

sorties relais. Il est supprimé du serveur par la passerelle après récupération en local. Après exécutions des commandes un fichier d'acquittement est envoyé au serveur (prefixID_ACK_AAMMJ_hhmmss.csv).

Fichier de commandes : prefixID_CMD.csv.

Les paramètres des commandes diffèrent selon le type de commande envoyée comme indiqué ci-dessous :

index;IO;indexIO;action

Avec :

index	1 à N : Identifiant unique permettant d'identifier la commande
indexIO	1 à N : Index de l'élément à commander. Cet index correspond au premier champ de l'élément décrit dans le fichier de définition IO.
action	0 : Ouverture du contact 1 : Fermeture du contact 2 : Impulsion (1s)

Sur réception d'une commande des sorties TOR la passerelle force une acquisition de ses entrées TOR, analogique et ses sorties TOR. Ses données seront donc disponibles dans le prochain fichier de données IO déposé sur le serveur.

Fichier d'acquittement : prefixID_ACK_AAMMJ_hhmmss.csv.

Le fichier d'acquittement reprend le fichier de commande avec l'ajout d'un horodatage et de l'acquittement :

Date-time;index;IO;indexIO;action;;ack

Avec ack=OK ou ERROR.

10 Gestion des équipements Modbus

Le présent chapitre décrit l'ensemble des éléments permettant la gestion d'équipement Modbus par la WebdynSun. Elle peut gérer jusqu'à 247 esclaves Modbus, mais cette limite dépend du type d'esclave présent sur le bus et du nombre de variables à collecter par équipement.

10.1 Câblage du bus

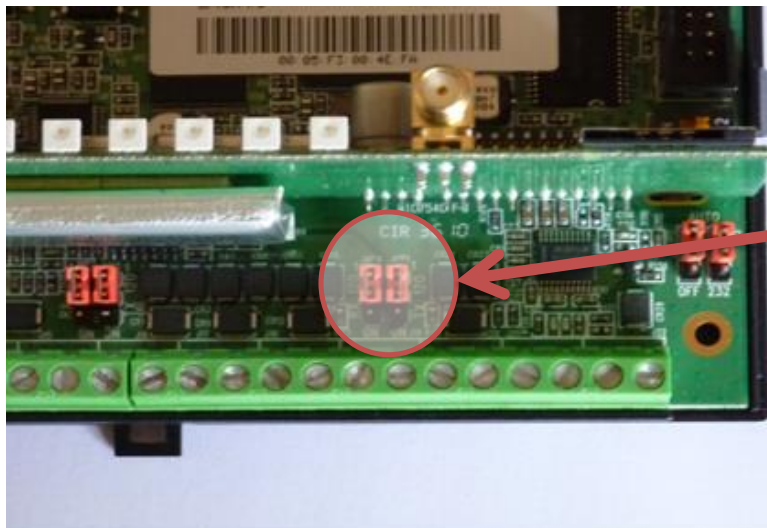
La communication avec les équipements Modbus peut être RS485 (2 ou 4 fils), RS232 ou Ethernet. Ce choix se fait par configuration logicielle via le fichier de configuration *prefixID_daq.ini*.

Dans le cas d'une configuration RS485 la passerelle peut se trouver à l'extrémité du bus de communication RS485 ou en milieu de bus. Afin d'assurer le fonctionnement du bus de données RS485, ce dernier doit être terminé aux deux extrémités par un bouchon 120 Ohms.

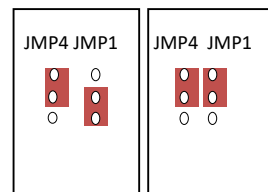
Suivant le positionnement de la passerelle sur le bus, ce bouchon doit être activé ou désactivé via un jeu de deux cavaliers (JMP4 et JMP1) présents à l'intérieur du boîtier.

**Configuration des cavaliers de
terminaison de bus**

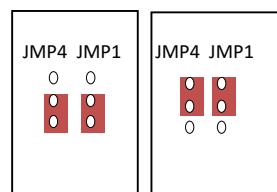
JMP4 et JMP1



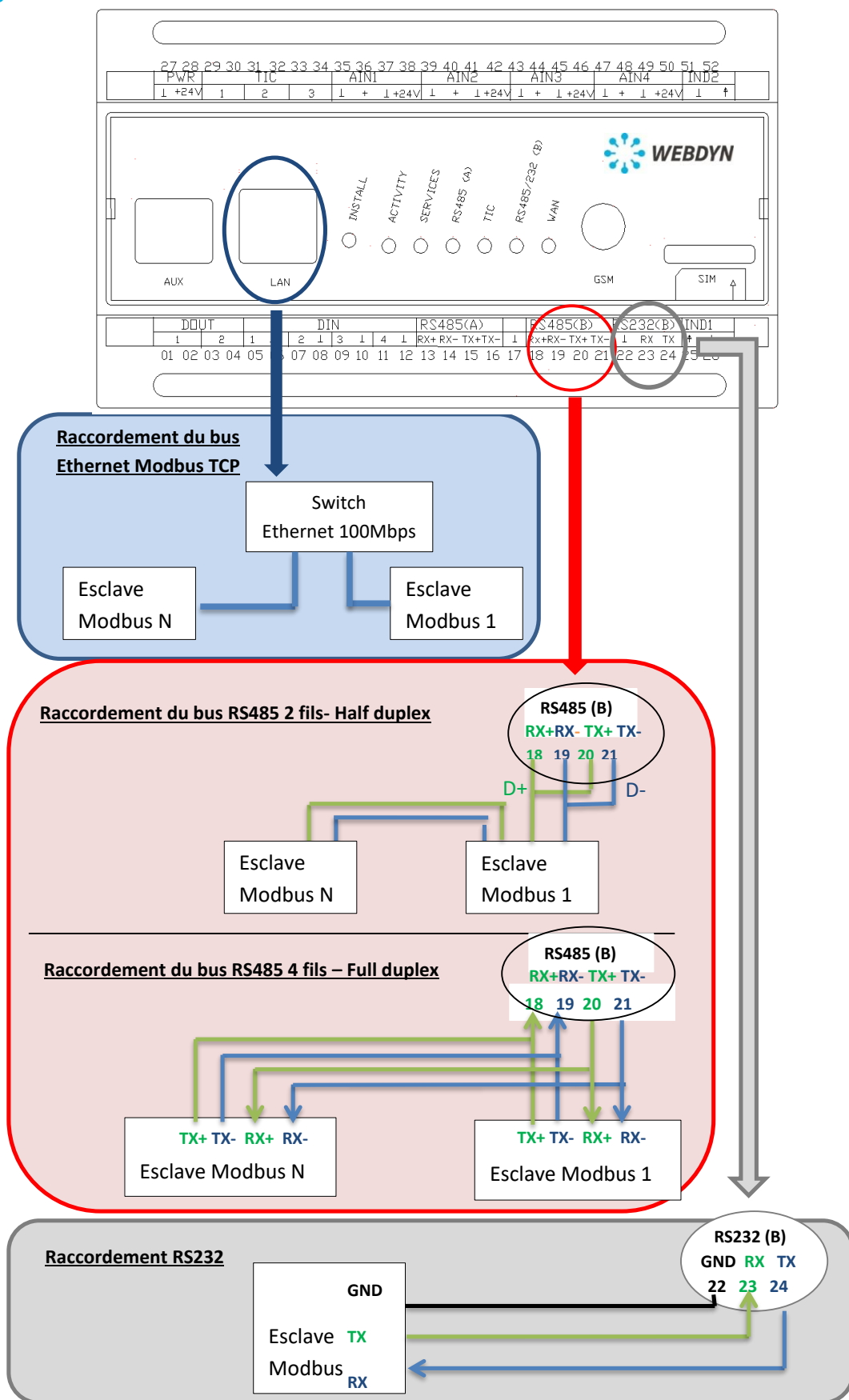
RS485 2 fils – Half Duplex



RS485 4 fils – Full Duplex



Consultez la documentation des équipements Modbus concernant leur principe de raccordement et de câblage.



10.2 Configuration et déclaration des esclaves Modbus

La passerelle WebdynSun supporte aussi bien les équipements communiquant via Modbus RTU que ceux communiquant via Modbus TCP. Des équipements RTU et TCP peuvent communiquer simultanément avec une même passerelle WebdynSun.

L'ensemble des équipements Modbus RTU à raccorder à la WebdynSun doit utiliser la même interface, le choix RS232, RS485 2 fils ou RS485 4 fils est exclusif. La liaison série associée doit être configurée à l'identique pour tous les équipements Modbus RTU. Seule l'adresse esclave doit être unique pour chaque équipement, permettant ainsi leur identification sur le bus.

Contrairement aux onduleurs et aux compteurs TIC il n'y a pas de découverte des équipements Modbus possible. Il faut donc déclarer dans le fichier *prefixID_daq.ini* chaque équipement présent sur le bus. Cette configuration s'effectue via le serveur FTP.

Le fichier *prefixID_daq.ini* doit être renseigné avec les paramètres listés ci-après :

Paramètres communs à toutes les interfaces :

Variable	Définition	Commentaire	Valeur par défaut
DAQ_Period	Période de relevé en minutes commune à toutes les acquisitions (Onduleurs, TIC, E/S, Modbus)	Valeur possible de 1 à 59 minutes	10
DAQ_PeriodSec	Période de relevé en secondes commune à toutes les acquisitions (Onduleurs, TIC, E/S, Modbus) Pris en compte seulement si DAQ_Period est égale à 0.	Valeur possible de 0 à 59 secondes	0
DAQ_HeaderOption	Activer/Désactiver l'affichage des entêtes de colonne dans les fichiers de donnée 0=désactivé 1=activé		0

Paramètres génériques caractérisant le bus :

Variable	Définition	Commentaire	Valeur par défaut
MODBUS_Mode	Mode de communication série Modbus : 0 : RS232 1 : RS485 2 fils 2 : RS485 4 fils	Bus RS232 ou RS485 en 2 ou 4 fils	1
MODBUS_BaudRate	Vitesse de la liaison série: 1200 2400 9600 19200 38400 57600 115200		9600
MODBUS_Parity	Parité de la liaison série: 0 : Aucune 1 : Paire 2 : Impaire		0
MODBUS_DataBit	Nombre de bits de données : 8 7		7
MODBUS_StopBit	Nombre de bits de stop : 1 2		1

Paramètres spécifiques par esclave Modbus:

Variable	Définition	Commentaire	Valeur par défaut
MODBUS_Addr[n]	Adresse de l'équipement Modbus <i>n</i> entre 1 et 254	n=0 à 246	1
MODBUS_Name[n]	Nom de l'équipement Modbus <i>n</i> (29 caractères max)	n=0 à 246	vide
MODBUS_FileDefName[n]	Nom du fichier de définition de l'équipement Modbus <i>n</i> (59 caractères max)	n=0 à 246	vide
MODBUS_Type[n]	Type d'équipement : générique ou spécifique si différent de 0	n=0 à 246 Utilisé seulement dans des cas spécifiques comme les équipements MERSEN.	0
MODBUS_Interface[n]	Média utilisé : 0 : Interface série (RS485/RS232) 1 : Ethernet	n=0 à 246	0
MODBUS_IpAddr[n]	Adresse IP de l'équipement dans le cas d'une communication en Modbus TCP.	n=0 à 246	vide

Exemple :

- *Configuration du bus :*

Mode RS485 2 fils, Baudrate: 19200 bauds, 8 bits de données, 1 bit de stop, pas de parité.

```
MODBUS_Mode=1
MODBUS_BaudRate=19200
MODBUS_DataBit=8
MODBUS_StopBit=1
MODBUS_Parity=1
```

- *Déclaration des esclaves Modbus:*

Esclave 0 : Boîte de jonction GreenString de MERSEN : Modbus spécifique MERSEN (Type 1), adressé en 1, nommé « SLAVE1 » et défini par le fichier de définition MODBUS_GREENSTRING.ini.

Esclave 1 : Equipement Modbus standard : Modbus générique (Type 0), adressé en 2, nommé SLAVE2» et défini par le fichier de définition MODBUS_SLAVE.ini.

```
MODBUS_Type[0]=1
MODBUS_Addr[0]=1
MODBUS_Name[0]= SLAVE1
MODBUS_FileDefName[0]=MODBUS_GREENSTRING.ini
MODBUS_Type[1]=0
MODBUS_Addr[1]=2
MODBUS_Name[1]=SLAVE2
MODBUS_FileDefName[1]=MODBUS_SLAVE.ini
```

10.3 Structure d'un fichier de définitions Modbus

Chaque esclave Modbus déclaré dans le fichier de configuration *prefixID_daq.ini* doit avoir un fichier de définition.

Ce fichier a pour but de décrire l'ensemble des variables disponibles pour un équipement Modbus.

Il décrit pour chaque variable :

- La méthode de récupération: utilisé par la passerelle pour récupérer la donnée.
- La méthode de traitement: moyenne, instantanée, paramètre ou alarme.
- La mise en forme : nom, unité et coefficient de mise à l'échelle.

Ce fichier doit être mis à disposition pour la passerelle sur le serveur FTP.

Par convention le nom du fichier a le format suivant:

prefixID_MODBUS_deviceName.ini, avec « *prefixID* » correspondant à l'identifiant de la passerelle et « *deviceName* » correspondant au nom de l'équipement.

Un seul fichier peut être utilisé pour plusieurs équipements dont le comportement est rigoureusement identique.

Le fichier comporte deux tables de déclaration :

- la table de déclaration des requêtes Modbus nommée « Modbus_RequestsTables ».
Chaque élément de cette table est défini par 10 champs séparés par un point-virgule.

Champ	Description
<i>index</i>	Index de la requête de 1 à N
<i>name</i>	Désignation de la requête Modbus
<i>readFunctionCode</i>	Code fonction et sous code fonction en lecture 0 à 65535 Codes standards gérés: 1 => "read coils" 2 => "read discrete inputs" 3 => "read holding registers" 4 => "read input registers"
<i>writeFunctionCode</i>	Code fonction et sous code fonction en écriture 0 à 65535 Codes standards gérés: 5 => "write single coils" 6 => "write single register" 15 => "write multiple coils" 16 => "write multiple registers"
<i>startRegister</i>	Adresse du premier registre Modbus 0 à 65535 <i>Les adresses sont en mots (16b) bit, octet, mots -> +1 sur adr double mot, float -> + 2</i>
<i>size</i>	Taille de la réponse sans en-tête et CRC (en octet) 1 à 250 <i>Les adresses sont en mots (16b) bit, octet, mots -> +1 sur adr double mot, float -> + 2</i>
<i>enableReading</i>	Activation de la requête en lecture : 0 => activé en mode instantanée (paramètres) 1 => activé en mode polling (mesures) 2 => désactivé
<i>enableWriting</i>	Activation de la requête en écriture 0 => activé en écriture avec une relecture avant écriture 1 => activé en écriture simple 2 => désactivé
<i>option1</i>	Réservé
<i>option2</i>	Réservé

- la table de déclaration des variables nommée « Modbus_VariablesTables ».
- Chaque élément de cette table est défini par 12 champs séparés par un point-virgule.

Champ	Description
<i>index</i>	Index de la variable de 1 à N
<i>indexRequest</i>	Désignation de la requête Modbus définie dans la table Modbus_RequestsTables.
<i>name</i>	Désignation de la variable Modbus
<i>type</i>	Type de variable 1 => bit 2 => octet 3 => mot 4 => mot inversé 5 => double mot 6 => double mot inversé 7 => flottant 8 => flottant inversé 9 => chaîne de caractères 10 => Format spécifique afficheur Siebert
<i>signed</i>	Signe de la variable 1 => signé 2 => non signé
<i>position</i>	Position de la variable dans la trame 1 à N
<i>option1</i>	Reserve
<i>option2</i>	Réserve
<i>coeffA</i>	Coefficient A de mise à l'échelle de la variable (Ax +B)
<i>coeffB</i>	Coefficient B de mise à l'échelle de la variable (Ax +B)
<i>unit</i>	Unité de la variable
<i>action</i>	Méthode de traitement de la variable : 0 : variable non relevé. 1 : variable traitée comme paramètres 2 : récupération du minimum, maximum et calcul de la moyenne. 4 : valeur instantanée 8 : déclencheur d'alarme sur changement d'état

Les adresses sont en mots (16b)
bit, octet, mots -> +1 sur adr
double mot, float -> + 2

Exemple :

Définition d'un esclave Modbus gérant 4 compteurs d'impulsions, 4 sorties et 4 entrées.

*tout est en
mot -> 16b*

Déclaration des requêtes Modbus:

	Requête 1	Requête 2	Requête 3
index	1	2	3
name	counter	output	input
readFunctionCode	3 (read holding registers)	1 (read coils)	3 (read holding registers)
writeFunctionCode	0	5 : write single coil	0
startRegister	0 <i>(p_u16*) counter = 0</i>	4 <i>(p_u16*) output = 4</i>	8 <i>(p_u16*) input = 8</i>
size	4 <i>counter[4] 4 x 16b</i>	4 <i>output[4] 4 x 16b</i>	4 <i>input[4] 4 x 16b</i>
enableReading	1 (activé)	1 (activé)	1 (activé)
enableWriting	2 (désactivé)	1 (activé)	2 (désactivé)
option1	Non renseigné	Non renseigné	Non renseigné
option2	Non renseigné	Non renseigné	Non renseigné

```
Modbus_RequestsTables={
1;counter;3;0;0;4;1;2
2;output;1;5;4;4;1;1
3; input;2;0;8;4;1;2
}
```

Déclaration des variables :

	Variable 1	Variable 2	...	Variable 12
index	1	2		12
indexRequest	1 (requête « counter »)	1 (requête « counter »)		3 (requête « input »)
name	counter1	counter2		input4
type	3 (mot) <i>u16</i>	3 (mot) <i>u16</i>		1 (bit) <i>1 bit dans u16</i>
signed	2 (non signé)	2 (non signé)		2 (non signé)
position <i>!!! commence à 1 !</i>	1 (1er élément de la trame). 1 = le 1er !!!	2 (2eme élément de la trame).		4 (4eme élément de la trame)
option1	Non renseigné	Non renseigné		Non renseigné
option2	Non renseigné	Non renseigné		Non renseigné
coeffA	1	1		1
coeffB	0	0		0
unit	pulse	pulse		Non renseigné
action	4 (valeur instantanée)	4 (valeur instantanée)		8 (alarme)

position : 1 à n



 index des
variables
1 à v

```

Modbus_VariablesTables={
1;1;counter1;3;2;1;;1;0;pulse;4      compteur 16b (u16)
2;1;counter2;3;2;2;;1;0;pulse;4
3;1;counter3;3;2;3;;1;0;pulse;4
4;1;counter4;3;2;4;;1;0;pulse;4
5;2;output1;1;2;1;;1;0;;4             output 1 bit dans u16
6;2;output2;1;2;2;;1;0;;4
7;2;output3;1;2;3;;1;0;;4
8;2;output4;1;2;4;;1;0;;4
9;3;input1;1;2;1;;1;0;;8              input 1 bit dans u16
10;3;input2;1;2;2;;1;0;;8
11;3;input3;1;2;3;;1;0;;8
12;3;input4;1;2;4;;1;0;;8
}
  
```

10.4 Vérification du bon fonctionnement des équipements Modbus

Il est conseillé de vérifier le bon fonctionnement des équipements Modbus après leur installation et leur configuration. Ceci s'effectue par le biais de l'interface Web embarqué en accédant au menu « Supervision/Modbus » :

Supervision des équipements Modbus			
Etat	Nom	Adresse	Fichier de définition
1 	Slave 1	1	prefixID_MODBUS_TYPE1.ini
2 	Slave 2	2	prefixID_MODBUS_TYPE2.ini

Actualiser

Etat :

Indique l'état de l'équipement Modbus configuré.



L'esclave Modbus est correctement configuré et communique avec la WebdynSun.



L'esclave Modbus n'est pas correctement configuré ou ne communique pas avec la WebdynSun.

Fichier de définition :

Indique l'état du fichier de définition associé à l'équipement Modbus configuré.

prefixID_File.ini : fichier récupéré localement et conforme.

prefixID_File.ini : fichier non récupérer localement ou non conforme.

Vous pouvez également regarder la LED RS485/232(B) en face avant du produit pour vérifier l'activité sur le bus Modbus. Cette LED clignote rapidement sur réception de trame Modbus.

10.5 Les données Modbus

Une fois paramétrée, la WebdynSun collecte en permanence les données des équipements Modbus puis les enregistre dans un fichier texte au format CSV. Ce fichier est compressé au format GZ puis déposé périodiquement sur le serveur FTP pour pouvoir être exploité.

10.5.1 Syntaxe du nom de fichier

Le fichier de données déposé sur le serveur FTP respecte le format suivant :

```
prefixID_MODBUS_AAMMJ_hhmmss.csv.gz
```

Avec :

prefixID : identifiant de la passerelle.

AAMMJ_hhmmss : horodatage de l'archive au format « année-mois-jour-heure-minute-seconde ».

10.5.2 Format des données

Le format du fichier est le suivant : (en vert les données optionnelles activables/désactivables dans *IDSite_daq.ini*).

```
ADDRMODBUS;slaveAddr_1;NumDevice1
TypeMODBUS;fileDefinitionName_1
nbVariableDevice1;indexVariable_1_Device1;indexVariable_2_Device1;indexVariable_x_Device1
date-time_1;variable_1_value_1_Device1;variable_2_value_1_Device1;variable_x_value_1_Device1
date-time_2;variable_1_value_2_Device1;variable_2_value_2_Device1;variable_x_value_2_Device1
date-time_n;variable_1_value_n_Device1;variable_2_value_n_Device1;variable_x_value_n_Device1
ADDRMODBUS;slaveAddr_N;NumDeviceN
TypeMODBUS;fileDefinitionName_N
nbVariableDeviceN;indexVariable_1_DeviceN;indexVariable_2_DeviceN;indexVariable_x_DeviceN
date-time_1;variable_1_value_1_DeviceN;variable_2_value_1_DeviceN;variable_x_value_1_DeviceN
date-time_2;variable_1_value_2_DeviceN;variable_2_value_2_DeviceN;variable_x_value_2_DeviceN
date-time_n;variable_1_value_n_DeviceN;variable_2_value_n_DeviceN;variable_x_value_n_DeviceN
```

Avec:

slaveAddr_N : adresse de l'esclave Modbus (1 à 247).

NumDeviceN : numéro de l'équipement au format DDD (001 à 247)

fileDefinitionName_N : nom du fichier de définition associé à l'équipement.

nbVariableDeviceN : nombre de variables collecté par équipement

indexVariable_x_DeviceN : index de la variable collectée

date-time_n : horodatage de l'acquisition des données au format AA/MM/JJ-hh:mm:ss

variable_x_value_n : valeur n relevée de la variable x à date-time n.

Le fichier de définition étant:

```
# Définition des trames
#
Id;Name;ReadFctCode;WriteFctCode;StartReg;NbReg;EnableReading;EnableWriting;Option1;Option2
Modbus_RequestsTables={
1;request;3;0;0;4;1;2
}
# Définition des variables
# Id;ReqId;Name;Type;Signed;Position;Option1;Option2;CoeffA;CoeffB;Unit;Action;
Modbus_VariablesTables={
indexVariable_1;1;data1;3;2;1;;;1;0;unit;4
indexVariable_2;1;data2;3;2;2;;;1;0;unit;4
indexVariable_x;1;data_x;3;2;3;;;1;0;unit;4
}
```

Cas particulier d'une donnée moyennée :

Si une donnée est configurée en Min/Max/Moyenne dans le fichier de définition elle sera présentée dans le fichier de données de la façon suivant :

```
nbVariableDeviceN;indexVariable_1(min);indexVariable_1(max);indexVariable_1(moy);
date-time_n;variable_x_value_n_min;variable_x_value_n_max;variable_x_value_n_moy
```

10.5.3 Exemple

Exemple : 2 équipements Modbus, acquisition toutes les 15 minutes.

```
ADDRMODBUS;1;001
TypeMODBUS;prefixID_MODBUS_TYPE1.ini
12;1;2;3;4;5;6;7;8;9;10;11;12
27/03/13-09:45:00;32;52;5;102;1;0;1;0;0;0;0;0
27/03/13-10:00:00;35;57;5;108;1;1;0;0;0;0;0;1
ADDRMODBUS;2;002
TypeMODBUS;prefixID_MODBUS_TYPE2.ini
6;1 (min);1 (max);1 (moy);2 (min);2 (max);2 (moy)
27/03/13-09:45:00;16;32;26.00;52;58;54.00
27/03/13-10:00:00;4;6;5.50;102;105;103.00
```

Avec:

prefixID_MODBUS_TYPE1.ini

```
# Définition des trames
#
Id;Name;ReadFctCode;WriteFctCode;StartReg;NbReg;EnableReading;EnableWriting;Option1;Option
2
Modbus_RequestsTables={
1;counter;3;0;0;4;1;2
2;output;1;5;4;4;1;1
3;input;2;0;8;4;1;2
}
# Définition des variables
# Id;ReqId;Name;Type;Signed;Position;Option1;Option2;CoeffA;CoeffB;Unit;Action;
Modbus_VariablesTables={
1;1;counter1;3;2;1;;;1;0;pulse;4
2;1;counter2;3;2;2;;;1;0;pulse;4
3;1;counter3;3;2;3;;;1;0;pulse;4
4;1;counter4;3;2;4;;;1;0;pulse;4
5;2;output1;1;2;1;;;1;0;;4
6;2;output2;1;2;2;;;1;0;;4
7;2;output3;1;2;3;;;1;0;;4
8;2;output4;1;2;4;;;1;0;;4
9;3;input1;1;2;1;;;1;0;;8
10;3;input2;1;2;2;;;1;0;;8
11;3;input3;1;2;3;;;1;0;;8
12;3;input4;1;2;4;;;1;0;;8
}
```

prefixID_MODBUS_TYPE2.ini

```
# Définition des trames
#
Id;Name;ReadFctCode;WriteFctCode;StartReg;NbReg;EnableReading;EnableWriting;Option1;Option
2
Modbus_RequestsTables={
1;voltage;3;0;0;4;1;2
}
# Définition des variables
# Id;ReqId;Name;Type;Signed;Position;Option1;Option2;CoeffA;CoeffB;Unit;Action;
Modbus_VariablesTables={
1;1;U1;3;2;1;;;10;0;V;2
2;1;U2;3;2;2;;;10;0;V;2
}
```

Côté serveur, il faut faire le lien entre les données reçues et les fichiers de définition correspondant.

Après la mise en forme des données on obtient les résultats suivant :

Equipement à l'adresse 1 :

Valeur des compteurs :

	counter 1	counter2	counter3	counter4
27/03/13-09:45:00	32 pulse	52 pulse	5 pulse	102 pulse
27/03/13-10:00:00	35 pulse	57 pulse	5 pulse	108 pulse

Etat des sorties :

	output1	output2	output3	output4
27/03/13-09:45:00	1	0	1	0
27/03/13-10:00:00	1	1	0	0

Etat des entrées :

	input1	input2	input3	input4
27/03/13-09:45:00	0	0	0	0
27/03/13-10:00:00	0	0	0	1

Equipement à l'adresse 2 :

Valeur des tensions mesurées:

	U1			U2		
	min	max	moy	min	max	moy
27/03/13-09:45:00	160 V	320 V	260 V	520 V	580 V	540.00 V
27/03/13-10:00:00	40 V	60 V	55 V	1020 V	1080 V	1050.00 V

10.6 Les alarmes Modbus

Une variable déclarée en tant qu'alarme (champ action = 8 de la variable dans le fichier de définition) provoque le déclenchement d'une alarme sur changement d'état. Cette alarme est enregistrée dans un fichier au format CSV. Ce fichier est compressé au format GZ puis déposé sur le serveur FTP au prochain point d'acquisition.

10.6.1 Syntaxe du nom de fichier d'alarme

Le fichier d'alarmes déposé sur le serveur FTP respecte le format suivant :

```
prefixID_AL_AAMMJJ_hhmmss.csv.gz
```

Avec :

prefixID : identifiant de la passerelle.

AAMMJJ_hhmmss : horodatage de l'archive au format « année-mois-jour-heure-minute-seconde ».

10.6.2 Format des alarmes

Le fichier d'alarme CSV déposé peut contenir plusieurs alarmes de sources différentes. Il se présente sous le format suivant :

```
date-time_1;AlarmSource1;fileDefinitionName;deviceName;indexVariable,value
date-time_N;AlarmSourceN;fileDefinitionNameN;deviceNameN;indexVariableN,valueN
```

Avec:

date-time_N: horodatage du déclenchement de l'alarme au format AA/MM/JJ-hh:mm:ss
AlarmSourceN: source du déclenchement de l'alarme : ici MODBUS.
fileDefinitionName_N : nom du fichier de définition associé à l'équipement déclencheur.
deviceNameN : nom de l'équipement déclencheur
indexVariableN: index de la variable à l'origine de l'alarme
valueN: valeur de la variable à l'origine de l'alarme

10.6.3 Exemple de fichier d'alarmes

Réception du fichier d'alarme prefixID_AL_130327_100305.csv.gz suite au changement d'état des entrées input4 et input2 de l'équipement Slave 1.

Le fichier contient les informations suivantes :

```
27/03/13-10:01:16;MODBUS;prefixID_MODBUS_TYPE1.ini;Slave 1;12;1
27/03/13-10:01:18;MODBUS;prefixID_MODBUS_TYPE1.ini;Slave 1;10;0
```

Avec le fichier de définition prefixID_MODBUS_TYPE1.ini :

```
Modbus_RequestsTables={
1;counter;3;0;0;4;1;2
2;output;1;5;4;4;1;1
3;input;2;0;8;4;1;2
}
# Définition des variables
# Id;ReqId;Name;Type;Signed;Position;Option1;Option2;CoeffA;CoeffB;Unit;Action;
Modbus_VariablesTables={
1;1;counter1;3;2;1;;;1;0;pulse;4
2;1;counter2;3;2;2;;;1;0;pulse;4
3;1;counter3;3;2;3;;;1;0;pulse;4
4;1;counter4;3;2;4;;;1;0;pulse;4
5;2;output1;1;2;1;;;1;0;;4
6;2;output2;1;2;2;;;1;0;;4
7;2;output3;1;2;3;;;1;0;;4
8;2;output4;1;2;4;;;1;0;;4
9;3;input1;1;2;1;;;1;0;;8
10;3;input2;1;2;2;;;1;0;;8
11;3;input3;1;2;3;;;1;0;;8
12;3;input4;1;2;4;;;1;0;;8
}
```


Fichier prefixID_daq.ini:

```
MODBUS_Addr[0]=1
MODBUS_Name[0]=Slave1
MODBUS_FileDefName[0]=prefixID_MODBUS_TYPE1.ini
```

Après analyse du fichier d'alarme de des fichiers de définition on obtient le résultat suivant :

Passage à 1, le 27/03/13 à 10:01:16, de la variable input4 indexée 12 de l'équipement nommé Slave1.

Passage à 0, le 27/03/13 à 10:01:18, de la variable input2 indexée 10 de l'équipement nommé Slave1.

10.7 Ecriture de variables Modbus via fichier de commandes

Certaines tâches, appelées commandes, peuvent être demandées à distance à la WebdynSun. Ces commandes sont transmises à la passerelle sous forme de fichiers déposés sur le serveur FTP (prefixID_CMD.csv). Ce fichier peut contenir plusieurs types de commandes dont la commande d'écriture de variables Modbus. Il est supprimé du serveur par la passerelle après récupération en local. Après exécutions des commandes un fichier d'acquiescement est envoyé au serveur (prefixID_ACK_AAMMJ_hhmmss.csv).

Fichier de commandes : prefixID_CMD.csv.

Les paramètres des commandes diffèrent selon le type de commande envoyée comme indiqué ci-dessous :

index;MODBUS;indexDevice;indexVariable;value

Avec :

index	1 à N : Identifiant unique permettant d'identifier la commande
indexDevice	Index de l'équipement modbus à paramétrer. Cet index correspond à l'index utilisé dans le fichier daq.ini pour déclarer l'équipement (0 à N).
indexVariable	Index de la variable modbus à paramétrer Cet index correspond à l'index d'identification de variable présent dans le fichier de définition de l'équipement à paramétrer.
value	Valeur en décimal ou ASCII du paramètre.

Fichier d'acquiescement : prefixID_ACK_AAMMJ_hhmmss.csv.

Le fichier d'acquiescement reprend le fichier de commandes avec l'ajout d'un horodatage et de l'acquiescement :

Date-time;index;MODBUS;indexDevice;indexVariable;value;ack

Avec ack=OK ou ERROR.