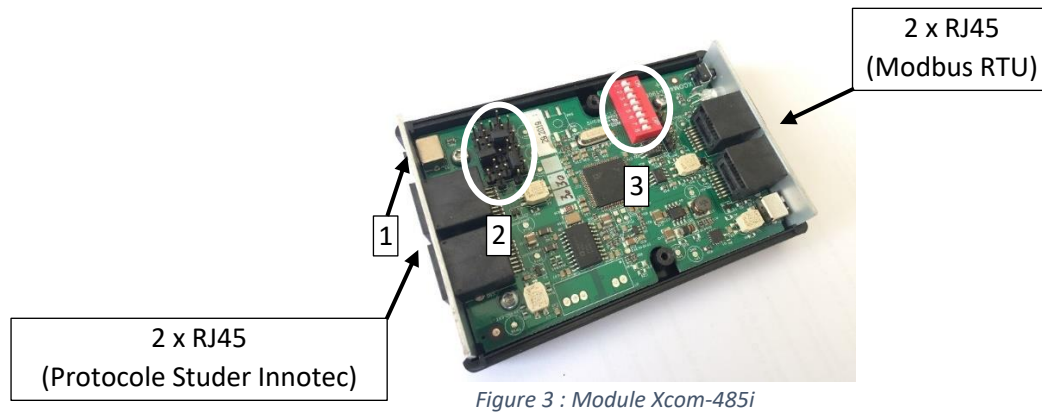
Figure 2 : Réseau Modbus RTU

### Paramètres de communication :

| Paramètre                 | Valeur par défaut   |
|---------------------------|---|
| Vitesse de transmission   | 9'600 bps   |
| Parité                    | Even (paire)  |
| Configuration de la trame | 1 bit de start<br>8 bits de données<br>1 bit de parité<br>1 bit de stop |
| Interface électrique      | RS 485 avec 2 fils (semi-duplex)  |

La plupart de ces paramètres peuvent être modifiés sur l'interface Homme-Machine.

Le module Xcom-485i sert d'interface de communication entre les appareils de Studer Innotec et l'automate programmable. Ce module possède 2 connecteurs RJ45 qui communiquent avec le protocole Modbus RTU et 2 connecteurs RJ45 qui communiquent avec le protocole propriétaire de Studer Innotec.



La configuration de l'interface est la suivante :

1. Position du commutateur externe  
Le commutateur de la terminaison de ligne doit être en position O (open).
2. Positions des jumpers selon la figure ci-après.

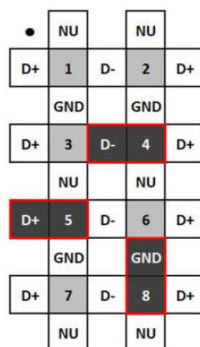


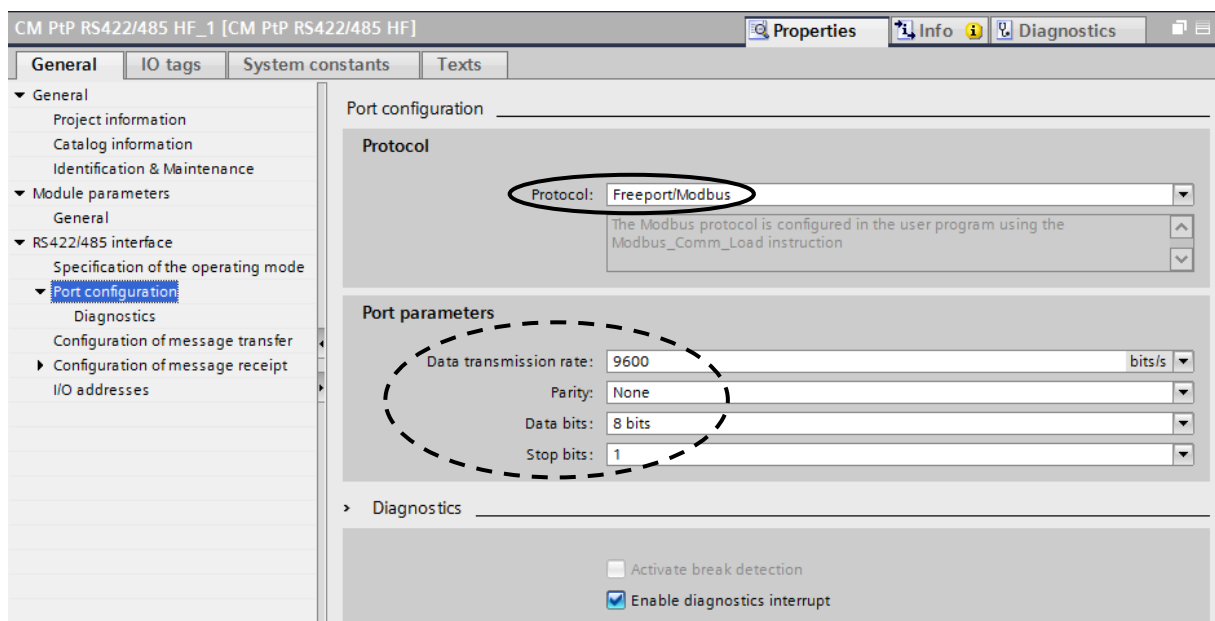
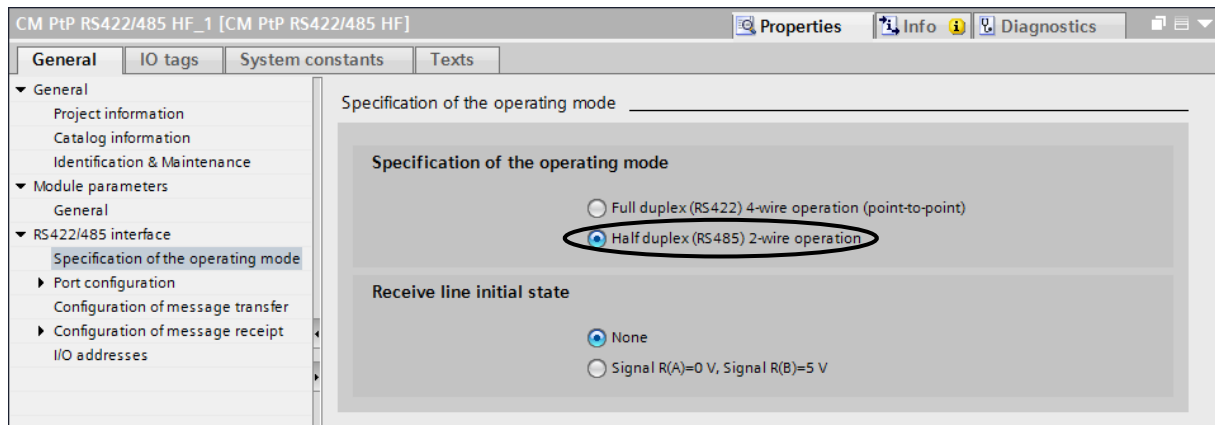
Figure 4 : Position des jumpers

3. Positions des commutateurs internes  
Tous les commutateurs doivent être sur la position Off.

## 6 Configuration du module CM PtP RS422/485 HF

Ce module permet à l'automate programmable de communiquer en qualité de maître au sein du réseau Modbus RTU.

Sa configuration est la suivante :



Les valeurs des champs sous "Port parameters" n'ont pas d'importance dans cette configuration. L'affectation de valeurs à ces paramètres est réalisée via le bloc fonctionnel " Modbus\_Comm\_Load" programmé sur l'automate programmable.

Tous les autres paramètres conservent leur valeur par défaut.

## 7 Câble de liaison entre le module CM PtP RS422/485 HF et l'interface Xcom-485i

| Module CM PtP RS422/485 HF<br>( Connecteur Sub-D 15 pôles ) |   | Interface Xcom-485i<br>( Connecteur RJ 45 ) | Description |
|---|---|---|-------------|
| Pin 4   | ↔ | Pin 4                                       | D-          |
| Pin 8   | ↔ | Pin 8                                       | Gnd         |
| Pin 11  | ↔ | Pin 5                                       | D+          |

## 8 Paramètres de communication Profinet

Le réseau Profinet est utilisé pour la communication entre l'automate programmable et l'interface Homme-Machine.

Les adresses IP nécessaires sont :

Automate programmable : 192.168.0.1  
Interface Homme-Machine : 192.168.0.2

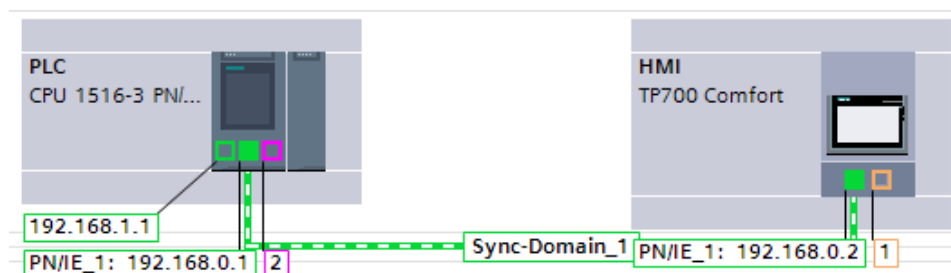


Figure 5 : Réseau Profinet



## 9 Programme de l'automate programmable

### 9.1 Blocs de programme

Les blocs de programme principaux utilisés pour lire ou écrire sur les appareils Studer Innotec sont les suivants :



Figure 6 : Blocs de programme pour lecture/écriture

### 9.2 Structure du programme

Le programme est composé de différents blocs fonctionnels qui permettent de structurer l'application. Chaque bloc fonctionnel effectue une tâche bien définie afin que l'ensemble du programme puisse effectuer des opérations de lecture et d'écriture avec les appareils Studer Innotec.

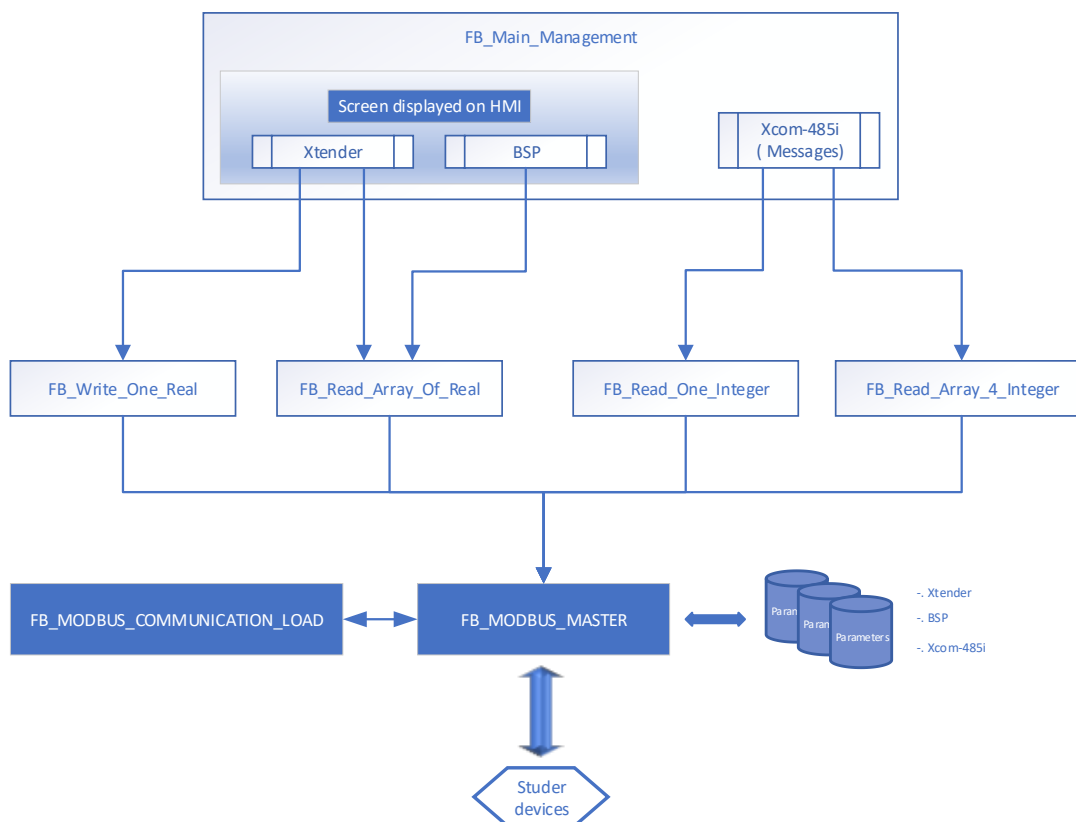


Figure 7 : Structure du programme

### 9.2.1 Gestion des opérations de lecture et d'écriture

Le bloc fonctionnel "FB\_Main\_Management" gère les opérations de lecture et d'écriture à intervalles réguliers avec les appareils Studer Innotec. La période (T) est réglée par défaut à 1 [sec.].

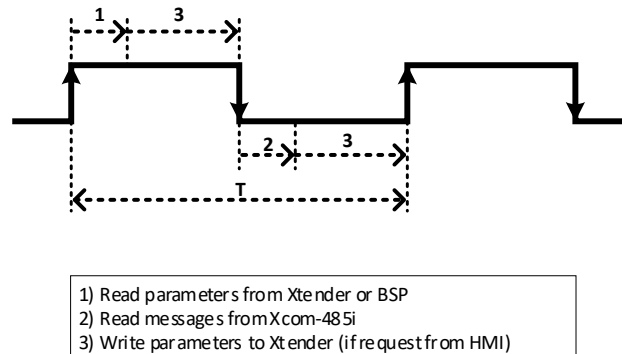


Figure 8 : Cycle de lecture et d'écriture

### 9.2.2 Séquence de lecture des paramètres depuis l'Xtender et BSP

La séquence de lecture des paramètres sur l'Xtender ou le BSP est effectuée périodiquement par le bloc fonctionnel "FB\_Read\_Array\_Of\_Real".

Seuls les paramètres de l'appareil affiché sur l'écran de l'IHM sont lus.

### 9.2.3 Séquence d'écriture des paramètres vers l'Xtender

La modification de la valeur d'un paramètre sur l'Xtender est initiée par l'opérateur via l'IHM. Le bloc fonctionnel "FB\_Write\_One\_Real" est appelé depuis le bloc fonctionnel "FB\_Main\_Management" pour effectuer l'écriture du paramètre. Ce bloc est traité si aucune opération de lecture est en cours. Si ce n'est pas le cas, l'opération d'écriture attend la fin de l'opération de lecture.

### 9.2.4 Séquence de lecture des messages depuis l'Xcom-485i

Le logiciel de l'interface Xcom-485i sauvegarde tous les messages émis par l'Xtender et le BSP.

La séquence de lecture des paramètres sur l'interface Xcom-485i est effectuée périodiquement par les blocs fonctionnels "FB\_Read\_One\_Integer" et "FB\_Read\_Array\_4\_Integer".

Le bloc fonctionnel "FB\_Read\_One\_Integer" est utilisé pour lire le nombre de messages présents sur l'interface et le bloc fonctionnel "FB\_Read\_Array\_4\_Integer" est utilisé pour lire les messages. Ces messages sont ensuite affichés sur l'IHM.

#### 9.2.4.1 Principe de fonctionnement d'une séquence

Les séquences programmées dans les blocs fonctionnels FB\_Read\_xxx et FB\_Write\_xxx sont relativement semblables. Elles se différencient principalement par les types de données à lire ou à écrire.

A titre d'exemple, le bloc fonctionnel "FB\_Read\_Array\_Of\_Real" est implémentée avec la séquence suivante.

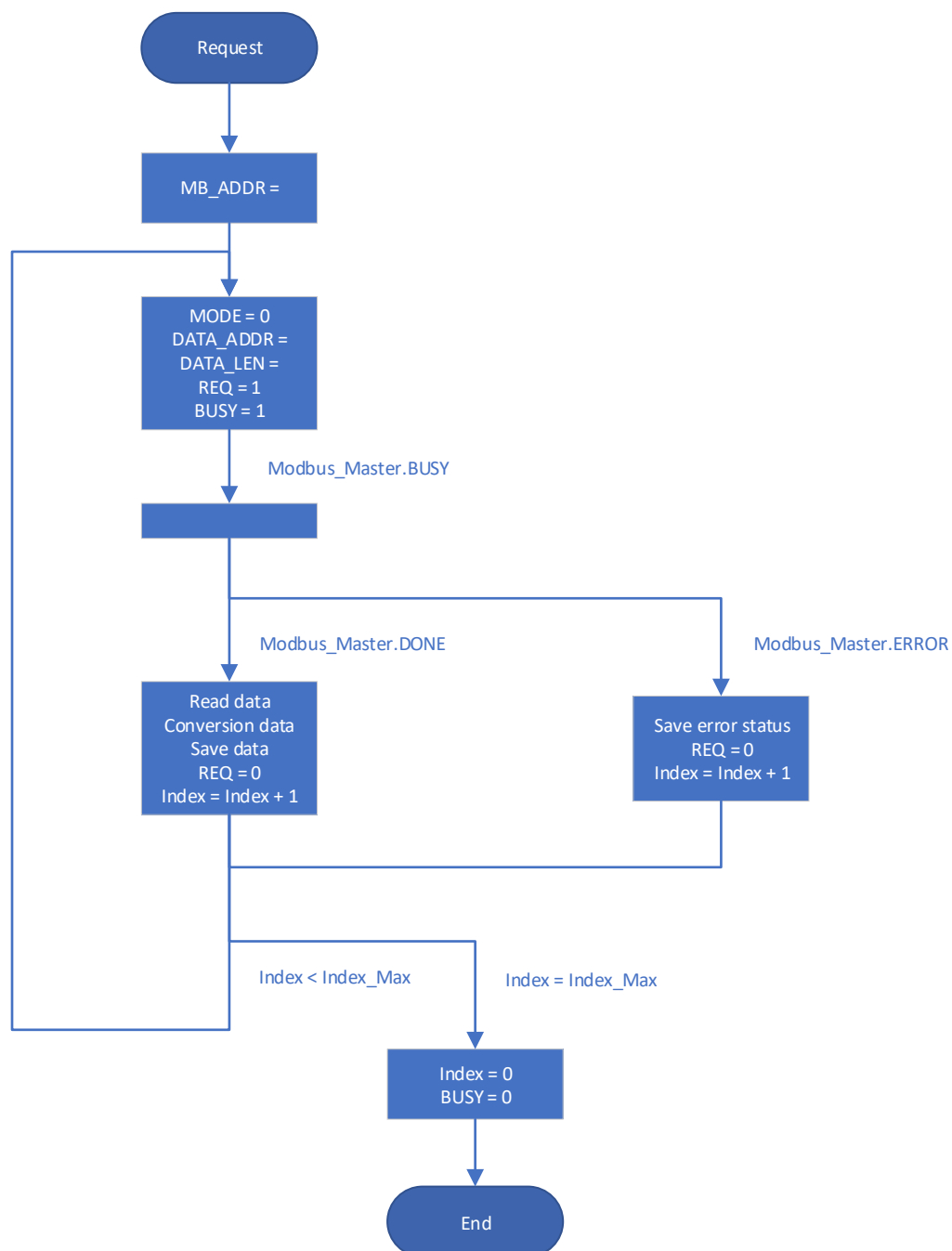


Figure 9 : Séquence du bloc fonctionnel "FB\_Read\_Array\_Of\_Real"

### 9.2.5 Fonctions de lecture et d'écriture Modbus RTU

La lecture et l'écriture des paramètres sur les appareils Studer Innotec via Modbus RTU sont effectuées avec les blocs fonctionnels "FB\_MODBUS\_COMMUNICATION\_LOAD" et "FB\_MODBUS\_MASTER".

Ces deux blocs utilisent à leur tour les blocs fonctionnels de la bibliothèque du TIA-Portal suivants :

- . Modbus\_Comm\_Load
- . Modbus\_Master

Ces blocs se trouvent dans la bibliothèque sous Communication → Processeur de communication → MODBUS (RTU).

| Communication                 |                      |
|-------------------------------|----------------------|
| Nom                           | Version              |
| ▶ Communication S7            | V1.3                 |
| ▶ Communication Open User     | <a href="#">V6.0</a> |
| ▶ OPC UA                      |                      |
| ▶ Serveur Web                 | V1.1                 |
| ▶ Autres                      |                      |
| ▼ Processeur de communication |                      |
| ▶ PtP Communication           | <a href="#">V3.1</a> |
| ▶ Communication USS           | <a href="#">V4.1</a> |
| ▼ MODBUS ( RTU )              | <a href="#">V4.2</a> |
| ▶ Modbus_Comm_Load            | V3.1                 |
| ▶ Modbus_Master               | <a href="#">V3.1</a> |
| ▶ Modbus_Slave                | <a href="#">V4.2</a> |
| ▶ Interface série ET200S      | V2.7                 |
| ▶ SIMATIC NET CP              | <a href="#">V4.1</a> |

Figure 10 : Blocs fonctionnels Modbus RTU

### 9.2.5.1 Modbus\_Comm\_Load

Le bloc fonctionnel "Modbus\_Comm\_Load" permet de sélectionner le module de communication et de définir les paramètres de communication du réseau Modbus RTU.

Il est appelé une seule fois lors du premier cycle de l'OB1 ou lorsqu'un paramètre de communication a été modifié sur l'IHM.

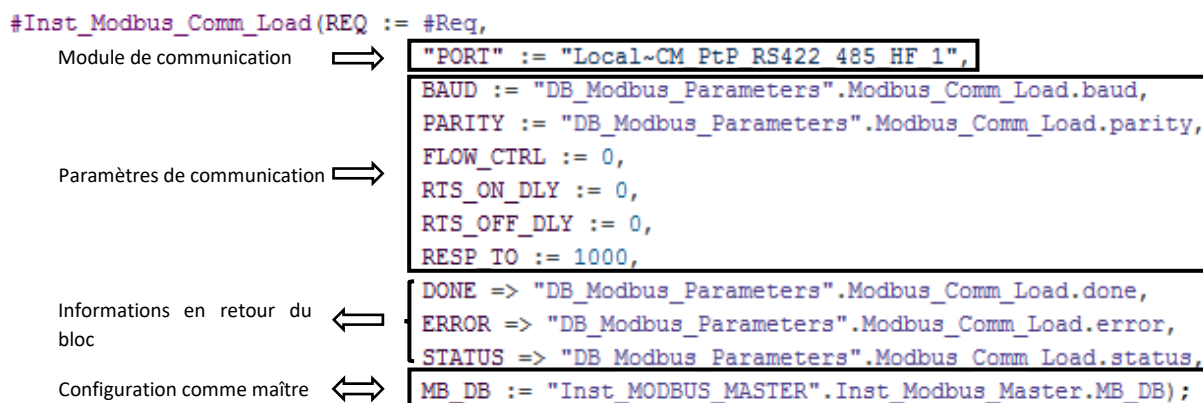


Figure 11 : Instance du bloc fonctionnel "Modbus\_Comm\_Load"

#### 9.2.5.1.1 Configuration

Après avoir inséré le module de communication CM PtP RS422/485 HF dans la configuration matérielle, on peut sélectionner le nom symbolique du module de communication au niveau du paramètre PORT. Le nom symbolique est accessible dans les constantes "système" qui sont répertoriées dans les variables de l'API.

Les paramètres de communication BAUD (vitesse de transmission) et PARITE (parité) doivent être identiques aux autres appareils du réseau (esclaves).

Le paramètre "MB\_DB" du bloc de données d'instance du bloc maître "Modbus\_Master" (cf. chap. 9.2.5.2) doit être transféré au paramètre MB\_DB du bloc fonctionnel "Modbus\_Comm\_Load". Ce paramètre définit le module de communication (paramètre PORT) comme maître Modbus.

De plus, le paramètre statique MODE dans les données d'instance de "Modbus\_Comm\_Load" doit être paramétré avec la valeur 4 qui correspond à une communication RS 485 semi-duplex en mode 2 fils.

|                       |                  |       |
|-----------------------|------------------|-------|
| Static                |                  |       |
| Inst_Modbus_Comm_Load | Modbus_Comm_Load |       |
| Input                 |                  |       |
| Output                |                  |       |
| InOut                 |                  |       |
| Static                |                  |       |
| ICHAR_GAP             | Word             | 16#0  |
| RETRIES               | Word             | 16#0  |
| MODE                  | USInt            | 4     |
| LINE_PRE              | USInt            | 16#00 |

Figure 12 : DB d'instance du bloc fonctionnel "Modbus\_Comm\_Load"

### 9.2.5.2 Modbus\_Master

Le bloc fonctionnel "Modbus\_Master" permet à l'automate programmable (maître) de lire ou d'écrire des paramètres sur les appareils Studer Innotec (esclaves) via Modbus RTU à l'aide du port configuré sur le paramètre "PORT" du bloc fonctionnel "Modbus\_Comm\_Load".

#### Lecture de paramètres :

En fonction de l'écran affiché sur l'IHM, les paramètres du module Xtender ou du module de batterie BSP sont lus en alternance avec les messages contenus dans le l'interface Xcom-485i si aucune écriture de paramètre est en cours (cf chap. 9.2.1).

#### Ecriture de paramètres :

Le bloc fonctionnel "Modbus\_Master" est appelé lorsqu'une demande de modification d'un paramètre a été déclenchée par l'opérateur sur l'IHM et si aucune lecture de paramètre est en cours (cf chap. 9.2.1).

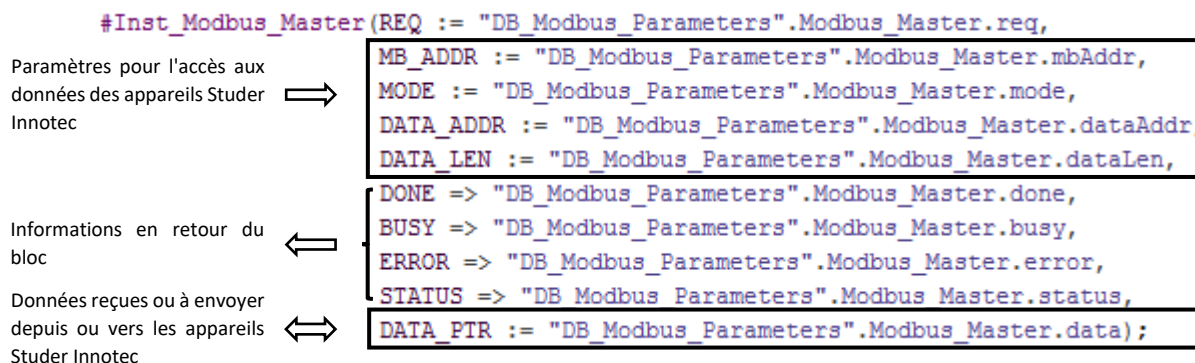


Figure 13 : Instance du bloc fonctionnel "Modbus\_Master"

#### 9.2.5.2.1 Configuration

Les paramètres du bloc fonctionnel permettent de sélectionner l'adresse Modbus de l'appareil Studer Innotec , de définir le code de fonction Modbus ainsi que la zone locale de stockage des paramètres (registres).

Le code de fonction Modbus qui est utilisé dans le télégramme Modbus est déterminé automatiquement par le bloc fonctionnel "Modbus\_Master" sur la base des paramètres MODE et DATA\_ADDR.

Le tableau ci-dessous décrit les paramètres.

| Paramètre | Description   |
|-----------|---|
| MB_ADDR   | Adresse Modbus de l'appareil Studer Innotec                                 |
| MODE      | Direction de la transmission (lecture ou écriture)                          |
| DATA_ADDR | Adresse du premier registre à accéder dans l'appareil Studer Innotec        |
| DATA_LEN  | Nombre de registres à accéder dans l'appareil Studer Innotec                |
| DATA_PTR  | Zone des données lues ou à envoyer depuis ou vers l'appareil Studer Innotec |

### 9.3 Structure des données

L'ensemble des paramètres ou des messages qui doivent être lus (ou écrits) sur les appareils Studer Innotec sont contenues dans les blocs de données suivants :

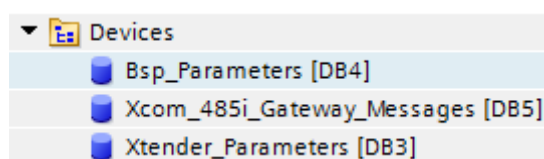


Figure 14 : Blocs de données contenant les paramètres et les messages

Chaque bloc de données contient l'adresse Modbus de l'appareil concerné ainsi qu'une structure sous forme de tableaux contenant les paramètres à lire ou à écrire.

| Xtender_Parameters |                            |   |                      |
|--------------------|----------------------------|---|----------------------|
|                    | Name                       | Data type                                   | Start value          |
| 1                  | ▼ Static                   |   |                      |
| 2                  | ■ Modbus_Address           | UInt  | 11                   |
| 3                  | ▼ Read_Parameters          | Array[0..*XTENDER_UPPER_INDEX_ARRAY_READ... |                      |
| 4                  | ■ ▼ Read_Parameters[0]     | "UDT_Read_1_Real_Register"                  |                      |
| 5                  | ■ Description              | String                                      | 'Battery voltage'    |
| 6                  | ■ Data_Address             | UDInt                                       | 30001                |
| 7                  | ■ Data                     | Real  | 0.0                  |
| 8                  | ■ Unit                     | String                                      | '[Vdc]'              |
| 9                  | ■ ▼ Read_Parameters[1]     | "UDT_Read_1_Real_Register"                  |                      |
| 10                 | ■ Description              | String                                      | 'Output voltage'     |
| 11                 | ■ Data_Address             | UDInt                                       | 30043                |
| 12                 | ■ Data                     | Real  | 0.0                  |
| 13                 | ■ Unit                     | String                                      | '[Vac]'              |
| 14                 | ■ ▶ Read_Parameters[2]     | "UDT_Read_1_Real_Register"                  |                      |
| 15                 | ■ ▶ Read_Parameters[3]     | "UDT_Read_1_Real_Register"                  |                      |
| 16                 | ■ ▶ Read_Parameters[4]     | "UDT_Read_1_Real_Register"                  |                      |
| 17                 | ■ ▶ Read_Parameters[5]     | "UDT_Read_1_Real_Register"                  |                      |
| 18                 | ▼ Parameters_To_Write      | Array[0..*XTENDER_UPPER_INDEX_ARRAY_PARA... |                      |
| 19                 | ■ ▼ Parameters_To_Write[0] | "UDT_Write_1_Real_Register"                 |                      |
| 20                 | ■ Description              | String                                      | 'Maximum current ... |
| 21                 | ■ Data_Address             | UDInt                                       | 40015                |
| 22                 | ■ Data                     | Real  | 2.0                  |
| 23                 | ■ Unit                     | String                                      | '[Aac]'              |
| 24                 | ■ Min                      | Real  | 2.0                  |
| 25                 | ■ Max                      | Real  | 50.0                 |
| 26                 | ■ ▶ Parameters_To_Write[1] | "UDT_Write_1_Real_Register"                 |                      |
| 27                 | ■ ▶ Parameters_To_Write[2] | "UDT_Write_1_Real_Register"                 |                      |
| 28                 | ■ Reading_In_Progress      | Bool  | false                |
| 29                 | ■ Writing_In_Progress      | Bool  | false                |

Figure 15 : Bloc de données de l'Xtender

Les type de données des variables contenues dans les tableaux sont définis sous forme de type de données API.

| UDT_Read_1_Real_Register |              |           |               |
|--------------------------|--------------|-----------|---------------|
|                          | Name         | Data type | Default value |
| 1                        | Description  | String    | "             |
| 2                        | Data_Address | UDInt     | 0             |
| 3                        | Data         | Real      | 0.0           |
| 4                        | Unit         | String    | "             |

Figure 16 : Type de donnée API "UDT\_Read\_1\_Real\_Register"

Le nombre de paramètres contenu dans les tableaux de données sont ajustables en modifiant la valeur des constantes utilisateur définies dans la table de variables.

| Modbus_RTU_Master > PLC [CPU 1516-3 PN/DP] > PLC tags > Table de variables standard [66] |   |           |       |                         |
|--|---|-----------|-------|-------------------------|
|  |   |           | Tags  | User constants          |
| Table de variables standard  |   |           |       |                         |
|  | Name  | Data type | Value | Comment                 |
| 1  | XCOM_485_UPPER_INDEX_ARRAY_MESSAGES_ARCHIVE   | UInt      | 126   | Number of archives -1   |
| 2  | XTENDER_UPPER_INDEX_ARRAY_READ_PARAMETERS     | UInt      | 5     | Number of parameters -1 |
| 3  | XTENDER_UPPER_INDEX_ARRAY_PARAMETERS_TO_WRITE | UInt      | 2     | Number of parameters -1 |
| 4  | BSP_UPPER_INDEX_ARRAY_READ_PARAMETERS         | UInt      | 4     | Number of parameters -1 |

Figure 17 : Constantes définissant le nombre de paramètres contenu dans les tableaux



## 10 Programme de l'Interface Homme-Machine (IHM)

L'Interface Homme-Machine (IHM) permet de :

- Visualiser certains paramètres de l'Xtender et du BSP
- Modifier certains paramètres de l'Xtender
- Visualiser les messages contenus dans l'interface Xcom-485i
- Modifier certains paramètres de communication Modbus RTU
- Visualiser un message d'information lors de chaque transfert de données
- Connaître la version du programme

Elle est composée des 6 écrans suivants :

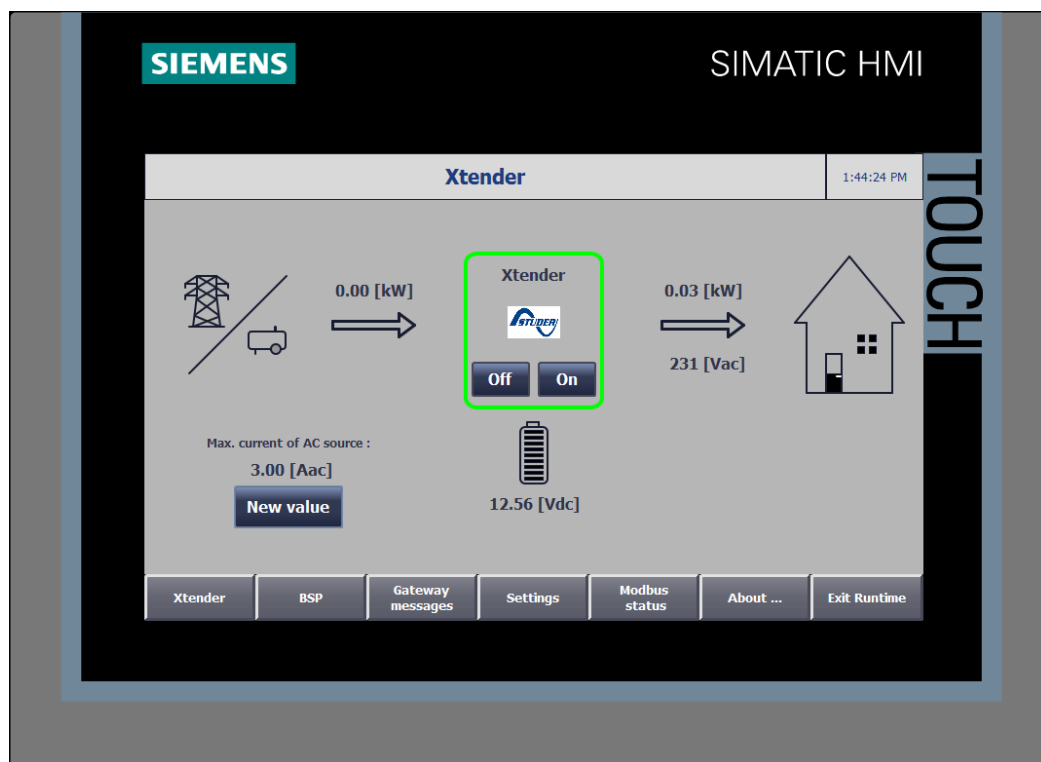


Figure 18 : Ecran "Xtender"

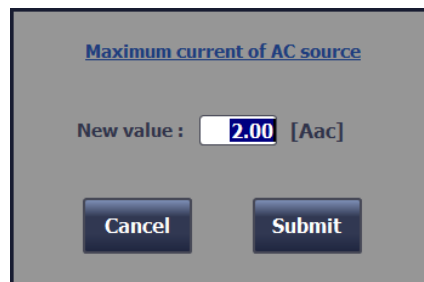


Figure 19 : Modification de la valeur maximum du courant de la source AC sur l'Xtender

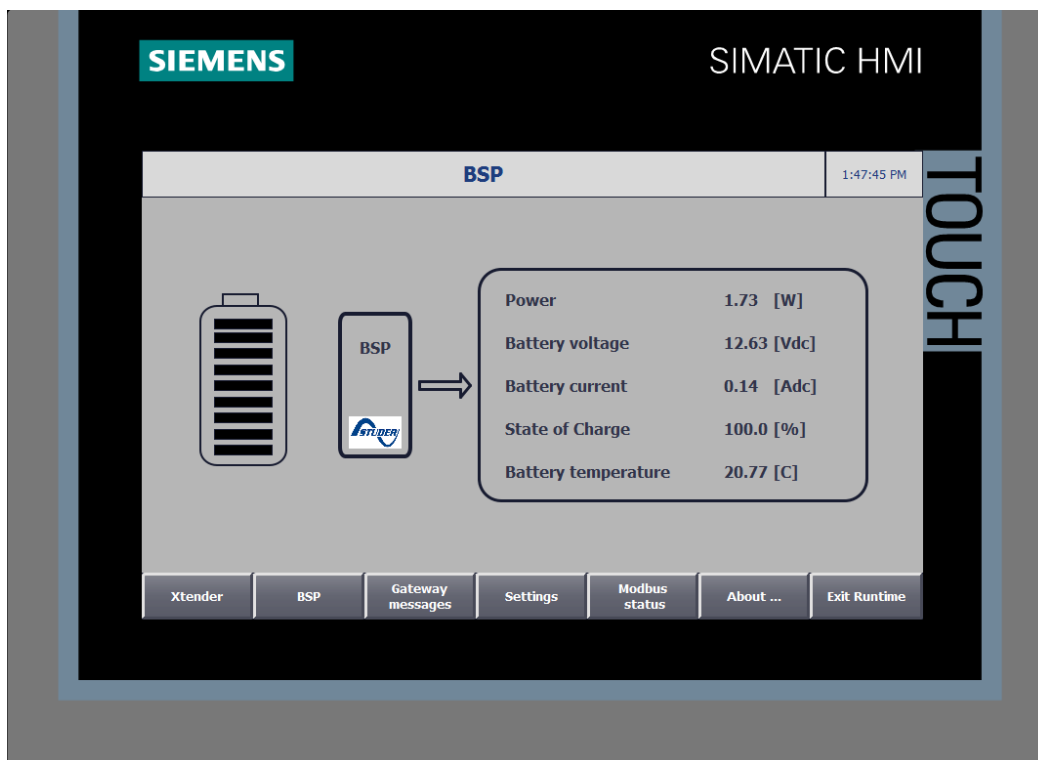


Figure 20 : Ecran "BSP"

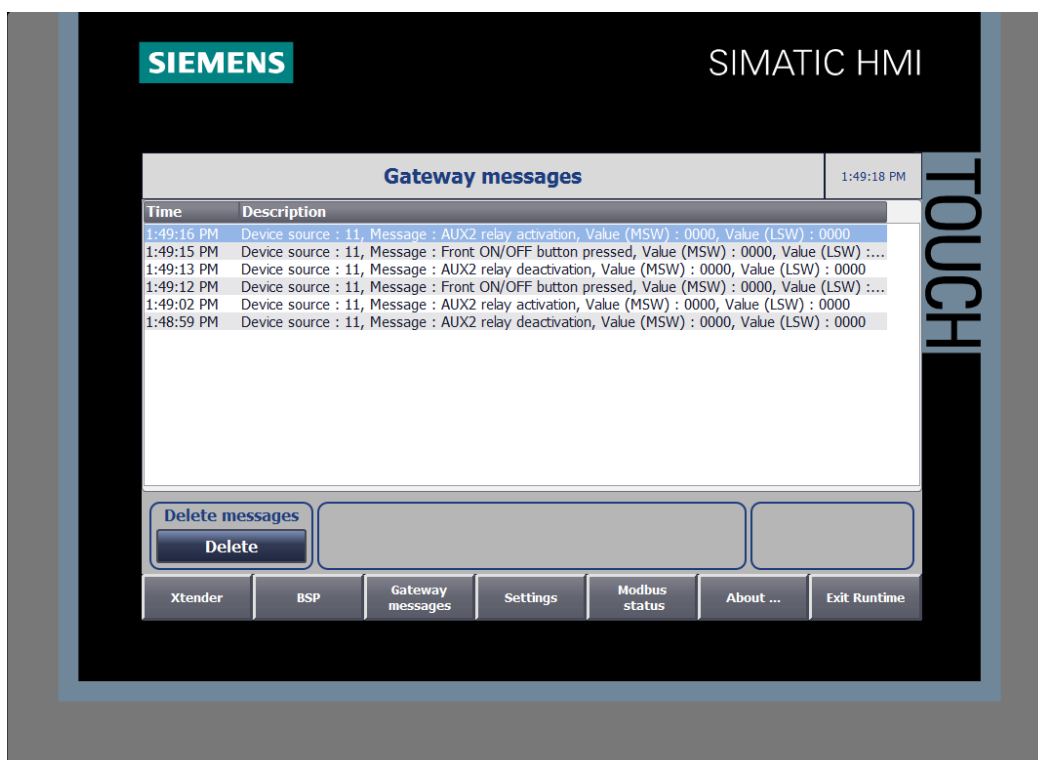


Figure 21 : Ecran "Gateway Messages"

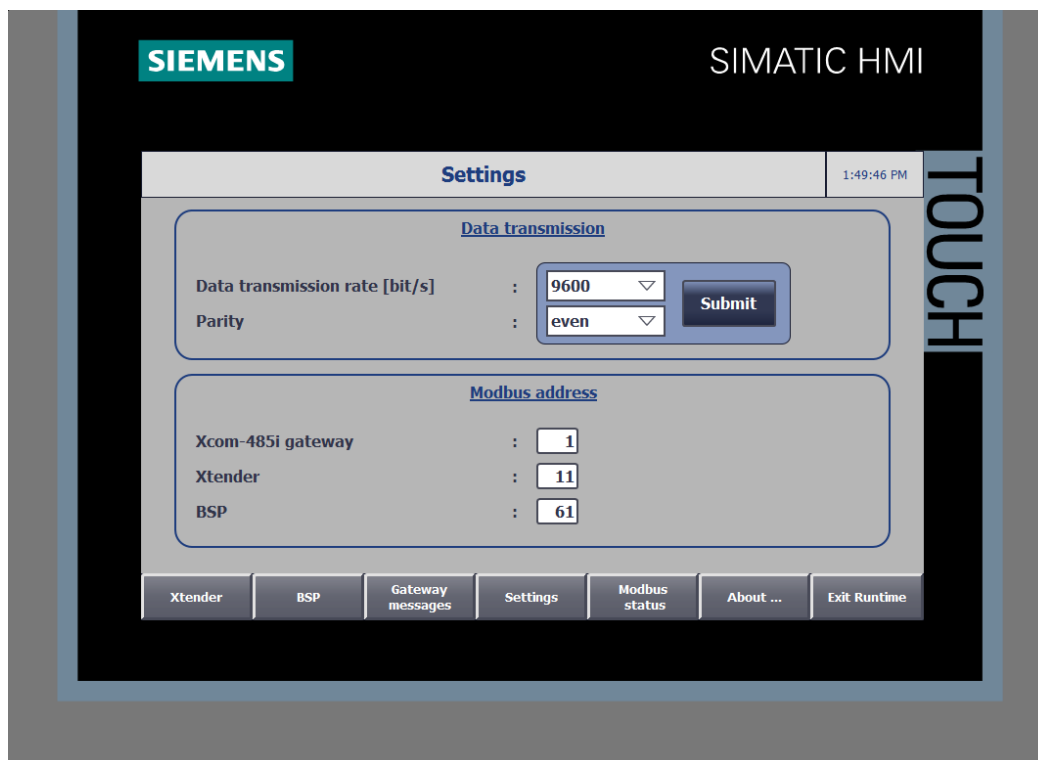


Figure 22 : Ecran "Settings"

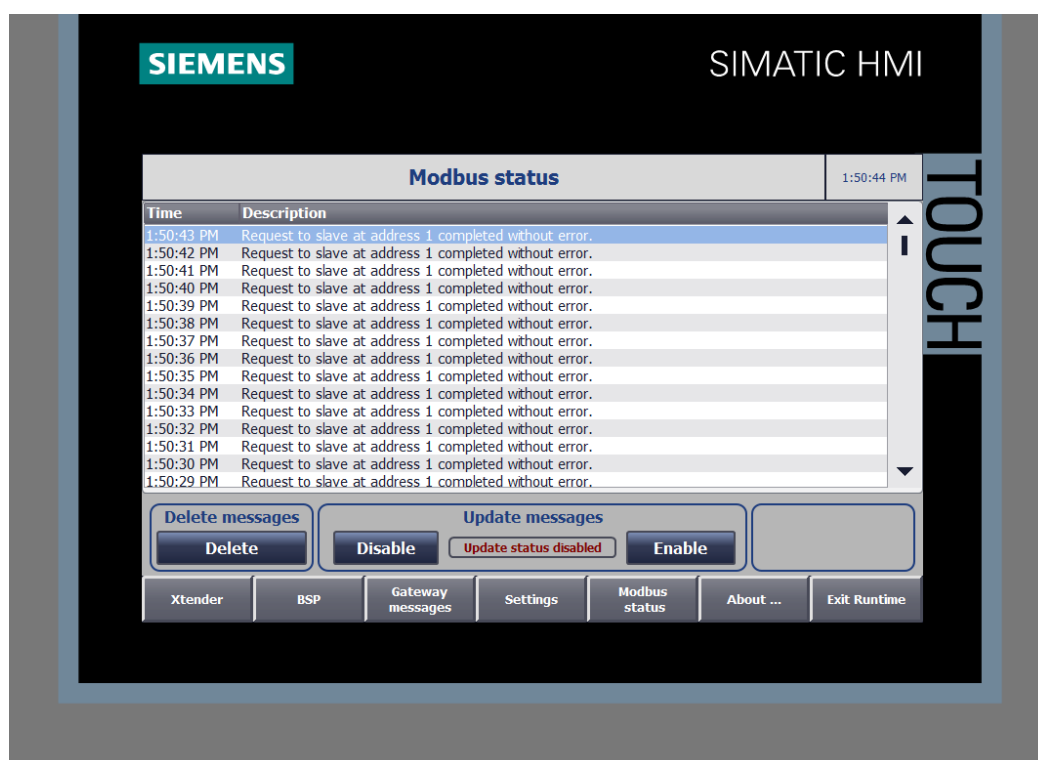


Figure 23 : Ecran "Modbus Status"

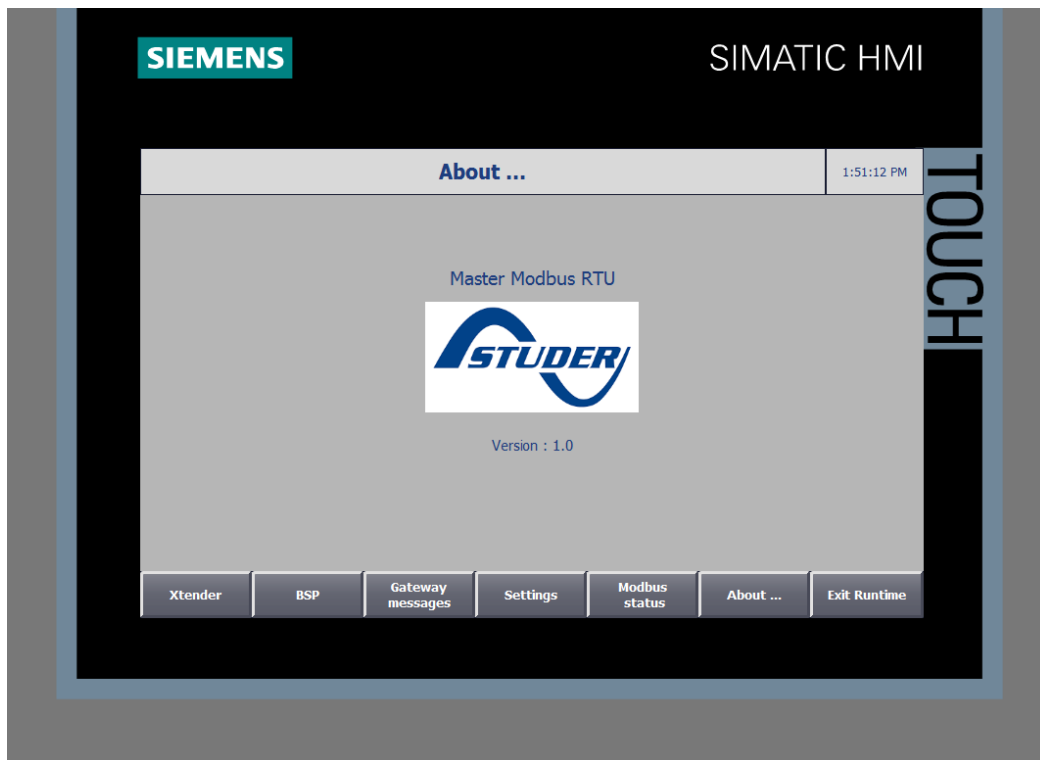


Figure 24 : Ecran "About"

Remarque :

Les messages qui sont lus dans l'interface Xcom-485i ainsi que les messages d'information générés lors des transferts de données sont sauvegardés dans une mémoire flash insérée sur l'IHM.