Appendix for Mersen GreenString inverters via Modbus

WebdynSun

The monitoring gateway for your solar installation

CONTENTS

1 Introduction 3
2 Connecting the gateway to GreenString devices 3
2.1 Connecting at the end of the bus 4
2.2 Connecting in the middle of the bus 5
3 Configuring the Modbus devices 6
4 Configuring the WebdynSun gateway 7
4.1 Configuring the bus and declaring devices 7
4.2 Details of the GreenString definition file 7
5 GreenString functions supported 13
6 Parameter files 14
7 Data files 15
8 Alarm files 16
9 Checking that a GreenString module is operating correctly 17
10 Command files 18
10.1 Taking a photovoltaic panel online or offline 19
10.2 Configuring GreenString modules 20

1 Introduction

The present document describes the operations of the WebdynSun gateway in conjunction with GreenString devices from Mersen.

The WebdynSun gateway can handle up to 247 Modbus devices on its RS485(B) communications port. These devices are declared to the gateway via the configuration file daq.ini, accessible via an FTP server.
The set of variables to be collected and that of the Modbus requests are described for each Modbus device in a file called the definition file.

Only generic Modbus function codes are supported by the basic gateway. A development to the Modbus module of the gateway has been produced, since GreenString modules use specific “user” function codes. This development is available from version V2.05.06 and up of the WebdynSun firmware.

2 Connecting the gateway to GreenString devices

- All wiring must be carried out only by a specialised qualified electrician.
- Prior to installation, all equipment connected to the corresponding communications bus must be disconnected on both sides (DC and AC).
- Respect all safety recommendations appearing in device documentation.

Connecting inverters to the RS485 bus:

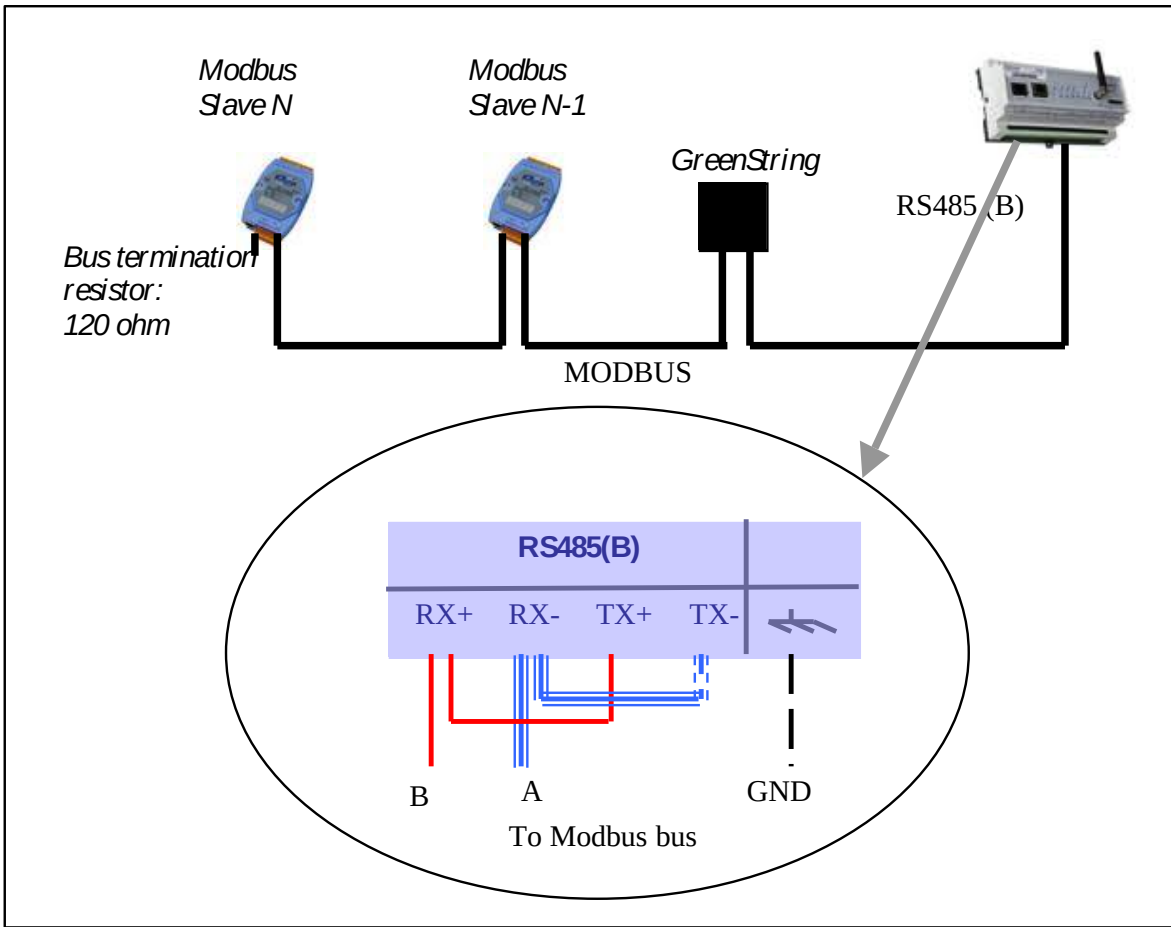
Please consult Mersen’s wiring recommendations provided in the documentation for GreenString modules.

Once the RS485 cable has been brought up to the gateway:

1. Strip around 4 cm of the outer sheath from the RS485 communications cable.
2. Trim the shielding back to match the cable sheath.
3. Bare around 6 mm of the wires.
4. Connect the conductors to the connector marked RS485(B), respecting the allocations in your RS485 communications bus.

The gateway may be connected at the end of the RS485 communications bus or in the middle of the bus.

2.1 Connecting at the end of the bus



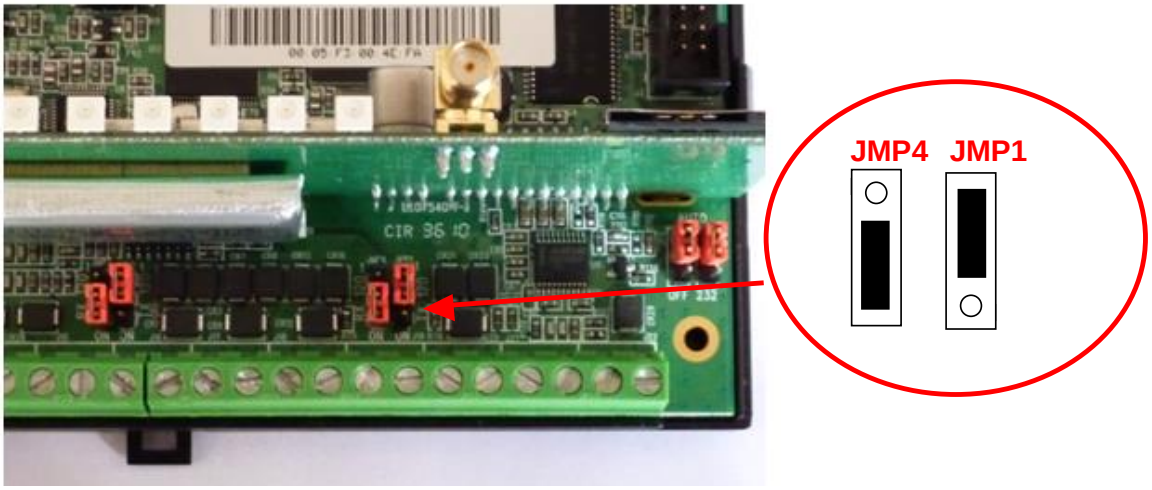
To ensure correct operation of the RS485 data bus, it must be fitted with a 120 ohm termination resistor at both ends.

Please refer to the Modbus device manufacturer's documentation to see how to provide termination on the Modbus end. At the WebdynSun end, this function is provided by configuring the jumpers JMP4 and JMP1.

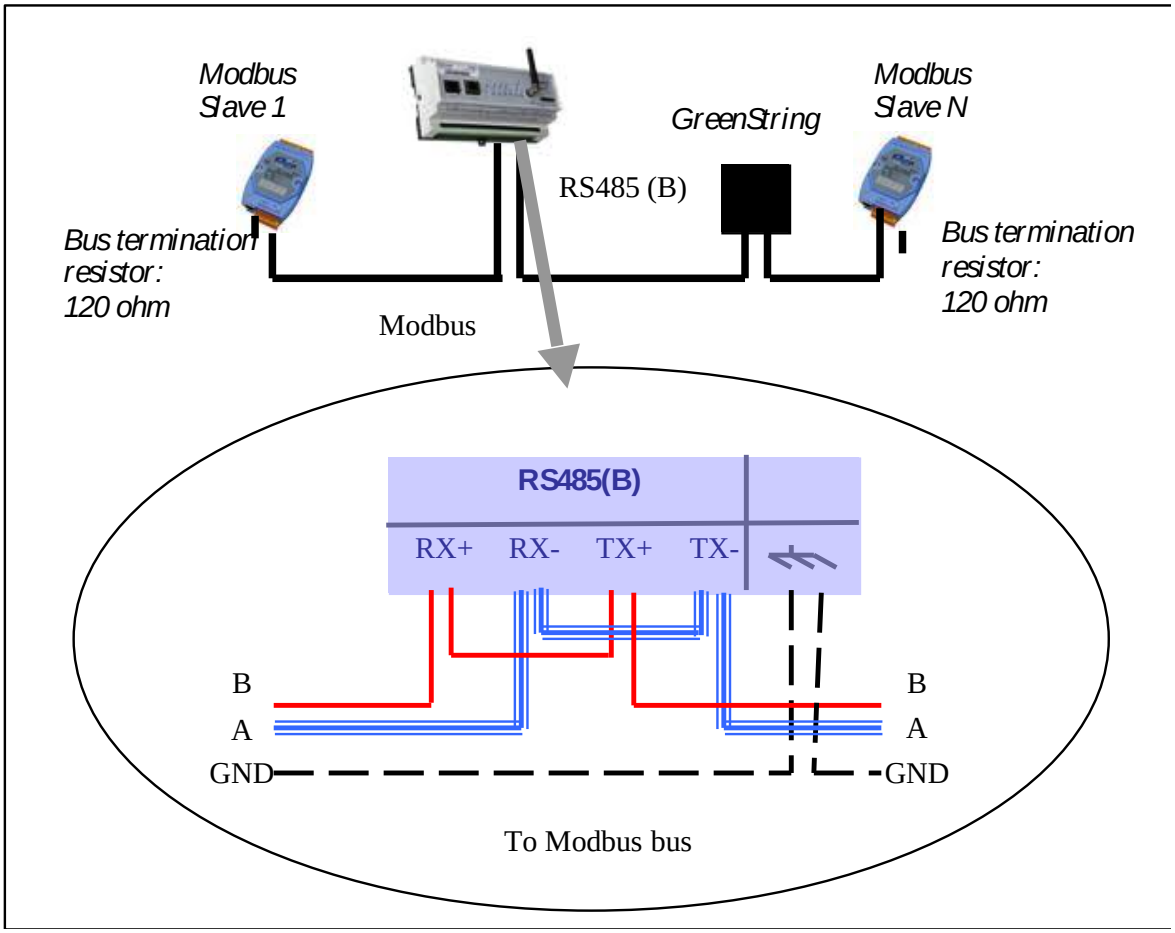
Positions of the JMP1 and JMP4 jumpers on the gateway:
Only the JMP4 jumper inside the casing needs to be set to activate the 120 ohm bus terminator plug.

	By default, the gateway is configured to be located at the end of a 4–wire RS485 bus. Consequently, the jumpers are plugged into the "Activation" position of the 120 ohm terminator. As the GreenString device uses 2–wire connection, the JMP1 jumper must be set to the OFF position.

Hardware version 2:

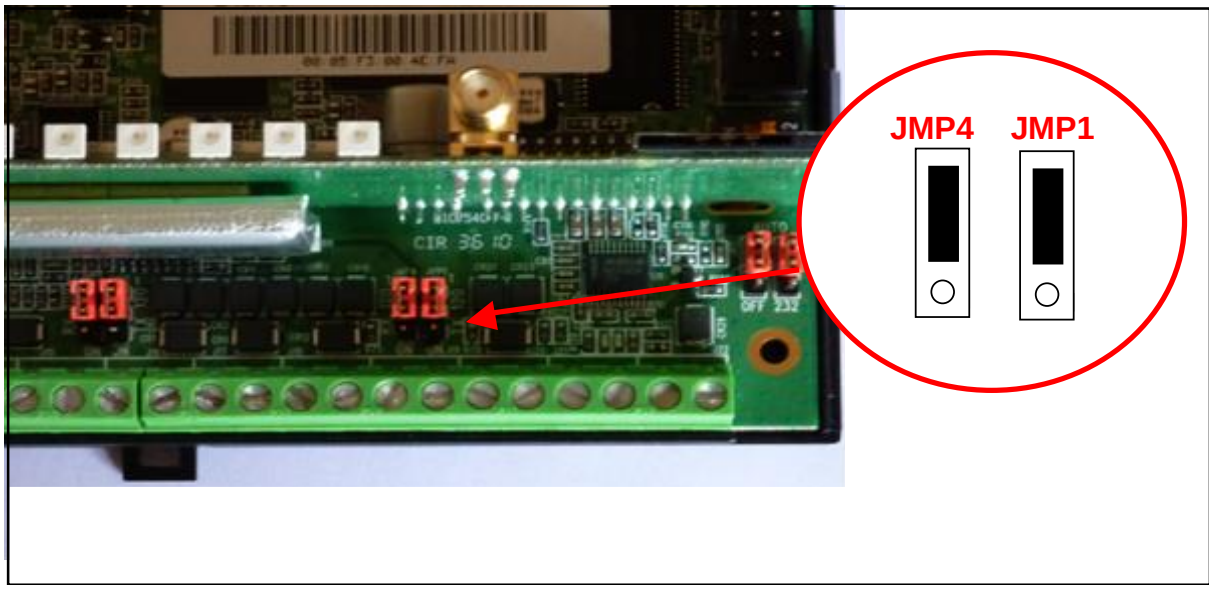


2.2 Connecting in the middle of the bus



Positions of the JMP1 and JMP4 jumpers on the gateway:
Both the JMP1 and the JMP4 jumpers inside the casing must be set to OFF to deactivate the 120 ohm bus terminator plug.

Hardware version 2:



3 Configuring the Modbus devices

To be visible to the gateway, a Modbus slave must be configured for operations via RS485, and must have an address. This address must be unique on the bus.

For the GreenString module, addressing is performed manually via three coding wheels. Please refer to the device documentation to perform this configuration.

Every Modbus device must be correctly configured before configuring the gateway.

The fact of using several devices from different manufacturers in the same installation implies a homogenous configuration. As GreenString modules are not configurable, it is important to comply with the following constraints:

- bus speed: 19200 baud
- number of data bits: 8
- number of stop bits: 1
- parity: even

4 Configuring the WebdynSun gateway

4.1 Configuring the bus and declaring devices

The WebdynSun must be configured after having correctly configured the Modbus slaves. This configuration is carried out via the FTP server.
Please refer to the User's Manual of the WebdynSun for further details of the gateway configuration mechanism.

The file "*IDsite*_daq.ini" must contain the following parameters:

Bus configuration:

Mode RS485 2-wire, baud rate: 19200 baud, 8 data bits, 1 stop bit, parity: even.

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

Declaring Modbus slaves (up to 247 devices possible:

Slave 0: GreenString Module: Specific Mersen Modbus (Type 1), at address 1, named "GREENSTRING1" and defined by the definition file MODBUS_GREENSTRING.ini.

```
MODBUS_Type[0]=1
MODBUS_Addr[0]=1
MODBUS_Name[0]=GREENSTRING1
MODBUS_FileDefName[0]=MODBUS_GREENSTRING.ini
```

…
Slave n: Standard Modbus device: Generic Modbus (Type 0), at address 2, named "SLAVE2" and defined by the definition file MODBUS_SLAVE.ini.

```
MODBUS_Type[n]=0
MODBUS_Addr[n]=2
MODBUS_Name[n]=SLAVE2
MODBUS_FileDefName[0]=MODBUS_SLAVE.ini
```

4.2 Details of the GreenString definition file

The purpose of this file is to describe the set of variables available for a Modbus device.

For each variable, it describes:
- its retrieval method: used by the gateway to capture the data idem within the inverter;
- its processing method: average, instantaneous, parameter or alarm;
- its formatting: name, units and scaling coefficient.

This file must be made available to the gateway on the FTP server.
Webdyn provides a comprehensive definition file for a GreenString module, which can in part be modified by the user as necessary.

By convention, the filename has the following format**:**
*IDSite*_MODBUS_*deviceName*.ini, where *"IDSite"* corresponds to the gateway identifier and "*deviceName"* corresponds to the name of the device.

A single file can be used for several devices.

The file consists of two declaration tables:

- The declaration table for Modbus requests, named "Modbus_RequestsTables". Each element in this table is defined by 10 fields separated by semicolons.

index	Index of the request, from 1 to N
name	Designation of the Modbus request
readFunctionCode	Function code and sub code for reading 0 to 65535
writeFunctionCode	Function code and sub code for writing 0 to 65535
startRegister	Address of the first Modbus register 0 to 65535
size	Size of the response, excluding headers and CRC (in bytes) 1 to 250
enableReading	Activation of read requests: 0 => enabled in instantaneous mode (parameters) 1 => enabled in polling mode (measurements) 2 => disabled
enableWritting	Activation of write requests 0 => enabled in read before write mode 1 => enabled in simple write mode 2 => disabled
option1	Not used
option2	Not used

- The declaration table for variables, named “Modbus_VariablesTables”.
  Each element in this table is defined by 12 fields separated by semicolons.

index	Index of the variable, from 1 to N
indexRequest	Designation of the Modbus request, defined in the table Modbus_RequestsTables.
name	Designation of the Modbus variable
type	Type of variable 2 => Byte. 13 => Word (method LSB first). 14 => Double word (method LSB first). 15 => Timestamp. 16 => Memory buffer.
signed	Sign of the variable 1 => signed 2 => unsigned
position	Position of the variable in the packet 1 to N
option1	Not used
option2	Not used
coeff-A	Scaling coefficient A for the variable (Ax +B)
coeff-B	Scaling coefficient B for the variable (Ax +B)
unit	Units of the variable
action	Processing method for the variable: *0*: variable not collected. *1*: variable treated as parameter. *2*: minimum and maximum collected, and average calculated. *4*: instantaneous value.

Examples:

Details of the declaration of the “Monitoring Standard” request:

1;Monitoring Standard;26368;0;0;220;1;2

- 2: Write disabled
- 1: Activated in polling mode
- 220: Length of response packet
- 0: Start address of the first register
- 0: Function code in decimal of the Modbus write request
- 26368: Function code in decimal of the Modbus read request
- Monitoring Standard: Packet designation
- 1: Index of Modbus request

Details of the declaration of the variable “CurrentMean-PVString1”:

3;1;CurrentMean-PVString1;13;2;7;;;1;0;mA;4

- 4: Instantaneous variable
- mA: Units of the variable
- 0: Coefficient B
- 1: Coefficient A
- 7: Offset of variable within response
- 2: Unsigned variable
- 13: Word type variable
- CurrentMean-PVString1: designation of variable
- 1: Index of the Modbus request
- 3: Index of the variable

Default definition file for GreenString:

```
Modbus_RequestsTables={
# Description of fields
#
reqIndex;reqName;reqReadFctCode;reqWriteFctCode;reqStartRegister;reqNbRegisters;reqEnableReading;reqEnableWritting;reqOption1;reqOption2
#1;Monitoring Standard;26368;0;0;228;1;2
#2;Monitoring Extended;26370;0;0;201;1;2
3;Monitoring Lite;26369;0;0;130;1;2
4;System Configuration;25857;25856;0;64;0;0
5;Device ID;25858;0;0;12;0;2
```

```
6;Remote Control Switch OFF String;0;26112;0;9;2;1
7;Remote Control Switch ON String;0;26113;0;9;2;1
8;Remote Control Switch OFF String RL1;0;26114;0;9;2;1
9;Remote Control Switch ON String RL1;0;26115;0;9;2;1
10;Remote Control Switch OFF String RL2;0;26116;0;9;2;1
11;Remote Control Switch ON String RL2;0;26117;0;9;2;1
12;Remote Control Switch OFF String RL3;0;26118;0;9;2;1
13;Remote Control Switch ON String RL3;0;26119;0;9;2;1
14;System Control System Reset;0;26625;0;1;2;1
15;System Control Device Reset;0;26626;0;8;2;1}

Modbus_VariablesTables={
# Description of fields
#
varIndex;varReqIndex;varName;varType;varSigned;varPosition;varOption1;varOption2;varCoeffA;var
CoeffB,varUnit,varAction;
1;3;TimeStamp;15;2;0;;;1;0;;4
2;3;AverageDCCurrent;13;2;5;;;1;0;mA;2
3;3;AverageDCVoltage;13;2;7;;;0.1;0;V;2
4;3;Current-PVString1;13;1;9;;;1;0;mA;2
5;3;Current-PVString2;13;1;11;;;1;0;mA;2
6;3;Current-PVString3;13;1;13;;;1;0;mA;2
7;3;Current-PVString4;13;1;15;;;1;0;mA;2
8;3;Current-PVString5;13;1;17;;;1;0;mA;2
9;3;Current-PVString6;13;1;19;;;1;0;mA;2
10;3;Current-PVString7;13;1;21;;;1;0;mA;2
11;3;Current-PVString8;13;1;23;;;1;0;mA;2
12;3;Current-PVString9;13;1;25;;;1;0;mA;2
13;3;Current-PVString10;13;1;27;;;1;0;mA;2
14;3;Current-PVString11;13;1;29;;;1;0;mA;2
15;3;Current-PVString12;13;1;31;;;1;0;mA;2
16;3;Current-PVString13;13;1;33;;;1;0;mA;2
17;3;Current-PVString14;13;1;35;;;1;0;mA;2
18;3;Current-PVString15;13;1;37;;;1;0;mA;2
19;3;Current-PVString16;13;1;39;;;1;0;mA;2
20;3;Current-PVString17;13;1;41;;;1;0;mA;2
21;3;Current-PVString18;13;1;43;;;1;0;mA;2
22;3;Current-PVString19;13;1;45;;;1;0;mA;2
23;3;Current-PVString20;13;1;47;;;1;0;mA;2
24;3;Current-PVString21;13;1;49;;;1;0;mA;2
25;3;Current-PVString22;13;1;51;;;1;0;mA;2
26;3;Current-PVString23;13;1;53;;;1;0;mA;2
27;3;Current-PVString24;13;1;55;;;1;0;mA;2
28;3;Current-PVString25;13;1;57;;;1;0;mA;2
29;3;Current-PVString26;13;1;59;;;1;0;mA;2
30;3;Current-PVString27;13;1;61;;;1;0;mA;2
31;3;Current-PVString28;13;1;63;;;1;0;mA;2
32;3;Current-PVString29;13;1;65;;;1;0;mA;2
33;3;Current-PVString30;13;1;67;;;1;0;mA;2
34;3;OutputVoltage-Slave0;13;1;69;;;0.1;0;V;2
35;3;OutputVoltage-Slave1;13;1;71;;;0.1;0;V;2
36;3;OutputVoltage-Slave2;13;1;73;;;0.1;0;V;2
37;3;OutputVoltage-Slave3;13;1;75;;;0.1;0;V;2
```

```
38;3;OutputVoltage-Slave4;13;1;77;;;0.1;0;V;2
39;3;PCB-Temperature-Slave0;13;1;79;;;0.1;0;°C;2
40;3;PCB-Temperature-Slave1;13;1;81;;;0.1;0;°C;2
41;3;PCB-Temperature-Slave2;13;1;83;;;0.1;0;°C;2
42;3;PCB-Temperature-Slave3;13;1;85;;;0.1;0;°C;2
43;3;PCB-Temperature-Slave4;13;1;87;;;0.1;0;°C;2
44;3;PCB-Temperature-Comm+;13;1;89;;;0.1;0;°C;2
45;3;ExternalSensor1;13;1;91;;;0.1;0;;2
46;3;ExternalSensor2;13;1;93;;;0.1;0;;2
47;3;ExternalSensor3;13;1;95;;;0.1;0;;2
48;3;ExternalSensor4;13;1;97;;;0.1;0;;2
49;3;ExternalSensor5;13;1;99;;;0.1;0;;2
50;3;ExternalSensor6;13;1;101;;;0.1;0;;2
51;3;ExternalSensor7;13;1;103;;;0.1;0;;2
52;3;ExternalSensor8;13;1;105;;;0.1;0;;2
53;3;DifferentialCurrentStatusPerString;14;2;107;;;0.1;0;;4
54;3;ReverseCurrentStatusPerString;14;2;111;;;0.1;0;;4
55;3;ZeroCurrentStatusPerString;14;2;115;;;0.1;0;;4
56;3;StringBoxStatus;2;2;119;;;0.1;0;;8
57;3;SlaveCardStatus;2;2;120;;;0.1;0;;8
58;3;DataLogingStatus;2;2;121;;;0.1;0;;8
59;3;SoftwareStatus;13;2;122;;;0.1;0;;8
60;3;IntercomStatistics;13;2;124;;;0.1;0;;4
61;3;RS485Statistics;13;2;126;;;0.1;0;;4
62;4;MaskOfConnectedPVString;14;2;0;;;1;0;;1
63;4;MaskOfPolarityPVString;14;2;4;;;1;0;;1
64;4;StringBoxHardwareConfiguration;2;2;8;;;1;0;;1
65;4;ExternalSensorConfiguration;13;2;9;;;1;0;;1
66;4;AveragingWindowSize;13;2;11;;;1;0;;1
67;4;RecordingPeriod;13;2;13;;;1;0;;1
68;4;TimeStampingMode;2;2;15;;;1;0;;1
69;4;SystemOptions;2;2;16;;;1;0;;1
70;4;WatchdogTimeout;2;2;17;;;1;0;;1
71;4;AlgorithmParameters0;2;2;18;;;1;0;;1
72;4;AlgorithmParameters1;2;2;19;;;1;0;;1
73;4;AlgorithmParameters2;2;2;20;;;1;0;;1
74;4;AlgorithmParameters3;2;2;21;;;1;0;;1
75;4;AlgorithmParameters4;2;2;22;;;1;0;;1
76;4;AlgorithmParameters5;2;2;23;;;1;0;;1
77;4;DiffentialCurrentPair1;13;2;24;;;1;0;;1
78;4;DiffentialCurrentPair2;13;2;26;;;1;0;;1
79;4;DiffentialCurrentPair3;13;2;28;;;1;0;;1
80;4;DiffentialCurrentPair4;13;2;30;;;1;0;;1
81;4;DiffentialCurrentPair5;13;2;32;;;1;0;;1
82;4;DiffentialCurrentPair6;13;2;34;;;1;0;;1
83;4;DiffentialCurrentPair7;13;2;36;;;1;0;;1
84;4;DiffentialCurrentPair8;13;2;38;;;1;0;;1
85;4;DiffentialCurrentPair9;13;2;40;;;1;0;;1
86;4;DiffentialCurrentPair10;13;2;42;;;1;0;;1
87;4;DiffentialCurrentPair11;13;2;44;;;1;0;;1
88;4;DiffentialCurrentPair12;13;2;46;;;1;0;;1

89;4;DiffentialCurrentPair13;13;2;48;;;1;0;;1
```

```
90;4;DiffentialCurrentPair14;13;2;50;;;1;0;;1
91;4;DiffentialCurrentPair15;13;2;52;;;1;0;;1
92;4;DiffentialCurrentPair16;13;2;54;;;1;0;;1
93;5;SerialNumberHigh;14;2;0;;;1;0;;1
94;5;SerialNumberLow;14;2;4;;;1;0;;1
95;5;FirmwareVersion;14;2;8;;;1;0;;1
96;8;SwitchOffStringRL1;16;2;0;;;1;0;;1
97;9;SwitchOnStringRL1;16;2;0;;;1;0;;1
98;10;SwitchOffStringRL2;16;2;0;;;1;0;;1
99;11;SwitchOnStringRL2;16;2;0;;;1;0;;1
100;12;SwitchOffStringRL3;16;2;0;;;1;0;;1
101;13;SwitchOnStringRL3;16;2;0;;;1;0;;1
102;14;SystemReset;16;2;0;;;1;0;;1
103;15;DeviceReset;16;2;0;;;1;0;;1
}
```

5 GreenString functions supported

The GreenString unit provides various functions. Not all of these are supported by the gateway.
Only the set of functions described below is supported:

Set System Configuration	Update parameters
Get System Configuration	Retrieve parameters
Device ID	Retrieve identifier
Remote Switch OFF string RL1	Deactivation of “PV string” remotely
Remote Switch ON string RL1	Activation of “PV string” remotely
Remote Switch OFF string RL2	Deactivation of “PV string” remotely
Remote Switch ON string RL2	Activation of “PV string” remotely
Remote Switch OFF string RL3	Deactivation of “PV string” remotely
Remote Switch ON string RL3	Activation of “PV string” remotely
System Reset	Hardware reset
Device Reset	Erase data logging memory buffer
Standard Monitoring	Retrieval of measurement variables in standard mode
Lite Monitoring	Retrieval of measurement variables in Lite mode
Extended Monitoring	Retrieval of measurement variables in Extended mode

6 Parameter files

The parameters of the Modbus devices present on the bus are sent to the FTP server in two cases:

- During the installation phase triggered by pressing the push-button on the gateway;
- Via a command file of the type GATEWAY: Send inverter parameters (see section *Managing command files* in the WebdynSun User's Manual).

Filename format:
*IDSite*_MODBUS_P_*AAMMJJ_hhmmss*.csv.gz
Where: *IDSite* : gateway identifier.

For each Modbus device, this file details the variables defined as parameters in the file *IDSite*_MODBUS_*famille*.ini.

> To minimise the sizes of transmitted files, the parameter file is compressed in .gz format.

The file format is as follows: (fields in green are optional data that can be enabled or disabled in *IDSite*_daq.ini (see section *7.3. Declaring and configuring inverters* in the User's Manual)).

31 => Number of parameters retrieved (optional)

2 => Modbus address of the device

001=> Number of the device in the configuration (optional)

MODBUS_GREENSTRING.ini => name of the definition file for the device

198;..;208; => Index of the parameters retrieved.

```
ADDRMODBUS;2;001;
TypeMODBUS;MODBUS_GREENSTRING.ini;
31;198;..;208;209;210;211;212;213;214;215;216;217;218;219;220;221;222;223;224;225;226;227;228;
06/10/11-12:03:49;4294967295;…;0;0;0;0;0;0;0;0;0;600;0;600;0;85;0;638066705;4;1;128;0;
```

4294967295;…;0;0;0 => List of parameters retrieved

06/10/11-12:03:49 => Timestamp in GMT when parameters were retrieved

In this example, the ellipses (…) correspond to the variables beyond 8 for ease of display. The column heading "31;198;199;200;201;202;203;204;205;206;207;208;209…" is added if the option (DAQ_HeaderOption) is enabled in *IDSite*_daq.ini (see section *7.3. Declaring and configuring inverters* in the User's Manual).

7 Data files

Modbus data are constantly collected by the gateway, then saved periodically to a collection file according to a period defined in the configuration file daq.ini.

This file is compressed, then uploaded to an FTP server every time the gateway connects to it.

The GreenString module locally calculates average, minimum, maximum and standard deviation values for most of its variables. It is therefore unnecessary to perform these calculations on the gateway. However, this mode of operating imposes the need for timestamping on the GreenString module. This necessitates synchronisation between the gateway and the GreenString modules declared. Three methods are available on the GreenString to carry out this synchronisation: synchronous mode, asynchronous mode, and hybrid mode. With a view to maintaining operational compatibility with other Modbus devices handled by the gateway, the method for timestamping data chosen for data collection is the asynchronous method.

The data therefore have two timestamps per collection:
- the timestamp applied by the gateway when recording the data
- the timestamp applied by the GreenString module during internal calculations.

As each GreenString module is autonomous, there may be discrepancies between the timestamps applied by the gateway and that of the modules. These discrepancies are related to the accuracy of the hardware.
Gateway deviation is controlled by periodically resynchronising the gateway with NTP. GreenString module deviation is controlled by resynchronising with the gateway after each restart of a GreenString module.

Filename format:
*IDSite*_MODBUS_*AAMMJJ*_*hhmmss*.csv.gz
Where: *IDSite*: gateway identifier.

To minimise the sizes of transmitted files, the data file is compressed in .gz format

All the data to be collected are described in the definition files for each Modbus device.

Data collection occurs sequentially. Consequently, the collection period depends on the number of devices declared on the bus and on the number of request sent per device.
The minimum time between each request is 250 ms.

The file format is as follows: (fields in green are optional data):

197 => Number of variables captured (optional)

2 => Modbus address of the device

001=> Number of the device in the configuration (optional)

MODBUS_GREENSTRING.ini => name of the definition file for the device

```
ADDRMODBUS;2;001;
TypeMODBUS;MODBUS_GREENSTRING.ini;
197;1;2;3;4;5;6;7;8;9;10;…;196;197;
06/10/11-12:20:05;2011/10/06-12:20:00.000;3543;4043;4043;4043;4043;…;2977;2977;
06/10/11-12:30:05;2011/10/06-12:30:00.000;3495;4089;4089;4089;4089;…;1985;1985;
```

2011/10/06-12:30:00.000;…;1985;1985=> List of the variables captured at the 12:30 collection time

06/10/11-12:30:05=> Timestamp in GMT when the variables were captured

8 Alarm files

On changing status, some Modbus variables can generate an alarm report.

This alarm is transmitted to the remote server in the form of a file on the FTP server (*IDSite*_AL_*YYMMDD_hhmmss*.csv.gz).
The data stored is also uploaded at the same time as an alarm.

Alarm of the MODBUS type:
This type of alarm is uploaded on a change in the status of a variable defined as an alarm in the associated definition file.

Date - Time	Type	Field 1	Field 2	Field 3	Field 4

Avec :

Date–Time	Timestamp when alarm triggered, in format YY/MM/DD-hh:mm:ss
Type	MODBUS: Alarm type is Modbus
Field 1	*IDsite*_MODBUS_*type*.ini: Name of reference definition file
Field 2	Name of the triggering device
Field 3	1 to N: Index of the triggering data item in the associated definition file.
Field 4	Status of the trigger item, in decimal

Please see the manufacturer's documentation for details of every variable.

Example:

```
28/09/11-12:48:28;MODBUS;MODBUS_GREENSTRING.ini;GreenString1;98;63
28/09/11-12:48:38;MODBUS;MODBUS_GREENSTRING.ini; GreenString1;56;0
```

The variable with index 56 in the definition file "MODBUS_GREENSTRING.ini" of the device GreenString1 changed its value to 63 at 12:48:28. Then the same variable switched to 0 at 12:48:38. This variable corresponds to the variable "StringBoxStatus".
According to the GreenString module specification, the value 63 corresponds to:

- DC tension surge protection U/S
- Main DC circuit breaker triggered
- Secondary DC circuit breaker triggered
- PCB overheating detected
- Theft alarm triggered
- Differential fault.

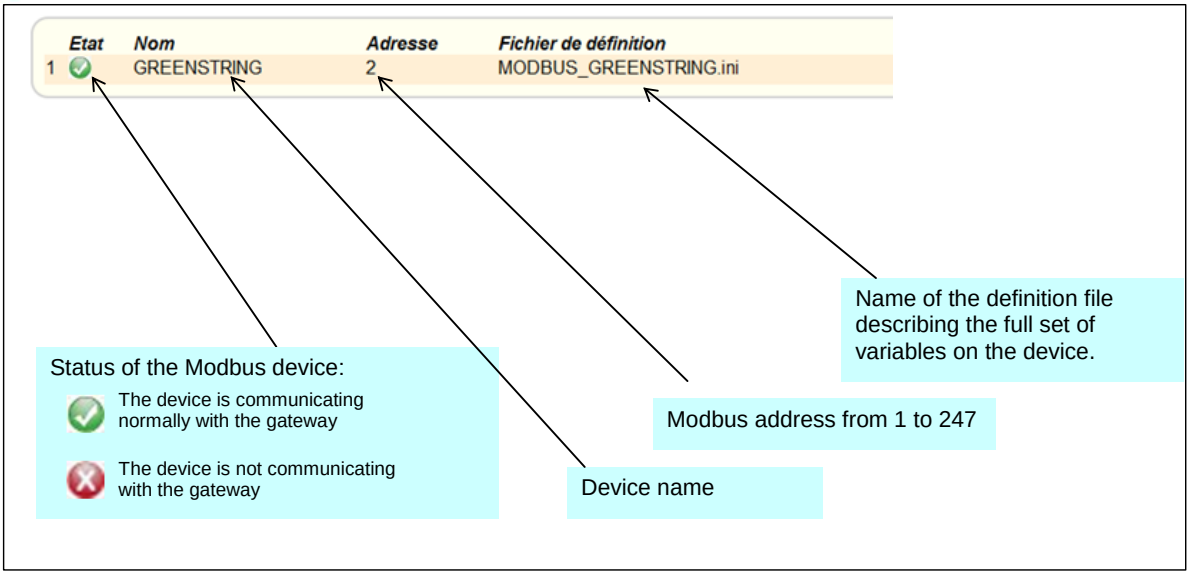
The remaining information encoding is defined below:

Bit	Description
7	*Reserved*
5:6	Indicates a configuration error: 0: No configuration error 1: Wrong differential current pairs 2: Not all PV string inputs have been successfully detected (e.g. slave card missing) 3: Configuration file error
4	If set, theft-alarm is on
3	If set, PCB over-temperature has been detected
2	'Secondary' DC breaker status: '0' = OFF, '1' = ON (in case the same string box is connected to two different inverters)
1	'Main' DC breaker status: '0' = OFF, '1' = ON
0	DC surge protection. Set to '1' means "end-of-life" indication

Table 24: String Box status encoding

9 Checking that a GreenString module is operating correctly

Once declared to the FTP server and the gateway, each GreenString module can be checked locally through the built-in Web server on the gateway.
This is carried out via the menu "Control/Modbus":



10 Command files

The gateway can send commands to Modbus devices. This is done via the command file *IDsite*_CMD.ini provided on the FTP server in the directory CMD. This file is downloaded by the gateway on every connection to the server.

In the case of a GreenString module, this enables configuration of some parameters and control of the electrical isolation of the photovoltaic panels.

Each line of the command file corresponds to one command.
For a Modbus module, this command must comply with the format below:

Index;Type;Parameter1;Parameter2;Parameter3

Where:

Index	1 to N: Unique Identifier enabling the command to be identified
Type	MODBUS: Command of the MODBUS type
Parameter 1	Index of the Modbus device to be configured. This index corresponds to the index used to declare the device in the file daq.ini. (0 to N)
Parameter 2	Index of the Modbus variable to be configured. This index corresponds to the variable identification index present in the definition file for the device to be configured.
Parameter 3	Decimal or ASCII value of the parameter.

10.1 Taking a photovoltaic panel online or offline

Remote control of the online status of photovoltaic panels must be used with care, as it acts directly on the power up or down status of the installation.

Taking panels offline or online is done using the Modbus Remote Control requests “SwitchOffString” and “SwitchOnString” described in the definition file for the module (index values 96 to 101):

```
96;8;SwitchOffStringRL1;16;2;0;;;1;0;;1
97;9;SwitchOnStringRL1;16;2;0;;;1;0;;1
98;10;SwitchOffStringRL2;16;2;0;;;1;0;;1
99;11;SwitchOnStringRL2;16;2;0;;;1;0;;1
100;12;SwitchOffStringRL3;16;2;0;;;1;0;;1
101;13;SwitchOnStringRL3;16;2;0;;;1;0;;1
```

These commands are controlled by a 64-bit password.

Taking a String offline: SwitchOffString RL1

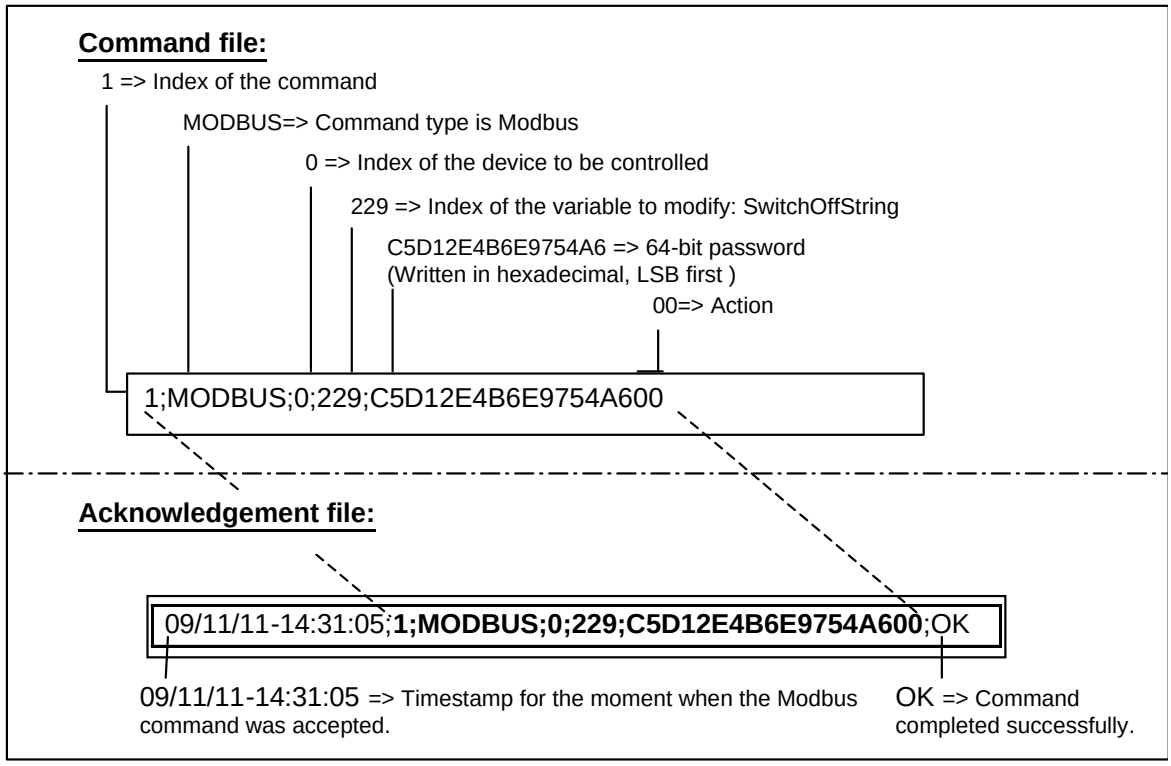
```
1;MODBUS;0;96;Password + 00
```

Bringing a String online: SwitchOnString RL1

```
1;MODBUS;0;97;Password + 00
```

Each command is acknowledged in an acknowledgement file named *IDsite*_ACK_*DateTime*.ini according to the format described below:

Example: Switch OFF for all panels on the GreenString module with index 0.



10.2 Configuring GreenString modules

A GreenString module is configured in the factory to handle the maximum number of photovoltaic panels. It is therefore necessary to reconfigure it to match the actual installation, in order to avoid alarms being raised unexpectedly on devices that are not connected. This is done via the command files of the WebdynSun gateway.

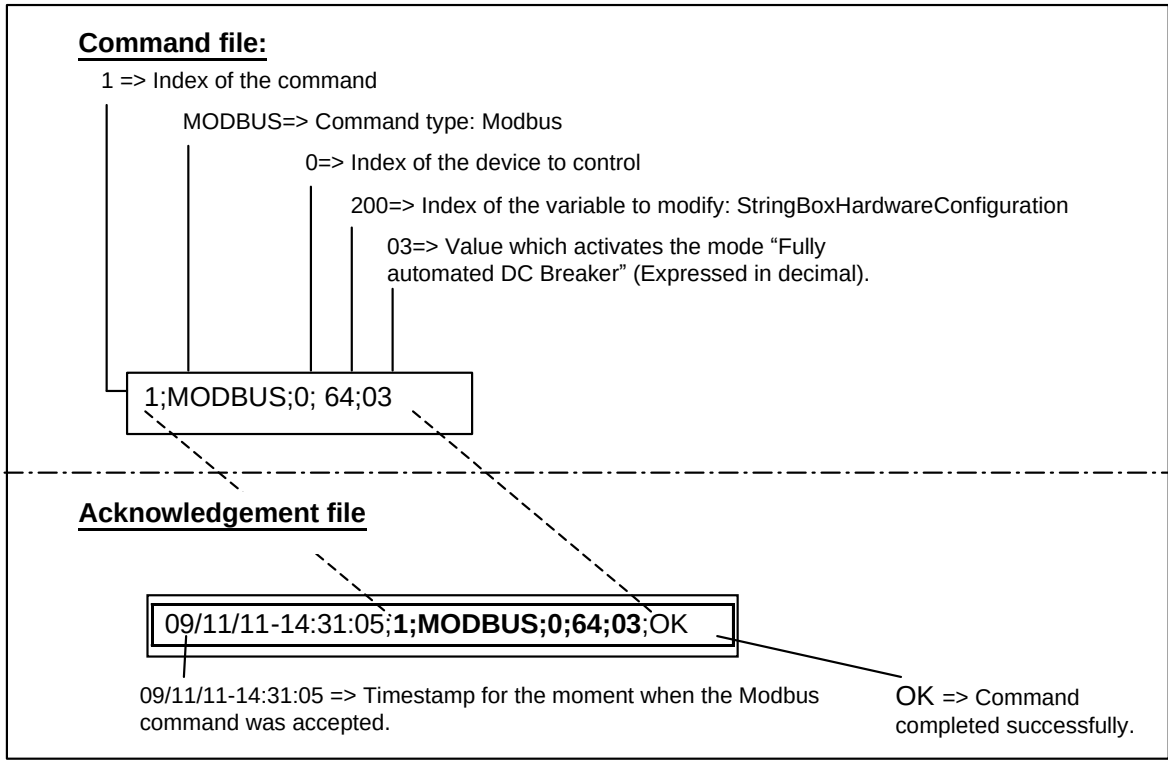
The list of supported GreenString parameters described in the definition file is:

```
62;4;MaskOfConnectedPVString;14;2;0;;;1;0;;1
63;4;MaskOfPolarityPVString;14;2;4;;;1;0;;1
64;4;StringBoxHardwareConfiguration;2;2;8;;;1;0;;1
65;4;ExternalSensorConfiguration;13;2;9;;;1;0;;1
66;4;AveragingWindowSize;13;2;11;;;1;0;;1
67;4;RecordingPeriod;13;2;13;;;1;0;;1
68;4;TimeStampingMode;2;2;15;;;1;0;;1
69;4;SystemOptions;2;2;16;;;1;0;;1
70;4;WatchdogTimeout;2;2;17;;;1;0;;1
71;4;AlgorithmParameters0;2;2;18;;;1;0;;1
72;4;AlgorithmParameters1;2;2;19;;;1;0;;1
73;4;AlgorithmParameters2;2;2;20;;;1;0;;1
74;4;AlgorithmParameters3;2;2;21;;;1;0;;1
75;4;AlgorithmParameters4;2;2;22;;;1;0;;1
76;4;AlgorithmParameters5;2;2;23;;;1;0;;1
77;4;DiffentialCurrentPair1;13;2;24;;;1;0;;1
78;4;DiffentialCurrentPair2;13;2;26;;;1;0;;1
79;4;DiffentialCurrentPair3;13;2;28;;;1;0;;1
80;4;DiffentialCurrentPair4;13;2;30;;;1;0;;1
```

```
81;4;DiffentialCurrentPair5;13;2;32;;;1;0;;1
82;4;DiffentialCurrentPair6;13;2;34;;;1;0;;1
83;4;DiffentialCurrentPair7;13;2;36;;;1;0;;1
84;4;DiffentialCurrentPair8;13;2;38;;;1;0;;1
85;4;DiffentialCurrentPair9;13;2;40;;;1;0;;1
86;4;DiffentialCurrentPair10;13;2;42;;;1;0;;1
87;4;DiffentialCurrentPair11;13;2;44;;;1;0;;1
88;4;DiffentialCurrentPair12;13;2;46;;;1;0;;1
89;4;DiffentialCurrentPair13;13;2;48;;;1;0;;1
90;4;DiffentialCurrentPair14;13;2;50;;;1;0;;1
91;4;DiffentialCurrentPair15;13;2;52;;;1;0;;1
92;4;DiffentialCurrentPair16;13;2;54;;;1;0;;1
```

Each variable is processed individually, while respecting the file format.

Example: Activation of the mode “Fully automated DC Breaker” via the variable “StringBoxHardwareConfiguration” (index 64) of the GreenString module (index 0).



Please see the manufacturer’s documentation for details of every variable.