



Note d'application

Script « Local Display »

Introduction

Cette note d'application décrit comment mettre en œuvre le script « **local display** ».

Ce script permet la mise à disposition des informations générales de la centrale photovoltaïque :

- Puissance active instantanée
- Énergie totale produite depuis la mise en service de la centrale
- Énergie journalière
- Taux d'autoconsommation (nécessite l'utilisation d'un compteur)
- Économies de CO2 totale ;
- Économies de CO2 journalière.

Ces informations peuvent être régulièrement transmises à un ou plusieurs équipements Modbus comme des afficheurs Siebert.

La présence d'un afficheur n'est pas nécessaire au fonctionnement du script.

Ces informations peuvent également être disponibles par interrogation de la WebdynSunPM via requêtes Modbus ou HTTPs.

Ce script est disponible gratuitement et son code source est accessible, il peut être librement adapté ou modifié.

Le script est disponible dans la bibliothèque interne de la WebdynSunPM à partir de la version 5.0.10.

Il peut être téléchargé à partir du lien suivant : <https://www.webdyn.com/download/LocalDisplay.zip>

Principe de fonctionnement

La plupart des informations sont générées par cumul des informations individuelles de chaque onduleur. Elles doivent donc être disponibles dans les données mises à disposition par les onduleurs.

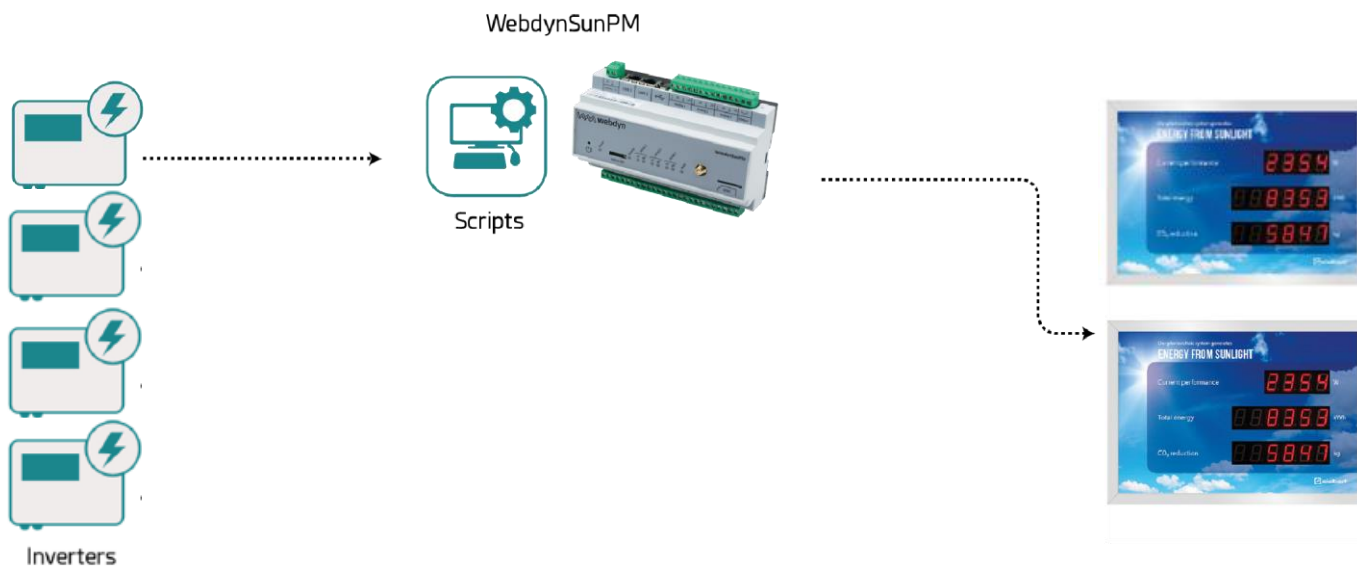
Le calcul du taux d'autoconsommation nécessite l'ajout d'un compteur afin de connaître la consommation locale du site.

Deux options, sont disponibles :

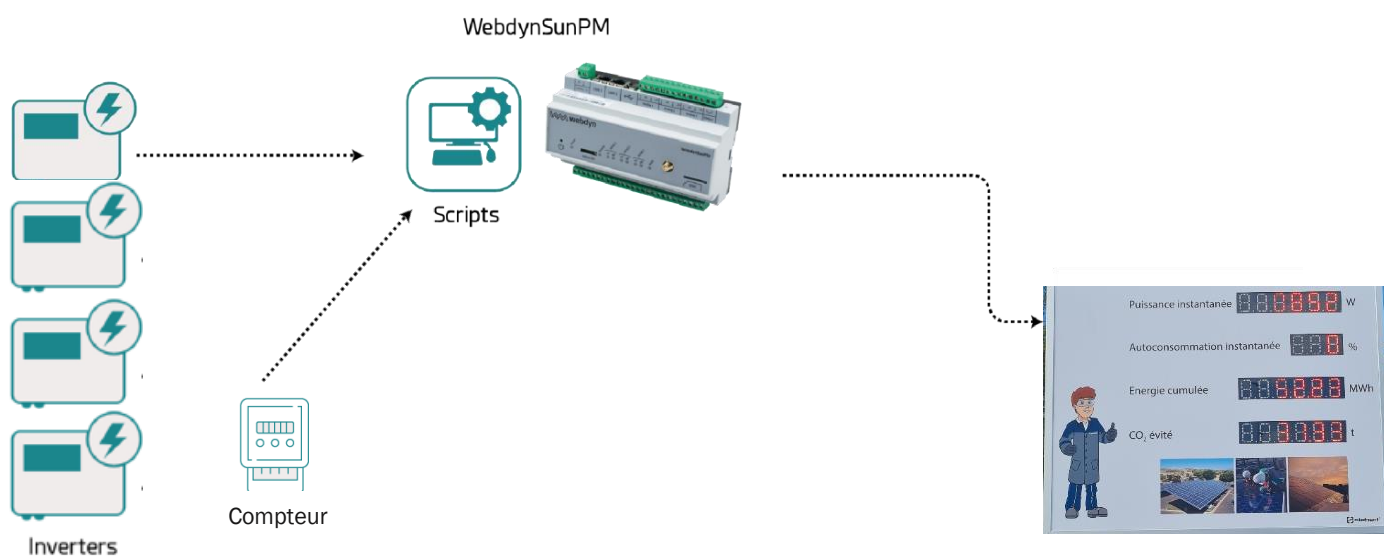
- Soit un compteur général au point d'injection sur le réseau ;
- Soit un compteur de la consommation locale.

Le calcul des économies d'émission de CO2 s'effectue par simple application d'un coefficient sur les données d'Énergie cumulée.

Standard



Taux d'autoconsommation



Prérequis

La mise à jour de la WebdynSunPM dans la dernière version de firmware disponible est conseillée.

Le script est présent dans la bibliothèque de script de la WebdynSunPM à partir de la version 5.0.10. Toutefois il peut être récupéré en suivant le lien ci-après et importé via l'interface Web ou le serveur : <https://www.webdyn.com/download/LocalDisplay.zip>

La connaissance des principes de base du fonctionnement de la WebdynSunPM est fortement recommandée.

Reférez-vous au manuel d'utilisation de la WebdynSunPM (https://www.webdyn.com/wp-content/uploads/2024/10/WebdynSunPM-Manuel-utilisateur_FR_V5.03.pdf) pour acquérir les connaissances suivantes :

- Chapitre **§3.2.3.2.2.1 Ajouter un équipement** page 102
- Chapitre **§3.1.2.2.2 Contenu du fichier de définition** page 66
- Chapitre **§3.2.4.1 Importer un service ou une licence** page 151
- Chapitre **§3.1.2.1.4 Fichier « <UID>_scl.ini »** page 63
- Virtual devices et Modbus Esclave

Les paramètres décrit ci-dessous sur les fichiers de définition des onduleurs, du compteur ou des afficheurs sont déjà réalisés dans la plupart des fichiers intégrés dans la bibliothèque interne de la WebdynSunPM.

Dans de tels cas, l'utilisation du script ne nécessite pas de paramétrage spécifique supplémentaire sur les fichiers de définition.

Paramétrage des onduleurs

Dans chaque fichier de définition utilisé par les onduleurs connectés au concentrateur, les éléments suivants sont requis :

- **Catégorie (identification des équipements)**

Dans l'en-tête du fichier de définition, le champs catégorie (première ligne, 2eme colonne) doit être défini avec le nom « **Inverter** ». C'est cette dénomination qui permet d'identifier tous les onduleurs à prendre en compte.

- **Tags (identification des variables)**

Tous les équipements identifiés par la catégorie « **Inverter** » devront avoir les tags suivants :

Tag « **RealPower** » Permet d'identifier la variable contenant la puissance active instantané.

Tag « **EnergyTotal** » : Permet d'identifier la variable contenant l'énergie produite par l'onduleur depuis sa mise en service.

Tag « **EnergyDay** » (selon les onduleurs) : Permet d'identifier la variable contenant l'énergie produite par l'onduleur depuis le début de la journée.



Il convient de vérifier que les unités des variables identifiées par un même tag soient identiques entre les onduleurs et cohérentes avec l'affichage souhaité.

Tips : Il est tout à fait possible d'utiliser ce script afin d'effectuer le cumul de variables différentes de celles préconisées ci-dessus en plaçant astucieusement les tags.

Les « Tag » doivent être renseignés dans la colonne G (champs 7) du fichier de définition de l'équipement.

Exemple :

	A	B	C	D	E	F	G	H	I	J	K
1	modbusTC?	Inverter	HUAWEI	V4							
77	76	3	32078	I32		Active power peak of current day		0.001	0 kW		4
78	77	3	32080	I32		Active power	RealPower	1	0 W		4
79	78	3	32082	I32		Reactive_power		1	0 Var		4
80	79	3	32084	I16		Power factor		0.001	0		4
81	80	3	32085	U16		Frequency		0.01	0 Hz		4
82	81	3	32086	U16		Inverter efficiency		0.01	0 %		4
83	82	3	32087	I16		Cabinet temperature		0.1	0 Â°C		4
84	83	3	32088	U16		Insulation resistance		0.001	0 M/Ohm		4
85	84	3	32089	U16		Device status		1	0		9
86	85	3	32090	U16		Fault code		1	0		4
87	86	3	32091	U32		Startup time		1	0 s		4
88	87	3	32093	U32		Shutdown time		1	0 s		4
89	88	3	32106	U32		Energy_total	EnergyTotal	0.01	0 kWh		4
90	89	3	32114	U32		Energy yield of current day	EnergyDay	0.01	0 kWh		4
91	90	3	32116	U32		Energy yield of the month		0.01	0 kWh		4

Paramétrage du compteur

Le paramétrage d'un compteur n'est nécessaire que si vous souhaitez afficher un taux d'autoconsommation.

L'utilisation de la fonctionnalité de calcul du taux d'autoconsommation nécessite l'édition du script « local display » afin de modifier le paramètre **AutoconsoRatio** ligne 24.

```
23 local CarbCoef = 0.7
24 local AutoconsoRatio = 0
25
```

• Nommage (identification du type de compteur)

Le calcul du taux d'autoconsommation nécessite l'ajout d'un compteur afin de connaître la consommation locale du site.

Deux cas, sont possibles :

- Soit un compteur général au point d'injection sur le réseau auquel on ajoutera la puissance provenant des onduleurs afin d'obtenir une valeur de consommation locale. Dans ce cas il faudra nommer le compteur « **MainMeter** »
- Soit un compteur fournissant directement la consommation locale. Dans ce cas il faudra nommer le compteur « **ConsoMeter** »

• Tags (identification des variables)

Dans le fichier de définition du compteur, il faut identifier la variable de puissance en lui affectant le tag « **ActivePowSumkW** », puis vérifier qu'elle est exprimée dans la même unité que les puissances provenant des onduleurs, ajuster le coefficient A (gain) si besoin.

Le « Tag » doit être renseigné dans la colonne G (champs 7) du fichier de définition de l'équipement. Si besoin référez-vous au manuel du fabricant de l'équipement pour identifier la variable souhaitée.



La puissance fournie par le compteur doit être la même que celle
La convention de signe utilisée pour les compteurs doit être la suivante :
Les valeurs positive indiquent un soutirage et les valeurs négatives une injection.

Exemple :

catégorie « meter »

Tag des variables de puissance instantanées

	A	B	C	D	E	F	G	H	I	J	K
1	ModbusTCP	Meter	Janitza	Generic							
2	1	4	19000	F32		ULN[0]		1	0 V		4
3	2	4	19002	F32		ULN[1]		1	0 V		4
4	3	4	19004	F32		ULN[2]		1	0 V		4
5	4	4	19006	F32		ULL[0]		1	0 V		4
6	5	4	19008	F32		ULL[1]		1	0 V		4
7	6	4	19010	F32		ULL[2]		1	0 V		4
8	7	4	19012	F32		ILN[0]		1	0 A		4
9	8	4	19014	F32		ILN[1]		1	0 A		4
10	9	4	19016	F32		ILN[2]		1	0 A		4
11	10	4	19018	F32		I_SUM3		1	0 A		4
12	11	4	19020	F32		PLN[0]	ActivePow1kW	0.001	0 kW		4
13	12	4	19022	F32		PLN[1]	ActivePow2kW	0.001	0 kW		4
14	13	4	19024	F32		PLN[2]	ActivePow3kW	0.001	0 kW		4
15	14	4	19026	F32		P_SUM3	ActivePowSumkW	0.001	0 kW		4



Paramétrage des afficheurs

Dans chaque fichier de définition utilisé par un équipement devant recevoir les données calculées par le script, comme un afficheur Siebert, les éléments suivants sont requis :

- **Catégorie (identification des équipements)**

Dans l'en-tête du fichier de définition, le champs catégorie (première ligne, 2eme colonne) doit être défini avec le nom « **Display** ». C'est cette dénomination qui permet d'identifier tous les afficheurs à adresser.

- **Tags (identification des variables)**

Tous les équipements identifiés par la catégorie « **Display** » devront impérativement avoir les tags suivants :

Tag « **Pac-Tag** » Permet d'identifier la variable qui recevra le cumul des puissances actives instantanées des onduleurs (identifié par le tag **RealPower**).

Tag « **Eac-Tag** » : Permet d'identifier la variable qui recevra le cumul des énergies produites par les onduleurs depuis leur mise en service. (Identifié par le tag **EnergyTotal**)

Les tags suivants sont optionnels :

Tag « **EacDay-Tag** » : Permet d'identifier la variable qui recevra le cumul des énergies produite par les onduleurs depuis le début de la journée. (Identifié par le tag **EnergyDay**). L'utilisation de ce tag nécessite que tous les onduleurs identifiés puissent fournir leur production journalière et donc que le tag **EnergyDay** soit déclaré dans chaque fichier de définition onduleur.

Lorsque le mode **Autoconsommation** est utilisé, c'est la variable identifiée par ce tag « **EacDay-Tag** » qui reçoit la valeur d'autoconsommation.

Tag « **Carb-Tag** » : Permet d'identifier la variable qui recevra le calcul des économies d'émission de CO2 depuis la mise en service de la centrale (Valeur envoyée à **Eac-Tag** multiplié par le coefficient carbone qui vaut 0.7 par défaut). Cette valeur est souvent calculée directement par l'onduleur, il convient alors de ne pas saisir ce tag afin que la valeur ne soit pas transmise.

Tag « **CarbDay-Tag** » : Permet d'identifier la variable qui recevra le calcul des économies d'émission de CO2 depuis la mise en service de la centrale (Valeur envoyée à **EacDay-Tag** multiplié par le coefficient carbone qui vaut 0.7 par défaut). Cette valeur est souvent calculée directement par l'onduleur, il convient alors de ne pas saisir ce tag afin que la valeur ne soit pas transmise.

Les « Tag » doivent être renseignés dans la colonne G (champs 7) du fichier de définition de l'équipement.

Il n'est pas nécessaire que tous les afficheurs aient les mêmes tags, chaque afficheur peut être configuré avec les tags qui lui sont spécifiques. Il n'est cependant pas possible d'avoir l'Energie journalière sur un afficheur et le taux d'autoconsommation sur un autre.

Exemple :

catégorie « Display »

	A	B	C	D	E	F	G	H	I	J	K
1	modbusR	U Display	3 EBERT	XC4XX	1						
2		1	3	1 U32_W		Puissance instantanée	Pac-Tag	1	0		4
3		2	3	3 U32_W		Energie journalière		1	0		4
4		3	3	5 U32_W		Energie cumulée	Eac-Tag	1	0		4
5		4	3	7 U32_W		Economie d'émission de CO2	Carb-Tag	1	0		4
6		5	3	9 U32_W		Economie d'émission de CO2 journalière	CarbDay-Tag	1	0		4

Taux de rafraîchissement et coefficients de mise à l'échelle.

Par défaut le taux de rafraîchissement des équipements qui reçoivent les données est de **10s**.
Ce taux peut être modifié en éditant le script et en ajustant la valeur du paramètre
« DisplayRefreshFrequency » ligne 19.

```
19 local DisplayRefreshFrequency = 10
20 local Pac_coef = 1
21 local Eac_coef = 1
22 local EacDay_coef = 1
```

Il est également possible d'ajuster les valeurs des données envoyées : Puissance Active, Energie cumulée et Energie journalière en modifiant leurs coefficients respectifs Pac_coef (ligne 20), Eac_coef (ligne 21) et EacDay_coef (ligne 22).

Il peut être utile de modifier ces coefficients lorsque l'utilisation d'un autre script comme celui de régulation d'injection nécessite l'utilisation d'une unité différente de celle souhaitée pour l'affichage. Il devient alors impossible d'utiliser les coefficients du fichier de définition des onduleurs. C'est une solution alternative à l'utilisation des coefficients du fichier de définition des afficheurs.

Coefficient de calcul des émissions de CO2

Le coefficient appliqué aux variables d'Energie afin de calculer les économies de CO2 peut être saisi comme paramètres du script :

GenSet-V1_04	Generator	1.04	Missing/Invalid	Disabled		
LocalDisplay	Local Display	8	Not required	Disabled		
RelayControl	Relay Control	2.0	Not required	D		
SendCommand	Send Command	1.0	Not required	D		

Script arg

Script logs

View

Il remplacera alors la valeur par défaut de 0.7 qui peut aussi être modifié directement dans le script ligne 23 :

```
22 local EacDay_coef = 1
23 local CarbCoef = 0.7
24 local AutosecRatio =
```


Script

Chargement du Script

Le script est présent dans la bibliothèque de script de la WebdynSunPM à partir de la version 5.0.10. Toutefois il peut être récupéré en suivant le lien ci-après et importé via l'interface Web ou le serveur : <https://www.webdyn.com/download/LocalDisplay.zip>

Il peut être chargé via l'interface web embarqué en cliquant sur le bouton Add script/licence file.

LocalDisplay	Local Display	8	Not required	Disabled	<input type="checkbox"/>	⋮
RelayControl	Relay Control	2.0	Not required	Disabled	<input type="checkbox"/>	⋮
SendCommand	Send Command	1.0	Not required	Disabled	<input type="checkbox"/>	⋮

Add script/licence file

Paramétrage du Script

Le coefficient carbone peut être ajusté par passage de paramètre (par défaut la valeur est 0.7).

GenSet-V1_04	Generator	1.04	Missing/Invalid	Disabled	<input type="checkbox"/>	⋮
LocalDisplay	Local Display	8	Not required	Disabled	<input type="checkbox"/>	⋮
RelayControl	Relay Control	2.0	Not required	Disabled	<input type="checkbox"/>	⋮
SendCommand	Send Command	1.0	Not required	Disabled	<input type="checkbox"/>	⋮

Add arguments

Script arguments

Enter arguments here

Cancel Save

Démarrage du Script

Après avoir activé le script au moyen du bouton en fin de ligne vous pouvez accéder au log du script :

LocalDisplay	Local Display	8	Not required	Enabled	<input checked="" type="checkbox"/>	⋮
--------------	---------------	---	--------------	---------	-------------------------------------	---

LocalDisplay	Local Display	8	Not required	Enabled	
RelayControl	Relay Control	2.0	Not required	D	Script arg
SendCommand	Send Command	1.0	Not required	D	Script logs
					View

```

2024-11-04 14:51:42 [LocalDisplay.lua 27] Local Display Script V8 Started
2024-11-04 14:51:42 [LocalDisplay.lua 177] No Carbon coefficient parameter, Using default value 0.7
2024-11-04 14:51:42 [LocalDisplay.lua 43] No change on template WPM0117B5_Script_LocalDisplay.csv
2024-11-04 14:51:42 [LocalDisplay.lua 68] 3 inverters found
2024-11-04 14:51:42 [LocalDisplay.lua 73] Inverter 1(INV2) has tag: RealPower
2024-11-04 14:51:42 [LocalDisplay.lua 80] Inverter 1(INV2) has tag: EnergyTotal
2024-11-04 14:51:42 [LocalDisplay.lua 87] Inverter 1(INV2) has tag: EnergyDay
2024-11-04 14:51:42 [LocalDisplay.lua 73] Inverter 2(INV3) has tag: RealPower
2024-11-04 14:51:42 [LocalDisplay.lua 80] Inverter 2(INV3) has tag: EnergyTotal
2024-11-04 14:51:42 [LocalDisplay.lua 87] Inverter 2(INV3) has tag: EnergyDay
2024-11-04 14:51:42 [LocalDisplay.lua 73] Inverter 0(INV1) has tag: RealPower
2024-11-04 14:51:42 [LocalDisplay.lua 80] Inverter 0(INV1) has tag: EnergyTotal
2024-11-04 14:51:42 [LocalDisplay.lua 87] Inverter 0(INV1) has tag: EnergyDay
2024-11-04 14:51:42 [LocalDisplay.lua 108] 1 Displays found
2024-11-04 14:51:42 [LocalDisplay.lua 112] Display0(Afficheur) has tag: Pac-Tag
2024-11-04 14:51:42 [LocalDisplay.lua 118] Display0(Afficheur) has tag: Eac-Tag
2024-11-04 14:51:42 [LocalDisplay.lua 127] Display0(Afficheur) no tag: EacDay-Tag
2024-11-04 14:51:42 [LocalDisplay.lua 134] Display0(Afficheur) no tag: Carb-Tag
2024-11-04 14:51:42 [LocalDisplay.lua 141] Display0(Afficheur) no tag: CarbDay-Tag

```

Au démarrage du script, celui-ci contrôle la présence des équipements et des tags nécessaires au fonctionnement.

Si un tag est manquant, le script apparait en erreur :

LocalDisplay	Local Display	8	Not required	Error		
--------------	---------------	---	--------------	-------	---	---

Et les logs indiquent le(s) tag(s) manquant.

Logs	
2024-11-04 13:49:53	[LocalDisplay.lua 27] Local Display Script V8 Started
2024-11-04 13:49:53	[LocalDisplay.lua 177] No Carbon coefficient parameter, Using default value 0.7
2024-11-04 13:49:53	[LocalDisplay.lua 43] Creation or update of template WPM0117B5_Script_LocalDisplay.csv
2024-11-04 13:49:53	[LocalDisplay.lua 68] 3 inverters found
2024-11-04 13:49:53	[LocalDisplay.lua 75] Inverter 1(INV2) missing tag RealPower
2024-11-04 13:49:53	[LocalDisplay.lua 80] Inverter 1(INV2) has tag: EnergyTotal
2024-11-04 13:49:53	[LocalDisplay.lua 87] Inverter 1(INV2) has tag: EnergyDay
2024-11-04 13:49:53	[LocalDisplay.lua 75] Inverter 2(INV3) missing tag RealPower
2024-11-04 13:49:53	[LocalDisplay.lua 80] Inverter 2(INV3) has tag: EnergyTotal
2024-11-04 13:49:53	[LocalDisplay.lua 87] Inverter 2(INV3) has tag: EnergyDay
2024-11-04 13:49:53	[LocalDisplay.lua 75] Inverter 0(INV1) missing tag RealPower
2024-11-04 13:49:53	[LocalDisplay.lua 80] Inverter 0(INV1) has tag: EnergyTotal
2024-11-04 13:49:53	[LocalDisplay.lua 87] Inverter 0(INV1) has tag: EnergyDay
2024-11-04 13:49:53	[LocalDisplay.lua 104] No category with name Display
2024-11-04 13:49:53	[LocalDisplay.lua 106] No display screen declared
2024-11-04 13:49:53	[LocalDisplay.lua 167] Config error
2024-11-04 13:49:53	[LocalDisplay.lua 168] STOP

Si le script s'exécute correctement les logs indiquent les valeurs calculées par le script et a qui elles sont envoyées.

```
2024-11-04 14:56:49 [LocalDisplay.lua 305] *****
2024-11-04 14:57:00 [LocalDisplay.lua 265] SendtoDisplay
2024-11-04 14:57:00 [LocalDisplay.lua 282] UpdateModbusSlave
2024-11-04 14:57:00 [LocalDisplay.lua 296] Instant Power: 20000 kW
2024-11-04 14:57:00 [LocalDisplay.lua 297] Total Energy: 199.99999552965 kWh
2024-11-04 14:57:00 [LocalDisplay.lua 301] Day Energy: 299.99999329448 kWh
2024-11-04 14:57:00 [LocalDisplay.lua 303] Co2 total savings: 139.99999687076 Co2
2024-11-04 14:57:00 [LocalDisplay.lua 304] Co2 Day savings: 209.99999530613 Co2
2024-11-04 14:57:00 [LocalDisplay.lua 305] *****
2024-11-04 14:57:12 [LocalDisplay.lua 265] SendtoDisplay
2024-11-04 14:57:12 [LocalDisplay.lua 282] UpdateModbusSlave
2024-11-04 14:57:12 [LocalDisplay.lua 296] Instant Power: 20000 kW
2024-11-04 14:57:12 [LocalDisplay.lua 297] Total Energy: 199.99999552965 kWh
2024-11-04 14:57:12 [LocalDisplay.lua 301] Day Energy: 299.99999329448 kWh
2024-11-04 14:57:12 [LocalDisplay.lua 303] Co2 total savings: 139.99999687076 Co2
2024-11-04 14:57:12 [LocalDisplay.lua 304] Co2 Day savings: 209.99999530613 Co2
2024-11-04 14:57:12 [LocalDisplay.lua 305] *****
2024-11-04 14:57:23 [LocalDisplay.lua 265] SendtoDisplay
2024-11-04 14:57:23 [LocalDisplay.lua 282] UpdateModbusSlave
2024-11-04 14:57:23 [LocalDisplay.lua 296] Instant Power: 20000 kW
2024-11-04 14:57:23 [LocalDisplay.lua 297] Total Energy: 199.99999552965 kWh
2024-11-04 14:57:23 [LocalDisplay.lua 301] Day Energy: 299.99999329448 kWh
2024-11-04 14:57:23 [LocalDisplay.lua 303] Co2 total savings: 139.99999687076 Co2
2024-11-04 14:57:23 [LocalDisplay.lua 304] Co2 Day savings: 209.99999530613 Co2
2024-11-04 14:57:23 [LocalDisplay.lua 305] *****
```

UpdateModbusSlave indique que les valeurs des registres de l'esclave Modbus sont mises a jour

SendToDisplay indique que les valeurs sont envoyées aux afficheurs de la catégorie « Display ».

Si aucun afficheur n'est déclaré, cette ligne n'apparaît pas.

Virtual Device

Il est possible d'associer à un script un fichier de définition, le script se comporte alors comme un équipement virtuel.

C'est le cas du script « LocalDisplay » pour lequel les variables suivantes ont été définies :

```
6  --*****
7  -- Virtual Device declaration
8  --*****
9  virtualDevice = {
10     U32_Pac = 0,
11     U32_Eac = 0,
12     U32_EacDay = 0,
13     U32_Carb = 0,
14     U32_CarbDay = 0
15 }
16 --*****
17 local VirtualDevice_Enable = 1
18 local VirtualDevice_Frequency = 0
```

Cette déclaration entraîne la génération d'un fichier de définition : **WPMxxxxx_Script_LocalDisplay.csv** contenant ces variables qui sera déposé sur le serveur FTP dans le répertoire /DEF

Voici le contenu de ce fichier :

```

1 none;Script;Script;LocalDisplay
2 1;;;U32;;Carb;Carb;1.000000;0.000000;;4
3 2;;;U32;;CarbDay;CarbDay;1.000000;0.000000;;4
4 3;;;U32;;Eac;Eac;1.000000;0.000000;;4
5 4;;;U32;;EacDay;EacDay;1.000000;0.000000;;4
6 5;;;U32;;Pac;Pac;1.000000;0.000000;;4
7

```

Ces variables peuvent être envoyée dans les fichiers de donnée standard de la WebdynSunPM afin qu’elles soient exploitées par le portail comme celle d’un équipement supervisé.

Pour se faire il faut renseigner la fréquence d’enregistrement de ces variables qui est indépendante de la fréquence de rafraîchissement des données de 10s. Par défaut une fréquence à 0 n’écrit pas dans le fichier de données.

```

17 local VirtualDevice_Enable = 1
18 local VirtualDevice_Frequency = 0

```

Ce fichier de définition d’équipement virtuel permet également de renseigner les valeurs calculées par le script et tant qu’entrée de l’esclave Modbus (voir fichier Modbus esclave ci-dessous).

Paramétrage de l’esclave Modbus

Voir chapitre 3.2.5.1.4 Modbus Slave sur le manuel utilisateur de la WebdynSunPM

L’activation de la fonctionnalité Modbus esclave s’effectue depuis le fichier daq.csv présent dans le répertoire /CONFIG du serveur distant tel que décrit dans l’exemple de fichier ci-dessous.

	A	B	C	D	E	F	G	H
1	type	pin	apn	login	password	authenticatic	server	dns
11	SERIAL1	9600		8 N		1	2 modbusRTU	0
12	SERIAL2	9600		8 N		1	2 modbusRTU	0
13	SERIAL3	9600		8 N		1	2 modbusRTU	0
14								
15	ModbusSlave	1		502 WPM01375A	ModbusSlave.csv			
16								

Fichier Modbus esclave

Le fichier de définition Modbus esclave doit être déposé sur le serveur FTP dans le répertoire /DEF et nommé tel que déclaré dans le fichier daq.csv.

Ci-dessous un exemple de déclaration des données du script dans l’esclave Modbus.

	A	B	C	D	E	F	G	H	I	J	K
1	ModbusSlave										
2	1	0_4		LocalDisplay	Pac		1	0			4
3	2	2_4		LocalDisplay	Eac		1	0			4
4	3	4_4		LocalDisplay	EacDay		1	0			4
5	4	6_4		LocalDisplay	Carb		1	0			4
6	5	8_4		LocalDisplay	CarbDay		1	0			4

Il est tout à fait possible de déclarer ces variables avec des adresses de registres différentes et de les mixer avec d’autres équipements virtuels ou non.

Autres méthodes de récupération des données du script

Par requête post

Il est possible de récupérer les valeurs de la puissance active et de l'énergie totale des onduleurs avec une requête POST via une application comme POSTMAN (<https://web.postman.co/>).

Remarque : Toutes les fonctions accessibles dans le script sont accessibles par une requête post.

- Ouvrir une session sur le webservice saisissez la requête POST :
`shhttp://<@IP_WebdynSunPm>/auth ;`
- Renseigner dans l'onglet « Body » : `{ "user":"userhigh","password":"high" } ;`
- Cliquer sur l'onglet Send.

Une fois le message envoyé, un cookie est alors récupéré et ne durera qu'une minute si aucune commande n'est envoyée à la WebdynSunPM.

Ouverture d'une session avec une requête POST :

The screenshot displays the Postman interface for a REST client. At the top, the URL is `http://172.20.21.30/auth`. The method is set to `POST`. The request body is a JSON object: `{ "user": "userhigh", "password": "high" }`. The response status is `200 OK` with a response time of `35 ms` and a body size of `223 B`. The response body is a single JSON string: `"KCOwQIDTHIKTRFC"`.

Pour pouvoir récupérer les données de l'afficheur :

- Envoyer la requête post
http://<@IP_WebdynSunPm>/scripts?<nom_du_fichier_script>.<nom_de_la_fonction> ;
- Cliquer sur l'onglet Send.

Requête post permettant d'obtenir l'énergie cumulé :

The screenshot shows a web browser's developer tools interface. At the top, the URL bar displays `http://172.20.21.30/scripts?LocalDisplay.GetEnergy`. Below the URL bar, the request method is set to `POST` and the request body is empty. The response status is `200 OK` with a response time of `26 ms` and a response size of `207 B`. The response body is displayed in a text area, showing a single line of text: `1`. The interface includes various tabs for request details (Params, Authorization, Headers, Body, Scripts, Tests, Settings) and response details (Body, Cookies, Headers, Test Results). The response body is currently displayed in a 'Pretty' view, showing the text `1`.

Annexe 1 : Afficheur Siebert

Ce script est souvent utilisé avec les afficheurs de la marque Siebert.

La configuration de ces afficheurs s'effectue au moyen du logiciel Siebert : « SolarDisplayConfigurator ». Cela nécessite donc le raccordement de l'afficheur vers un ordinateur sur lequel est installé cet utilitaire au moyen d'un câble USB/RS232

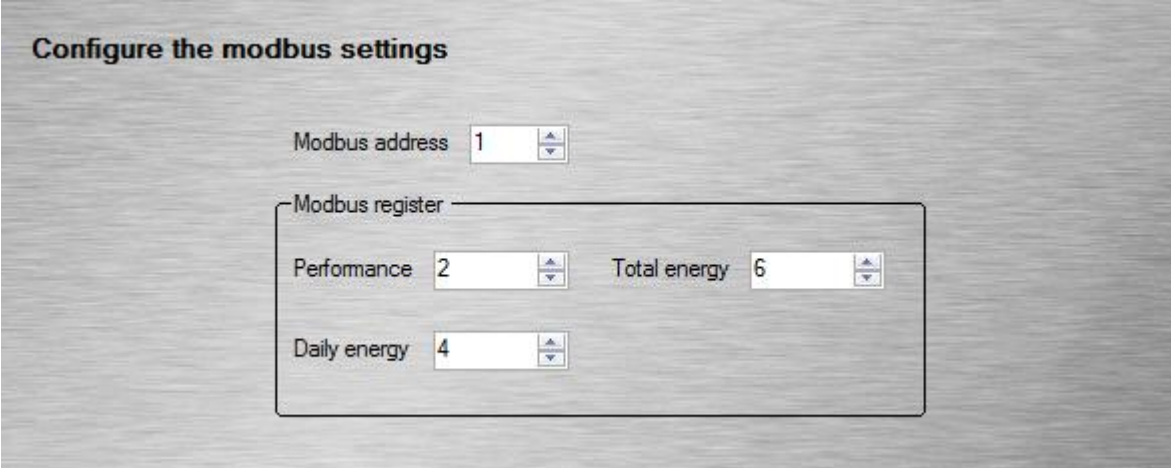
L'utilitaire vous guidera ensuite pas à pas, il convient de garder en mémoire l'adresse attribué à l'équipement ainsi que le paramétrage de la liaison RS485 afin de reporter ces informations dans la WebdynSunPM au moment de la déclaration de l'afficheur.

Par défaut :

Vitesse: 19200 Bauds
Parity: None
Address: 1

Le fichier de définition de l'afficheur Siebert intégré dans la bibliothèque de la WebdynSunPM correspond à la déclaration par défaut ci-dessous (attention il y a un décalage d'un registre)

Toute modification nécessitera l'adaptation du fichier de definition.



Configure the modbus settings

Modbus address 1

Modbus register

Performance 2 Total energy 6

Daily energy 4



Ces afficheurs ne permettent pas la lecture de leurs registres qui sont en écriture seule. Par conséquent ils apparaissent en rouge dans le Dashboard de la WebdynSunPM.

Afin de tester la communication avec ces afficheurs avant le démarrage du script il peut être pratique d'utiliser le script « **sendCommand** » tel que décrit ci-dessous.

Test de l'afficheur :

Le script Send_Command.lua est très simple d'utilisation, il faut saisir dans les paramètres :

Le nom avec lequel est déclaré l'équipement sur lequel écrire, par exemple : Afficheur.

Le tag de la variable sur laquelle écrire, par exemple : Pac_tag.

La valeur que l'on souhaite afficher, par exemple :526

Sous la forme : Afficheur ; Pac_tag ;526