

**WEBDYN**  
flexitron group

## GATEWAY SPECIFICATION



## WebdynModbus

The monitoring gateway for  
your Modbus equipments

## SOMMAIRE

1	Glossary.....	3
2	About this document .....	4
2.1	Scope .....	4
2.2	Targeted audience.....	4
2.3	Product versions .....	4
2.4	Safety advice .....	4
3	Principles of operation .....	4
	Overview diagram of a comprehensive <i>WebdynModbus</i> monitoring solution.....	5
4	Unit characteristics.....	5
4.1	Technical caractéristiques .....	5
4.2	List of available interfaces .....	6
4.3	Optional accessories .....	6
5	Prerequisites.....	9
5.1	Access to the FTP server .....	9
5.2	Access to the NTP server .....	10
5.3	Connection via GPRS or Ethernet .....	10
6	Gateway configuration .....	11
6.1	Role of the configuration files .....	11
6.2	Initialisation.....	12
6.3	Setting the unit date and time .....	21
6.4	Connection modes and period.....	22
6.5	Acquisition period and time slots .....	26
7	Modbus device management.....	27
7.1	Bus wiring .....	27
7.2	Configuring and declaring Modbus slave devices.....	29
7.3	Structure of a Modbus definition file.....	32
7.4	Checking that Modbus devices are operating correctly .....	36
7.5	Modbus data .....	37
7.6	Modbus alarms .....	40
7.7	Writing Modbus variables via a command file.....	42
8	Updating the unit.....	42
8.1	Updating via the Web server .....	42
8.2	Updating remotely via the FTP server .....	44
9	Using web services .....	44
9.1	Enabling and configuring .....	45
9.2	Format of HTTP requests .....	45
9.3	Examples of Web Services requests .....	47
10	Tools and diagnostics.....	48
10.1	Events journal .....	48
10.2	Modem information .....	50
10.3	Detecting power connection .....	50
10.4	LED indicators .....	51
10.5	Installation button.....	52
10.6	Diagnostic SMSs .....	52
10.7	Debug traces .....	53
10.8	Factory reset procedure.....	54
10.9	Support.....	55

## 1 Glossary

Name	Description
APN	<p>Access Point Name</p> <p>Name of the access point that enables the gateway to connect to the Internet via a mobile link.</p>
ERDF	<p>Électricité Réseau Distribution France, the French National Grid, which defines connection norms for installations and meters, especially smart meters. See TIC.</p>
FTP	<p>File Transfer Protocol</p> <p>Communications protocol used to share data files on a TCP/IP network.</p>
GPRS	<p>General Packet Radio Service</p> <p>Standard for mobile telephone communications derived from the GSM standard and enabling higher data transfer rates. Also known as 2.5G.</p> <p>DL: maximum 86 kbps</p> <p>UL: maximum 43 kbps</p>
GSM	<p>Global System for Mobile Communications</p> <p>The switched network for mobile telephones.</p>
HTTP	<p>HyperText Transfer Protocol</p> <p>Client-server communications protocol developed for the Web.</p>
IP	<p>Internet Protocol</p> <p>Message protocol controlling the addressing and transmission of TCP packets over the network.</p>
DIN Rail	<p>Standardised 35-mm metal rail used in Europe for rack-mounted industrial control equipment</p>
PSTN	<p>Public switched telephone network</p> <p>Switched network for telephone land lines.</p>
TCP	<p>Transmission Control Protocol</p> <p>A connection-oriented protocol for the Internet, which provides data segmentation into packets that are transmitted over the network via the IP protocol. This protocol provides a reliable data transfer service. See also IP.</p>
TCP/IP	<p>Transmission Control Protocol/Internet Protocol</p> <p>The suite of network protocols that provide interconnection services between computers with different hardware architectures and operating systems. TCP/IP includes standards for communication between computers and conventions for network interconnection and for routing.</p>

## 2 About this document

The purpose of this guide is to describe the installation and operation of a *WebdynModbus* gateway.

### 2.1 Scope

The present technical description is valid for *WebdynModbus* gateways from hardware version 1 onwards, and from software version V3.04.17 and up.

### 2.2 Targeted audience

This guide is intended for users of *WebdynModbus* gateways.

### 2.3 Product versions

Version 1.0

### 2.4 Safety advice

It is essential to respect all safety recommendations featured in this guide.

Failure to comply with these recommendations may cause damage to equipment and danger to the health safety of personnel.



#### **Electrical connections**

- All wiring must be carried out by a qualified electrician.
- Prior to installation, all equipment connected to the corresponding communications bus must be disconnected on both sides (DC and AC).



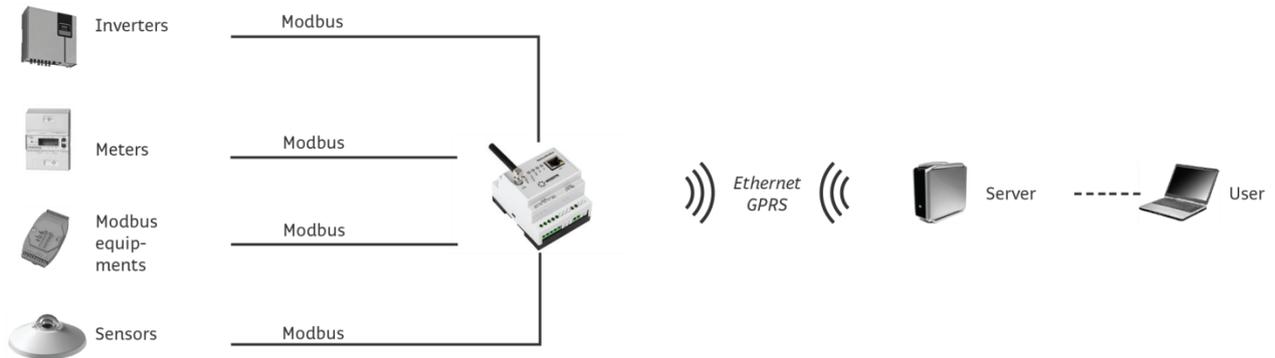
**The WebdynModbus unit is liable to damage from electrostatic discharge (ESD)**

**Avoid all contact with component connectors and terminals**

## 3 Principles of operation

The *WebdynModbus* gateway is the communications hub for your Modbus installation. It continuously collects desired data from Modbus equipments, concentrates this data into files, and sends these files to a remote FTP server. In doing so, this gateway enables you to be constantly informed on the operational state of your installation.

## Overview diagram of a comprehensive WebdynModbus monitoring solution



## 4 Unit characteristics

### 4.1 Technical characteristics

Electrical characteristics	
Input tension	12 / 24 V
Electricity consumption when idle	1.6 W (1)
Electricity consumption when connected via Ethernet	2 W (1)
Maximum peak power during GPRS connection	3 W (1)
<b>GSM/GPRS antenna: Microel EA-247</b>	
Frequency	900/1800 Mhz
Gain	0 dB
Polarisation	Verticale
<b>Memory</b>	
Storage capacity	100 Mo of compressed data
<b>Dimensions</b>	
Size	70mm*86mm*59mm (L,h,p) , RAIL DIN
Environmental conditions	
Operational temperature range	-5°C à +55°C
Storage temperature	-5°C à +45°C

(1) Measurements carried out on the Unit + DIN Rail DR15-24 power supply assembly

## 4.2 List of available interfaces

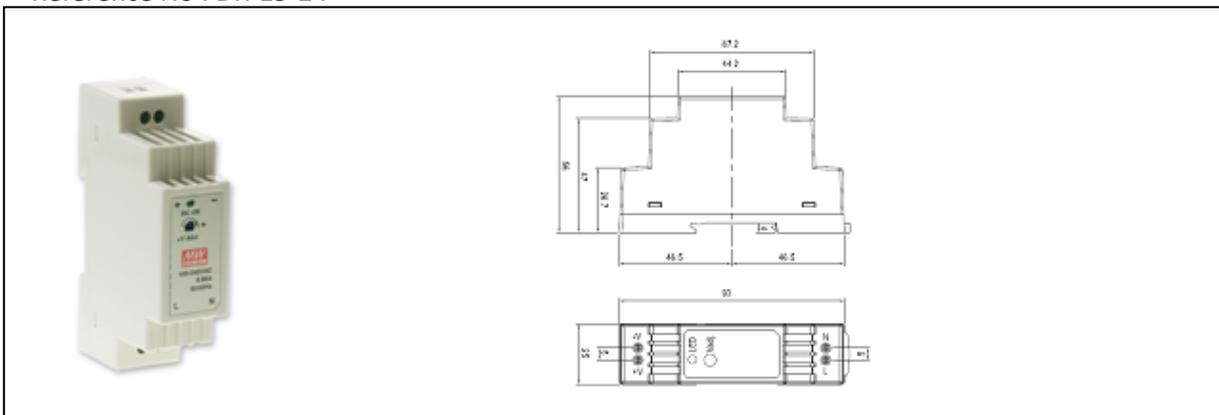
Data source	Interface	Characteristics
Modbus equipments	RS485 2/4 wires, Ethernet	Maximum number 247
Communications channel	Protocol	
Ethernet 10/100Mbits/s	IP services	
GSM/GPRS modem	IP services	
Remote servers	Protocole	
FTP server	FTP	
HTTP server (Web Services option)	HTTP	
NTP server	NTP	

## 4.3 Optional accessories

### 4.3.1 RAIL DIN power supply

Brand : MEANWELL

Reference No : DR-15-24



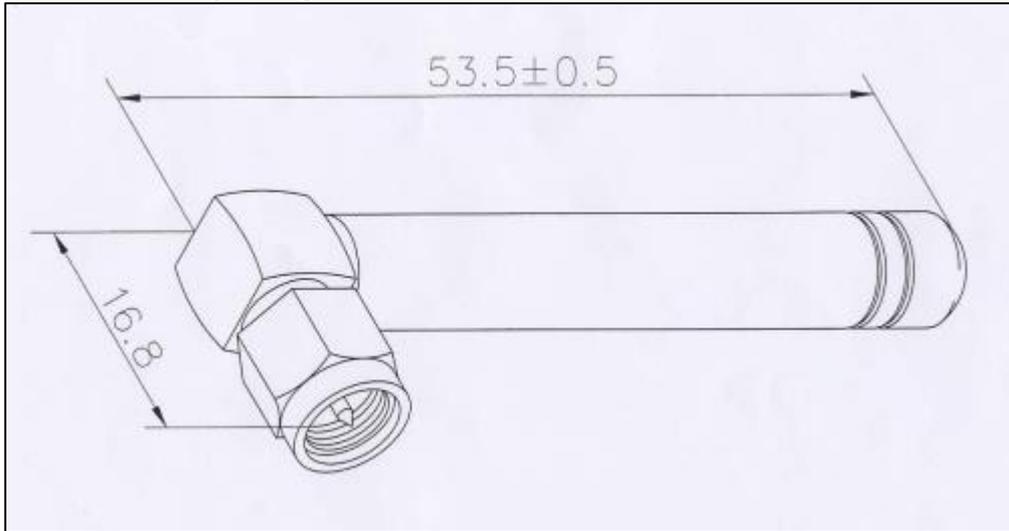
This power supply unit can be mounted next to the gateway on the standard 35-mm DIN RAIL metal rail, no other mounting rail is required.

Characteristics:

OUTPUT	DC VOLTAGE	24V
	RATED CURRENT	0.63A
	CURRENT RANGE	0 ~ 0.63A
	RATED POWER	15.2W
	RIPPLE & NOISE (max.) Note.2	150mVp-p
	VOLTAGE ADJ. RANGE	21.6 ~ 26.4V
	VOLTAGE TOLERANCE Note.3	±1.0%
	LINE REGULATION	±1.0%
	LOAD REGULATION	±1.0%
	SETUP, RISE TIME	1000ms, 50ms/230VAC

		1000ms, 50ms/115VAC at full load
	HOLD UP TIME (Typ.)	70ms/230VAC 16ms/115VAC at full load
INPUT	VOLTAGE RANGE	85 ~ 264VAC    120 ~ 370VDC
	FREQUENCY RANGE	47 ~ 63Hz
	EFFICIENCY (Typ.)	85%
	AC CURRENT (Typ.)	0.88A/115VAC 0.48A/230VAC
	INRUSH CURRENT (Typ.)	COLD START 35A/115VAC 65A/230VAC
PROTECTION	OVERLOAD Note.5	105 ~ 160% rated output power Protection type : Constant current limiting, recovers automatically after fault condition is removed
	OVER VOLTAGE	27.6 ~ 32.4V
ENVIRONMENT	WORKING TEMP.	-20°C ~ +60°C (Refer to output load derating curve)
	WORKING HUMIDITY	20 ~ 90% RH non-condensing
	STORAGE TEMP., HUMIDITY	-40°C ~ +85°C, 10 ~ 95% RH
	TEMP. COEFFICIENT	±0.03%/ (0 ~ 50) c°
	VIBRATION	10 ~ 500Hz, 2G 10min./1cycle, period for 60min. each along X, Y, Z axes; Mounting: Compliance to IEC60068-2-6
SAFETY & EMC	SAFETY STANDARDS	UL60950-1, TUV EN60950-1 approved, design refer to EN50178
	WITHSTAND VOLTAGE	I/P-O/P:3KVAC
	ISOLATION RESISTANCE	I/P-O/P:100M Ohms / 500VDC / 25 / 70% RH
	EMI CONDUCTION & RADIATION	Compliance to EN55011, EN55022 (CISPR22), EN61204-3 Class B
	HARMONIC CURRENT	Compliance to EN61000-3-2,-3
	EMS IMMUNITY	Compliance to EN61000-4-2, 3, 4, 5, 6, 8, 11, ENV50204, EN55024, EN61000-6-2, EN61204-3, heavy industry level, criteria A
OTHERS	MTBF	1172.3K hrs min.    MIL-HDBK-217F (25 )
	DIMENSION	25*93*56mm (W*H*D)

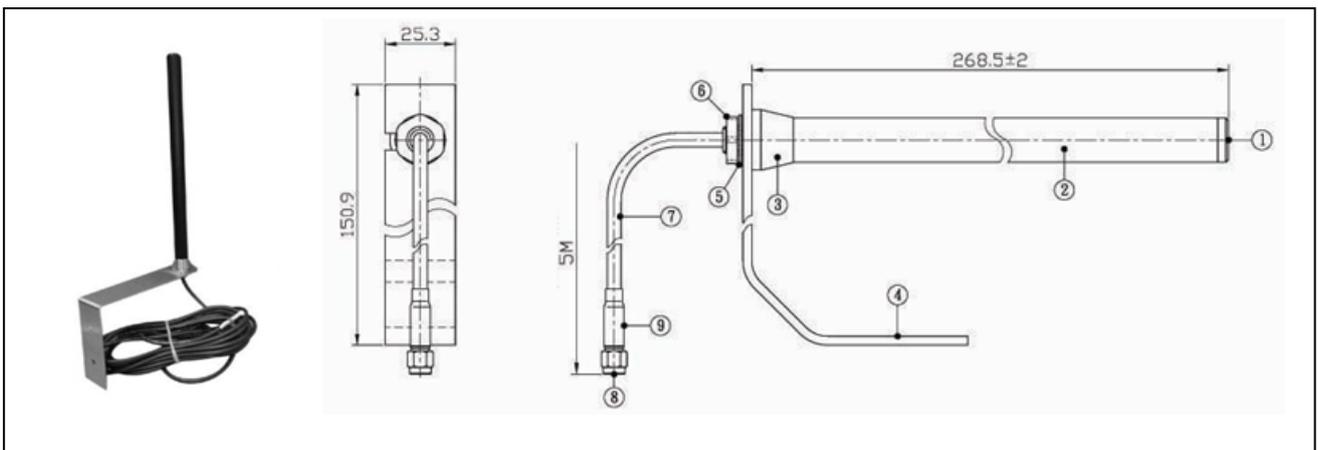
#### 4.3.2 GSM/GPRS right-angled stub antenna



Characteristics:

Frequency range	900-1800MHz
-----------------	-------------

#### 4.3.3 GSM/GPRS externally mounted antenna (optional accessory)



Characteristics:

Cable	RG-58
Cable length:	disponible en 5,10 ou 20 mètres
Frequency range	824-960MHz 1710-1990MHz
Options	Mounting bracket

## 5 Prerequisites

It is crucial for the WebdynModBus to have an internet access, either Ethernet or GPRS, in order to enable file transfers between the gateway and a configured remote server, to enable the synchronization of the gateway's internal clock with the one's of general purpose systems.

As mentioned above, the gateway exchanges files, of different types containing categorized information, with an off-board FTP server.

- Alarm, data, log, and debug files must be uploaded by the gateway to the remote server for user access to this information.
- Update and command files can be downloaded by the gateway from the remote server to take into account new functionalities and bug fixes comprised in new firmware versions and for the user to transmit commands to the gateway.
- Configuration and definition files can be uploaded and downloaded by the gateway to/from the remote server to give read and/or write access to the gateway's configuration.

In addition, the factory default time and date setting of the gateway is 1<sup>st</sup> January 1970 at 00:00, known as EPOCH time. In order for the gateway's time to be synchronized with the time of other systems it must communicate with an NTP server from which it receives UTC time.

Precisions on these prerequisites are listed below.

### 5.1 Access to the FTP server

It is essential that the remote FTP server is configured the following way:

- Read/Write/Rename access authorised;
- Passive mode enabled;
- Port 21 (by default);
- Short handshake message;
- Login and Password must be less than 30 characters.

As the *WebdynModbus* gateway does not create any directories on the remote FTP server these must be manually created by the user or the person in charge of the Information System used by the gateway. Below is the list of directories which MUST be present on the remote FTP server. These directories don't however have to be at the root of the FTP server, a parent directory can be created as long as the path of the directories below is manually updated in the gateway's configuration::

- **/ALARM:** Contains the alarm files. **/BIN:** contains the binary files of the gateway for update purposes.
- **/CMD:** Contains the command and acknowledgement files.
- **/CONFIG:** Contains the configuration files for the gateway.
- **/DATA:** Directory for upload of data files. This directory contains four subdirectories:
  - **/MODBUS:** Modbus data.
- **/DEF:** Contains the definition files for the modules and sensors to be controlled. This directory contains the subdirectory:
  - **/MODBUS:** Definition files for the monitored Modbus modules.
- **/LOG:** Contains the gateway's system logs (*IDSite\_DATE.log*) and the debug logs, disabled by default (*IDSite\_DATE\_debug.log*).

## 5.2 Access to the NTP server

To set its date and time correctly to UTC time, the *WebdynModbus* synchronises itself with an NTP server before each connection to the FTP server. By default, the gateway is configured to get the time from the NTP server “pool.ntp.org”. As this NTP server is accessible only via Internet, the gateway must have operational Internet access (UDP port 123 must be open for outgoing traffic) for synchronisation to occur.

## 5.3 Connection via GPRS or Ethernet

Access to the FTP and NTP servers can be done either via an Ethernet connection or a GPRS connection.

### 5.3.1 Ethernet connection

For connection to Internet via Ethernet, the following configuration parameters must be correctly informed to the gateway:

- IP address of the *WebdynModbus* on the local network (factory IP address: 192.168.1.12);
- Subnet mask;
- IP address of the router or ADSL modem;
- IP address of the DNS server.

### 5.3.2 GPRS connexion

For connection via GPRS, it is essential to insert an activated SIM card with a DATA option in the gateway, and to inform it with the following information correctly:

- APN (Access Point Name): Name of the GPRS access point. This depends on the operator and the type of subscription;
- User name and password for connection to the APN

### 5.3.3 Management of the SIM card PIN code

The *WebdynModbus* handles the SIM card’s PIN code automatically, the PIN code cannot be informed to the gateway. The SIM card must therefore either have no PIN code or have a PIN code set to 0000 when it is first inserted.

The management of the SIM card's PIN code is as follows :

- If the PIN code is deactivated: GPRS communication is operational for the gateway.
- If the PIN code is activated and equal to "0000": when the gateway is fired up for the first time, a new PIN code is attributed to the SIM card. This PIN code is defined on the basis of the ICCID (Integrated Circuit Card Identification) number of the SIM card installed. It is calculated using a proprietary Webdyn algorithm. This feature allows fraudulent use of the SIM card to be prevented, while providing ease of use.
- If the PIN code is activated and different to "0000": GPRS communication is not operational for the gateway.

In addition, this same SIM card can be reused in another *WebdynModBus*, *WebdynSun*, *WebdynPulse*, or *WebdynTIC* unit without any additional configuration. If the card has a PIN code that is activated but is neither "0000" nor the code attributed by the *WebdynModBus* gateway, communications (including SMSs) will not be operational.



**Never insert a SIM card with an activated PIN code that is neither "0000" nor a value attributed by a *WebdynModBus*, *WebdynSun*, *WebdynPulse*, or *WebdynTIC*. If you do so, the SIM card will be blocked and the user will have to unblock it by entering the PUK code manually.**

## 6 Gateway configuration

Configuration of the *WebdynModbus* gateway should be carried out in several stages. The first stage, initialisation, consists of configuring the *WebdynModbus* gateway so that it can connect to the FTP server. This can be either via the web interface, accessible by connecting the gateway to a PC or local network via Ethernet, or via SMS. The second stage consists of configuring the gateway remotely via the previously configured FTP server. During this stage, it is possible to modify the whole set of configuration parameters via the files available on the server, including the ones concerning ModBus equipments. For these equipments it is necessary to modify the `prefix_daq.ini` file and to create a definition file for the addressing of the variables.

### 6.1 Role of the configuration files

Except for the locally, and remotely via SMS, accessible parameters enabling connection to the remote FTP server, the totality of the configuration parameters are accessible via configuration files available on the said server.

Each configuration file is prefixed with a unique identifier: the "prefixID". This prefix enables the configuration for each gateway connected to a given server to be customised. Two categories of files are necessary for configuring the unit: the configuration files and the definition files.



The *WebdynModbus* gateway does not overwrite the configuration and definition files available on the server. Care must be taken to maintain consistency between the configuration of the gateway and the files on the server. Where the configuration is modified locally, it is advisable to delete the files from the server so that the gateway will recreate them. Conversely, if one of the files on the server is modified, the gateway detects this and will download it. The standard configuration of the gateway is thus overwritten.

### 6.1.1 Configuration files

The *WebdynModbus* has three configuration files:

- *prefixID\_config.ini*: this contains the general parameters of the WebdynModBus.
- *prefixID\_daq.ini*: this contains the parameters required for data acquisition.
- *prefixID\_var.ini*: this contains the information for scheduling connection and data acquisition times.

These 3 files are contained in the configuration directory on the FTP server. By default, this directory is “/CONFIG”, but can be modified by means of the variable “FTP\_DirConfig” in the file *prefixID\_config.ini*.

If the gateway does not detect these files on the FTP server, it creates them from its current configuration. Furthermore, at every connection to the server, the gateway checks the modification dates and sizes of the files to detect any modification(s) on a given file. Should a modification be detected, the gateway downloads the new file.

The details of each parameter in the configuration files will be explained as required in the remainder of this documentation.

### 6.1.2 Definition files

The role of the definition file is to inform the gateway on the data which it will be reading/writing on the Modbus slave as well as how the gateway will access this data.

The link between the definition files and the gateway is provided via the configuration file *prefixID\_daq.ini*. The format of the definition files, including a section defining the requests to be sent and another describing the variables which are to be written/read is described in chapter 7.3 « Structure of a Modbus definition file ».

## 6.2 Initialisation

Initial configuration of the unit is necessary to enable connection to the remote FTP server. This configuration can be carried out either via the on-board Web Server or by SMS if a functional SIM card has been inserted in the gateway.

Local configuration of the *WebdynModbus* affects only the variables in the configuration file: *prefixID\_config.ini*.

Here is a list of the variables accessible via the local web interface (http) and/or via SMS commands:

Variable	Definition	Default value	HTTP	SMS
ID	Gateway identifier (up to 29 characters)	WDxxxxxx where xxx... is the last 6 digits of the MAC address	X	
INV_Type	Inverter protocol used.  (Not used in the case of the <i>WebdynModbus</i> product)	0	X	
LAN_IpAddr	IP address of the gateway (router) on the LAN (local area network) Communication via Ethernet (up to 15 characters)	192.168.1.12	X	
LAN_SubnetMask	LAN subnet mask Communication via Ethernet (up to 15 characters)	255.255.255.0	X	
LAN_Gateway	Address of the gateway (router) on the LAN Communication via Ethernet (up to 15 characters)	0.0.0.0	X	
LAN_DNS	Address of the DNS server on the LAN Communication via Ethernet (up to 15 characters)	0.0.0.0	X	
LAN_DHCP_Enable	Enable/Disable DHCP: For automatic provision of Ethernet IP addresses. 0=Disabled 1=Enabled	0	X	
GPRS_APN	GPRS Access Point Name (APN) Provided by mobile operator (up to 29 characters)	m2minternet	X	X
GPRS_Login	GPRS APN identifier Provided by mobile operator (up to 29 characters)	sfr	X	X
GPRS_Password	GPRS APN password Provided by mobile operator (up to 29 characters)	sfr	X	X

GPRS_PhoneNumber	Phone number for GPRS In France: *99***1# (up to 13 characters)	*99***1#	X	X
WAN_Connection Interface	Remote server connexion interface selected: 0=Ethernet 1=modem (GPRS or PSTN depending on unit version)	1	X	
FTP_Server	Name of remote FTP server (up to 29 characters)		X	X
FTP_Login	Identifier for connexion to remote FTP server (up to 29 characters)		X	X
FTP_Password	Password for connexion to remote FTP server (up to 29 characters)		X	X
FTP_Port	Port used for connexion to remote FTP server	21	X	X
FTP_DirConfig	FTP directory name for gateway configuration files (up to 29 characters)	/CONFIG	X	
FTP_DirDef	FTP directory name for gateway definition files (up to 29 characters)	/DEF	X	
FTP_DirData	FTP directory name for data files (up to 29 characters)	/DATA	X	
FTP_DirLog	FTP directory name for log files (up to 29 characters)	/LOG	X	
FTP_DirBin	FTP directory name for gateway firmware Used for upgrading gateway (up to 29 characters)	/BIN	X	
FTP_DirAlarm	FTP directory name for alarm files (up to 29 characters)	/ALARM	X	
FTP_DirCmd	FTP directory name for command files (up to 29 characters)	/CMD	X	
FTP_Option	Enable/Disable of two-phase data uploading. (File with extension ".tmp" uploaded, then extension ".tmp" deleted after transfer.	0	X	

	0=Disabled 1=Enabled			
WebService_Enable	Enable/Disable of web services: 0= Disabled 1= Enabled	0	X	
WebService_Url	Web Service http:// address (up to 29 characters)		X	
CONFIG_ForceUpload	Authorise the overwriting of the files on the FTP server 0= Disabled 1= Enabled	0	X	
Language	Language chosen for built-in website: fr = French en= English	fr	X	

These variables can be modified at any time on the remote server.



**The server configuration always overrides the local configuration via the web interface. Please ensure that the two configurations are consistent.**



**To ensure consistency between the server and the unit, it is advisable to delete the unit's configuration file *prefixID\_config.ini* from the remote server before any local modification is carried out. Thus, the gateway will upload its new *prefixID\_config.ini* configuration file at the next connection to the FTP server rather than downloading the one present on the remote server. This would cause any modification brought to the configuration via the web interface to be overridden.**

### 6.2.1 Configuration via the built-in Web server

Access to the built-in Web interface on the *WebdynModbus* gateway is provided via the gateway's LAN connection. An Ethernet crossover cable must be used if a direct connection is made between the computer and the gateway. In addition, both the computer used and the gateway must belong to the same local network. If the *WebdynModbus* gateway has a static IP address (the default situation), the computer must also be configured to use a compatible static IP address.

This static address must belong to the same subnet as the *WebdynModbus* gateway.

**On delivery, the settings for the *WebdynModbus* gateway are as follows:**

- IP address: 192.168.1.12
- Subnet mask: 255. 255. 255.0

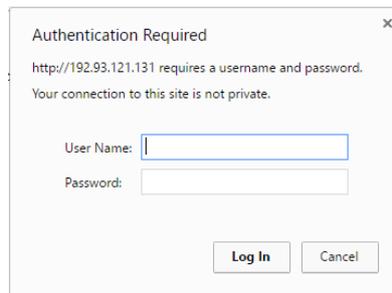


**Network administrator**

If a network administrator manages your local network, contact him or her before connecting the *WebdynModbus* gateway up to your network.

**Connecting to the built-in Web interface:**

1. Once your computer has been correctly configured:
2. Launch your Web browser (Internet Explorer, Firefox, etc.).
3. Go to the home page of the *WebdynModbus* gateway using the browser's address bar to specify the address <http://192.168.1.12>.
4. The following window is displayed:



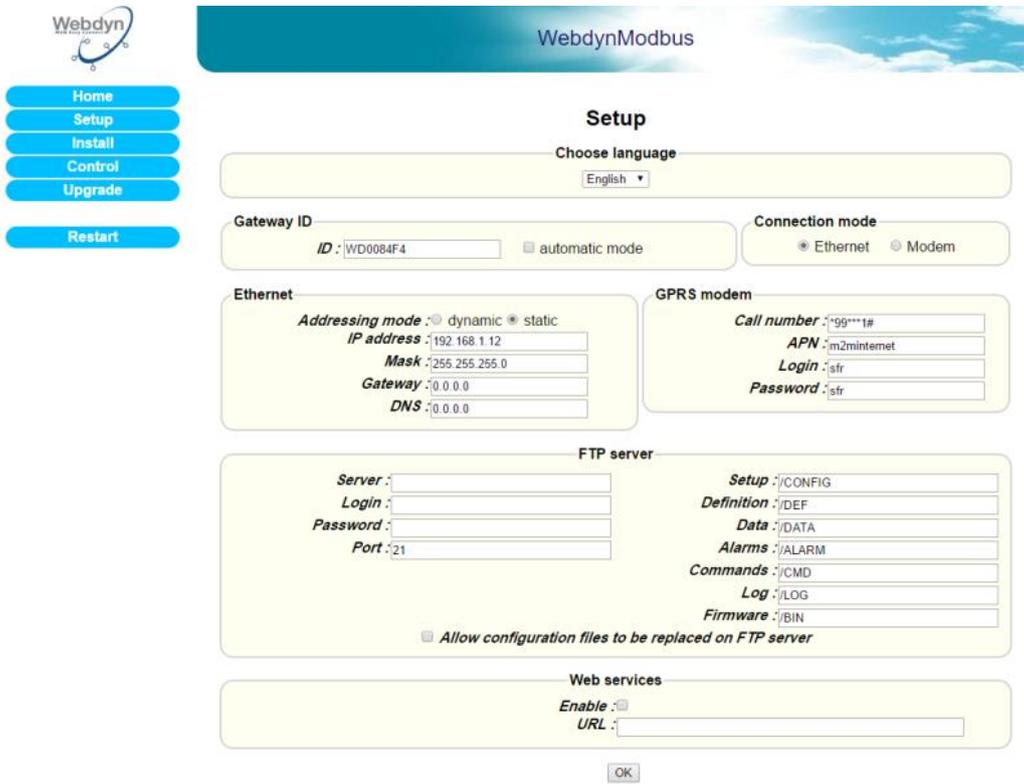
5. Web interface access credentials:



**On delivery, the settings of the *WebdynModBus* gateway are as follows:**

User Name: userhigh  
Password: high

6. The following home page is displayed:



If

the

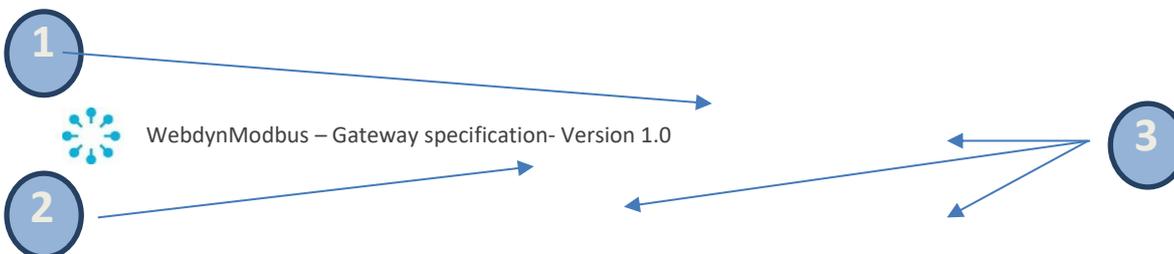
gateway is not yet operating, the following message is displayed:



**General configuration:**

The *Configuration* page allows you to:

- Choose the Web interface language.
- Configure the gateway identifier
- Complete the GPRS / Ethernet connection parameters, and the http/ftp servers.



### 1 Choosing the language:

Selection the Web interface language from the popup menu.

This field corresponds to the variable "Language" in the configuration file *prefixID\_config.ini*.

### 2 Gateway identifier:

This field corresponds to the variable "ID" in the configuration file *prefixID\_config.ini*. The value of this variable enables identification of the gateway when interacting with the FTP server. The names of the files available on the server will be prefixed with the value of this variable to link them to the gateway (i.e. the site) concerned.

There are two ways of configuring the gateway identifier:

- Manually, in the *ID* field (by default "ID=WDXXXXXX" where XXXXXX is the last six digits of the MAC address).
- Automatically, by ticking the *Automatic mode* checkbox and leaving the *ID* field empty ("ID= "). For this option, Web Services must be enabled. The gateway will pick up its identifier just before the initial connection to the FTP server. The ID will be filled in with the value returned by the Web Services server. Should the variable "ID" be deleted once more ("ID= "), the gateway will request a new identifier on its next connection.

### 3 Connection mode:

Choose "Ethernet" or "Modem", depending on the mode to be used for the connection.

#### Ethernet:

If the connexion mode chosen is "Ethernet", enter parameters that are valid for your Ethernet network:

- *Address mode*: you can obtain the Ethernet parameters automatically if the network infrastructure and the version of the WebdynModBus can handle this. If this is the case, click

the *dynamic* radio button and consult the configuration of your DHCP server to find the IP address attributed to your gateway.

- *IP Address*: enter the IP address at which the WebdynModBus gateway is accessible.
- *Mask*: enter the subnet mask for your network. This mask limits the Ethernet network to one range of defined IP addresses, and separates one network range from another.
- *Gateway*: enter the address of the gateway to your network. The gateway address is the IP address IP of the device that establishes the connexion to the Internet. In general, the address entered here is that of the router or ADSL modem.
- *DNS*: enter the address of the DNS server. The DNS (Domain Name System) server translates symbolic Internet addresses (e.g. "www.webdyn.com") into their corresponding IP addresses. Here you should enter the address of the DNS server you received you're your Internet Service Provider (ISP). You can also enter the IP address of your router or ADSL modem.



If a network administrator manages your Ethernet network, contact him or her to have your *WebdynModbus* gateway included in the existing Ethernet network.

**Modem:**

If the connexion mode chosen is "*Modem*", enter parameters that are valid for your GPRS subscription:

- *Call number*: enter the phone number for the GPRS connection. The default number is "*\*99\*\*\*1#*", which is valid in most cases. This number is not the phone number of the SIM card fitted to the unit.
- *APN*: enter the Access Point Name (APN) supplied by your mobile operator.
- *Login*: enter the user ID for the APN supplied by your mobile operator.
- *Password*: enter the password for the APN supplied by your mobile operator.



Consult your mobile operator to obtain the information pertaining to your SIM card (APN, user ID and password).

4

**FTP server:**

To enable the gateway to communicate with a remote FTP server (via Ethernet OR modem), enter the following information:

- *Server*: IP Address or symbolic name of the remote FTP server.
- *Login*: User ID used by the gateway for connection to the remote FTP server.
- *Password*: Password used by the gateway for connection to the remote FTP server.
- *Port*: Port number used for communications with the remote FTP server (default: 21).
- *Setup*: Name of the Configuration directory (default: /CONFIG).
- *Definition*: Name of the Definition directory (default: /DEF).
- *Data*: Name of the Data directory (default: /DATA).
- *Alarms*: Name of the Alarms directory (default: /ALARM).
- *Commands*: Name of the Commands directory (default: /CMD).
- *Log*: Name of the Log directory (default: /LOG).
- *Firmware*: Name of the directory for downloading new firmware (default: /BIN).
- Authorise the overwriting of the distant files: when this option is active, the concentrator can, further to a local modification, overwrite configuration and definition files available on the FTP server.



**Check that the FTP directories defined have been created on the FTP server. The gateway does not create any directories on the server. For UNIX servers, the names are case sensitive (lower/UPPER case).**

5

**Web Services:**

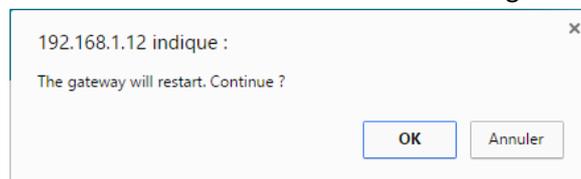
If the gateway uses a connection to Web Services, enable this option and fill in the URL of the server.

Otherwise, disable this option.



**Once all the parameters have been specified, click on OK. A message will appear at the top of the page to indicate that the gateway must be restarted to take the new parameters into account.**

Click on Restart in the menu on the left and confirm this in the dialogue box:



Wait for the *WebdynModbus* gateway to complete the restart, then reconnect to the built-in Web server.



**It is advisable to force connection to the remote server after restarting the unit. This is to check that all the new parameters are correct.**



To ensure consistency between the server and the unit, it is advisable to delete the configuration file `prefixID_config.ini` from the unit before connecting. In this case, the gateway will generate and upload a new `prefixID_config.ini` file.

### 6.2.2 Configuration via SMS

It is possible to configure the information required for connection to the remote FTP server via SMS. To do so, you must first insert an active GPRS SIM card into the *WebdynModbus* unit and ensure that you know its telephone number.

#### SMS to configure the APN:

Send the following SMS to the *WebdynModbus*:

**`"apn=apn_name;usr=user_name;pwd=password;"`**

Where:

*apn\_name*: Name of the APN.

*user\_name*: User ID for access to the APN.

*password*: Password for access to the APN.

#### SMS to configure the remote FTP server:

Send the following SMS to the *WebdynModbus*:

**`"ftp=server_name:user_name:password:port;"`**

Where:

*server\_name*: Symbolic name or IP address of the remote FTP server.

*user\_name*: User ID for access to the remote FTP server.

*password*: Password for access to the remote FTP server.

*port*: TCP port number for access to the remote FTP server (default: 21).



It is advisable to force connection to the remote server after restarting the unit. This is to check that all the new parameters are correct

#### SMS to connect to the remote FTP server:

Send the following SMS to the *WebdynModbus*:

**`"Connect"`**



To ensure consistency between the server and the unit, it is advisable to delete the configuration file `prefixID_config.ini` from the unit before connecting. In this case, the gateway will generate and upload a new `prefixID_config.ini` file

## 6.3 Setting the unit date and time

The *WebdynModbus* unit timestamps all its data and log entries. Thus, it is necessary for its time to be correctly synched with UTC time. The real-time clock is therefore synchronised with a remote NTP server as a matter of course every time the unit connects to the Internet.

To choose an NTP server, you must modify the variables listed below in the *prefixID\_config.ini* configuration file available on the server, then force the unit to connect to the remote server.

Variable	Definition	Default value
NTP_Server1	IP address for the main NTP server (up to 29 characters)	pool.ntp.org
NTP_Server2	IP address for the backup NTP server (up to 29 characters)	

An option to force NTP resynchronisation after a restart following a power failure is enabled by setting to 1 the variable “NTP\_SyncPowerLoss” in the configuration file *prefixID\_config.ini*.

Variable	Definition	Default value
NTP_SyncPowerLoss	Option to force NTP resynchronisation after a power failure. If this option is enabled, an NTP connection will be established after the gateway restarts following a power failure. 0=Disabled 1=Enabled	0



**All timestamping of data and events is carried out using GMT.**

## 6.4 Connection modes and period

There are four modes of connection to a remote server:

- Manual connection
- Periodic automatic connection
- Automatic connection at fixed times
- Automatic connection on data capture
- Automatic connection on alarm

In the case of automatic connection, the connection type is chosen depending on the *prefixID\_var.ini* configuration file.

The *WebdynModbus* always carries out the same tasks, regardless of the type of connection requested:

- *NTP synchronisation*
- *Connection to the remote FTP server*
  - *Alarm management*
  - *Command file management*
  - *Data management*

- Configuration file management
- Definition file management
- Log management
- Firmware update management

### 6.4.1 Manual connection

- **Connection by pressing the push-button:**

You can force a connection manually by using a tool to press and hold the push-button labelled “INSTALL” until the “SERVICES” LED begins to flash rapidly.

- **Connection via the built-in Web server:**

You can force a connection manually via the built-in Web interface by going to the “Install/Connection” menu and clicking on the “Connect” button.



- **Connection via SMS:**

Send the SMS “connect” to the WebdynModBus to force immediate connection.

### 6.4.2 Periodic automatic connexion

Periodic automatic connection consists of providing the WebdynModBus with a period for connection to the remote server. This period is expressed in hours and is repeated every day.

This is done by using the variables “Connection\_Period”, “Connection\_Hour” and “Connection\_Minute” in the configuration file *prefixID\_var.ini*.

Variable	Definition	Default value
Connection_Period	Period for connection to the remote FTP server (in hours, range 0 to 23) If >0: number of hours between two connections. The variable “Connection_Minute” is used to specify the minute number within the hour for the connection. If =0: connection every day at the time specified by the variables “Connection_Hour” and “Connection_Minute”	0
Connection_Hour	Hour for connection to the remote FTP server	1

Connection_Minute	Minute for connection to the remote FTP server	0
-------------------	--	---

**Example:**

Configuration:

<i>Connection_Period=7</i> <i>Connection_Minute=25</i>
---

Connection time:

<i>Day d: 00:25, 07:25, 14:25, 21:25.</i> <i>Day d+1: 00:25, 07:25, 14:25, 21:25.</i>
--

### 6.4.3 Automatic connection at fixed times

Automatic connection at fixed times consists of programming the *WebdynModbus* with up to 5 times for connecting to the remote server per day.

This mechanism is only activated if the variable "Connection\_Period" is equal to 0.

Programming the connection times is carried out by updating the variables listed below in the configuration file *prefixID\_var.ini*.

Variable	Definition	Default value
Connection_Hour	Hour for connecting to remote FTP server	1
Connection_Minute	Minute for connecting to remote FTP server	0
Connection_Hour1	Hour for connecting to remote FTP server	0
Connection_Minute1	Minute for connecting to remote FTP server	0
Connection_Hour2	Hour for connecting to remote FTP server	0
Connection_Minute2	Minute for connecting to remote FTP server	0
Connection_Hour3	Hour for connecting to remote FTP server	0
Connection_Minute3	Minute for connecting to remote FTP server	0
Connection_Hour4	Hour for connecting to remote FTP server	0
Connection_Minute4	Minute for connecting to remote FTP server	0

**Example:**

- Configuration:

<i>Connection_Hour=7</i> <i>Connection_Minute=5</i> <i>Connection_Hour1=12</i> <i>Connection_Minute1=10</i> <i>Connection_Hour2=18</i> <i>Connection_Minute2=15</i>
--

- Connection times:

<i>Day d: 07:05, 12:10, 18:15.</i>
------------------------------------

*Day d+1: 07:05, 12:10, 18:15.*

#### 6.4.4 Automatic connection on data capture

Automatic connection on data capture consists of instructing the *WebdynModbus* to connect to the remote FTP server, to upload the newly captured data, as soon as it is available. This is configured by setting the variable “*Connection\_OnDataAcquisition*” in the configuration file *prefixID\_var.ini* to 1. In this operating mode, configuration checking and time synchronisation still take place periodically or at fixed times, depending on the configuration, as shown in the foregoing chapters.

##### Example:

- Configuration:

```
Connection_OnDataAcquisition=1
Connection_Period=0
Connection_Hour=23
Connection_Minute=0
```

*And the data capture period is defined to be 15 minutes:*

```
DAQ_Period=15
```

- Connection times:

```
Every 15 minutes for uploading data
At 23:00 for time synchronisation and configuration checking.
```

#### 6.4.5 Keeping the connection open

With a view to optimising the connection when automatic connection on data capture has been enabled, it is possible to keep the connection open to avoid pointless disconnections and reconnections.

This is carried out by correctly configuring the variable “*Connection\_WaitBeforeCloseDelay*” in the configuration file *prefixID\_var.ini*.

This delay, which is expressed in minutes, must be greater than the data acquisition time “*DAQ\_Period*” defined in the file *prefixID\_daq.ini*. The maximum authorised value is 59 minutes.

##### Example:

- Configuration:

```
Connection_OnDataAcquisition=1
Connection_WaitBeforeCloseDelay=5
Connection_Period=0
Connection_Hour=23
Connection_Minute=0
```

*And the data acquisition period is defined as 2 minutes:*

```
DAQ_Period=2
```

- Connection times:

Connection kept open with data being uploaded every 2 minutes  
 Time synchronisation and configuration checking occurs at 23:00 every day.

#### 6.4.6 Automatic connexion on alarm

By default, the *WebdynModBus* will trigger a connection to the remote FTP server immediately after the alarm will be detected. However, to limit the number of exchanges with FTP server, it's possible to disable this functionality and to delay the connection at the next acquisition point. In this second case, the product will collect all the alarms and they will be uploaded to the FTP server after the next acquisition point.

To enable or disable this functionality, you must change the value of the variable « ALM\_Delay » in the file *prefixID\_var.ini*.

Variable	Definition	Default value
ALM_Delay	0 : Alarms sent in real time 1 : Alarms sent after the next acquisition point (by default 10 minutes)	0

#### 6.4.7 Optimising the connexion

To avoid excessive GPRS consumption, it is possible to enable an optimisation option for FTP communications. This is specified through the variable *Connection\_CheckConfigPeriod* in the configuration file *prefixID\_var.ini*.

When this optimisation is enabled, the gateway can be programmed not to analyse the configuration and definition directories on every connection.

Variable	Definition	Default value
Connection_CheckConfigPeriod	0: Disabled n: Number of days between 2 analyses.  Note: When the gateway processes a command file, it goes on to analyse the configuration and definition directories regardless of whether optimisation has been enabled.	0

Where the variable *Connection\_Period* is zero, the time for the analysis is defined by the variable *Connection\_Hour*. If it is not, the gateway will launch the analysis during the first connection of the day.

### 6.5 Acquisition period and time slots

The role of the *WebdynModbus* is to collect data from different modbus sources (inverters, meters, sensors, etc.), and then to write them periodically to CSV files for provision via a remote FTP server.

Data collection is scheduled using the variables DAQ\_Period and DAQ\_PeriodSec in the configuration file *prefixID\_daq.ini*.

Variable	Definition	Comments	Default value
DAQ_Period	Collection interval in minutes common to all data sources	Possible value from 0 to 59 minutes	10
DAQ_PeriodSec	Collection interval in seconds common to all data sources Considered only if DAQ_Period is equal to 0.	Possible value from 0 to 59 seconds	0



**If the collection period configured is less than the real data acquisition period, the data will be timestamped at the acquisition period.**

A data acquisition time slot can be defined using the variables listed below, in the file *prefixID\_var.ini*.

Variable	Definition	Default value
DAQ_TimeZoneStartHour	Acquisition start hour	0
DAQ_TimeZoneStartMinute	Acquisition start minute	0
DAQ_TimeZoneStopHour	Acquisition end hour	0
DAQ_TimeZoneStopMinute	Acquisition end minute	0

## 7 Modbus device management

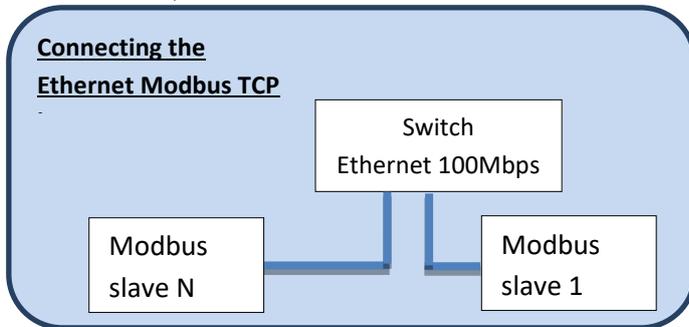
This chapter describes the full set of features that enable the *WebdynModBus* to manage Modbus devices. It can handle up to 247 Modbus slaves, but this limit depends both on the type of slave present on the bus and the number of variables to be collected from each device.

### 7.1 Bus wiring

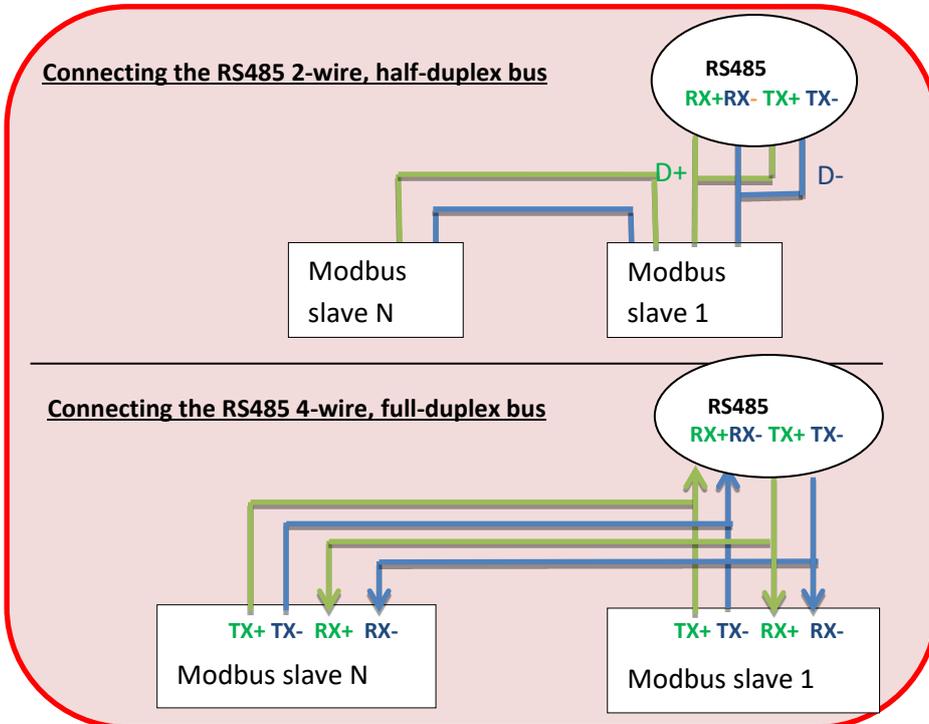
Communication with Modbus devices can be done via RS485 (using 2 or 4 wires), RS232 or Ethernet. This choice must be informed to the gateway via the configuration file *prefixID\_daq.ini*.



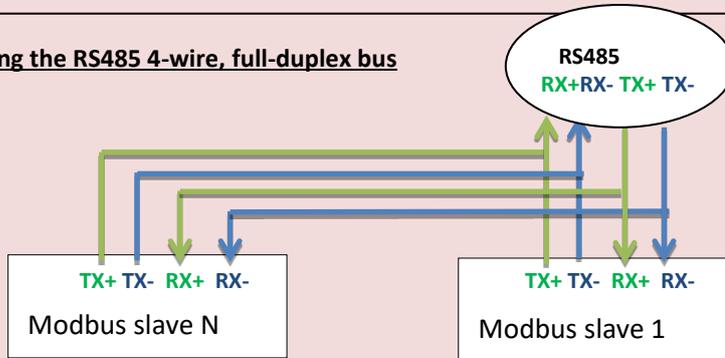
**Connecting the Ethernet Modbus TCP**



**Connecting the RS485 2-wire, half-duplex bus**



**Connecting the RS485 4-wire, full-duplex bus**

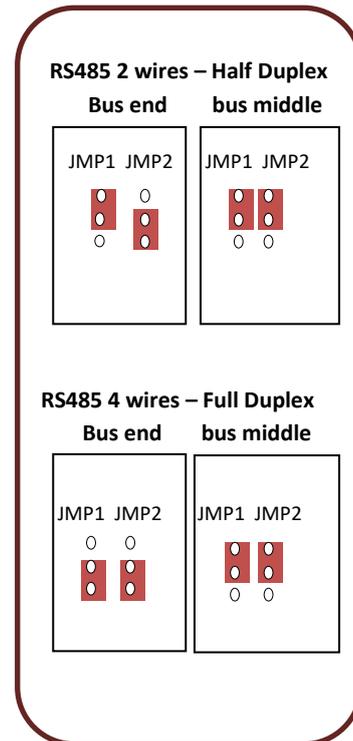
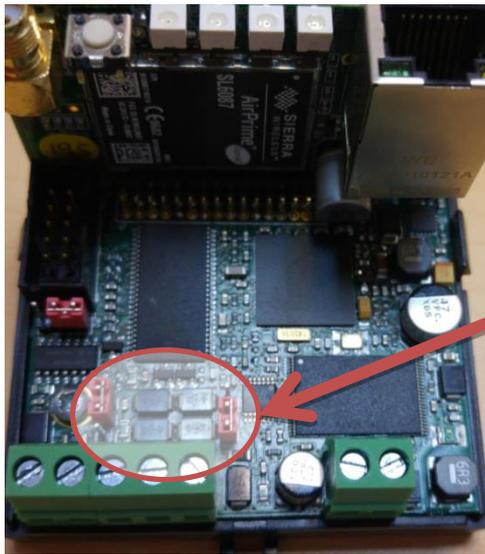


If the installation uses RS485 wiring, the gateway can be placed at the end or in the middle of the communication bus. To ensure correct operation of the RS485 communication bus, it must be terminated at both ends using a 120 ohm terminator.

Depending on the positioning of the gateway on the bus, this terminator must be enabled or disabled via a pair of jumpers (JMP4 and JMP1) fitted inside the casing.

**Configuration of bus termination jumper**

**JMP1 et JMP2**



See the documentation for the Modbus devices for details of their connections and wiring.

## **7.2 Configuring and declaring Modbus slave devices**

The choice between RS232, RS485 2-wires, RS485 4-4wires, or Ethernet is exclusive. Consequently, all the Modbus equipments connected to the WebdynModbus have to use the same interface and must be configured as before. Only the slave's address(es) must be unique, thus allowing their identification on the bus.

The discovery of Modbus devices is impossible. Thus, every device present on the bus must be declared in the *prefixID\_daq.ini* file. This configuration is done via the FTP server. The file *prefixID\_daq.ini* must contain the parameters listed below:

**Parameters that are common to all interfaces:**

Variable	Definition	Comments	Default value
DAQ_Period	Collection interval in minutes common to all data sources	Possible value from 1 to 59 minutes	10
DAQ_PeriodSec	Collection interval in seconds common to all data sources Considered only if DAQ_Period is equal to 0.	Possible value from 0 to 59 seconds	0
DAQ_HeaderOption	Enable/Disable display of column headers in the data files 0=disabled 1=enabled		0

**Generic parameters that characterise the bus:**

Variable	Definition	Comments	Default value
MODBUS_Mode	Mode for serial communication with Modbus devices: 0: RS232 1: RS485, 2 wires 2: RS485, 4 wires	Bus RS232 or RS485 with 2 or 4 wires	1
MODBUS_BaudRate	Speed of the serial link: 1200 2400 9600 19200 38400 57600 115200		9600
MODBUS_Parity	Parity of the serial link: 0: None 1: Even 2: Odd		0
MODBUS_DataBit	Number of data bits: 8 7		7
MODBUS_StopBit	Number of stop bits: 1 2		1

**Specific parameters for each Modbus slave unit:**

Variable	Définition	Comments	Default value
MODBUS_Addr[n]	Address of Modbus device <i>n</i> between 1 and 247	n=0 to 246	1
MODBUS_Name[n]	Name of Modbus device <i>n</i> (up to 29 characters)	n=0 to 246	empty
MODBUS_FileDefName[n]	Name of definition file for Modbus device <i>n</i> (up to 59 characters)	n=0 to 246	empty
MODBUS_Type[n]	Device type: generic or specific if not 0	n=0 to 246  Used only in specific cases such as Mersen devices.	0
MODBUS_Interface[n]	Medium used: 0: Serial interface (RS485/RS232) 1: Ethernet	n=0 to 246	0
MODBUS_IpAddr[n]	IP address of the device where communication is via Modbus TCP.	n=0 to 246	empty

**Example:**

- *Configuration of the bus:*

Mode RS485, 2 wires, baud rate: 19200 baud, 8 data bits, 1 stop bit, no parity.

```

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

- *Declaration of Modbus slaves:*

Slave 0: Mersen GreenString junction box: specific Mersen Modbus (Type 1), at address 1, named "SLAVE1" and defined by the definition file MODBUS\_GREENSTRING.ini.

Slave 1: Standard Modbus device: Generic Modbus (Type 0), at address 2, named "SLAVE2" and defined by the definition file 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
```

### 7.3 Structure of a Modbus definition file

Every Modbus slave declared in the configuration file *prefixID\_daq.ini* must have a definition file.

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

For each variable, it describes:

- Collection method: used by the gateway to collect the data.
- Processing method: average, instantaneous, parameter or alarm.
- Formatting: name, unit and scaling coefficient.

This file must be made available to the gateway on the FTP server.

By convention, the filename has the following format:

*prefixID\_MODBUS\_deviceName.ini*, where “*prefixID*” corresponds to the gateway identifier and “*deviceName*” corresponds to the name of the device.

A single file may be used for many devices that behave in exactly the same way.

The file contains two declaration tables:

- The declaration table for Modbus requests, named “Modbus\_RequestsTables”. Each entry in this table is defined by 10 fields separated by semicolons.

Field	Description
<i>index</i>	Index for the request, from 1 to N
<i>name</i>	Designation of the Modbus request
<i>readFunctionCode</i>	Read function code and function sub-code 0 to 65535  Standard codes handled: 1 => "read coils" 2 => "read discrete inputs" 3 => "read holding registers" 4 => "read input registers"
<i>writeFunctionCode</i>	Write function code and function sub-code 0 to 65535  Standard codes handled: 5 => "write single coils" 6 => "write single register" 15 => "write multiple coils" 16 => "write multiple registers"
<i>startRegister</i>	Address of the first Modbus register 0 to 65535
<i>Size</i>	Length of the response excluding header and CRC (in bytes) 1 to 250
<i>enableReading</i>	Activation of read requests: 0 => enabled in instantaneous mode (parameters) 1 => enabled in polling mode (measurements) 2 => disabled
<i>enableWriting</i>	Activation of write requests: 0 => write enabled with read before write 1 => write enabled, without read 2 => disabled
<i>option1</i>	Reserved for future use
<i>option2</i>	Reserved for future use

- The declaration table for variables, named "Modbus\_VariablesTables". Each entry in this table is defined by 12 fields separated by semicolons.

Champ	Description
<i>index</i>	Index for the variable, from 1 to N
<i>indexRequest</i>	Designation of the Modbus request, defined in the table Modbus_RequestsTables.
<i>name</i>	Designation of the Modbus variable
<i>type</i>	Type of variable 1 => bit 2 => byte 3 => word 4 => reversed word 5 => double word 6 => reversed double word 7 => floating point 8 => reversed floating point 9 => character string 10 => Specific Siebert display format
<i>signed</i>	Sign of the variable 1 => signed 2 => unsigned
<i>position</i>	Position of the variable in the packet 1 to N
<i>option1</i>	Reserved for future use
<i>option2</i>	Reserved for future use
<i>coeffA</i>	Scaling coefficient A for the variable (Ax +B)
<i>coeffB</i>	Scaling coefficient B for the variable (Ax +B)
<i>unit</i>	Units of the variable
<i>action</i>	Processing method for the variable: 0: variable not collected. 1: variable handled as a parameter. 2: minimum, maximum and calculated average collected. 4: instantaneous value 8: alarm triggered on change of status

**Example:**

Definition of a Modbus slave handling 4 pulse counters, 4 outputs and 4 inputs.

**Declaration of Modbus requests:**

	Request 1	Request 2	Request 3
<b>index</b>	1	2	3
<b>name</b>	counter	output	input
<b>readFunctionCode</b>	3 (read holding registers)	1 (read coils)	3 (read holding registers)
<b>writeFunctionCode</b>	0	5 (write single coil)	0
<b>startRegister</b>	0	4	8
<b>size</b>	4	4	4
<b>enableReading</b>	1 (enabled)	1 (enabled)	1 (enabled)
<b>enableWriting</b>	2 (disabled)	1 (enabled)	2 (disabled)
<b>option1</b>	Not supplied	Not supplied	Not supplied
<b>option2</b>	Not supplied	Not supplied	Not supplied

```

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
}
  
```

**Declaration of variables:**

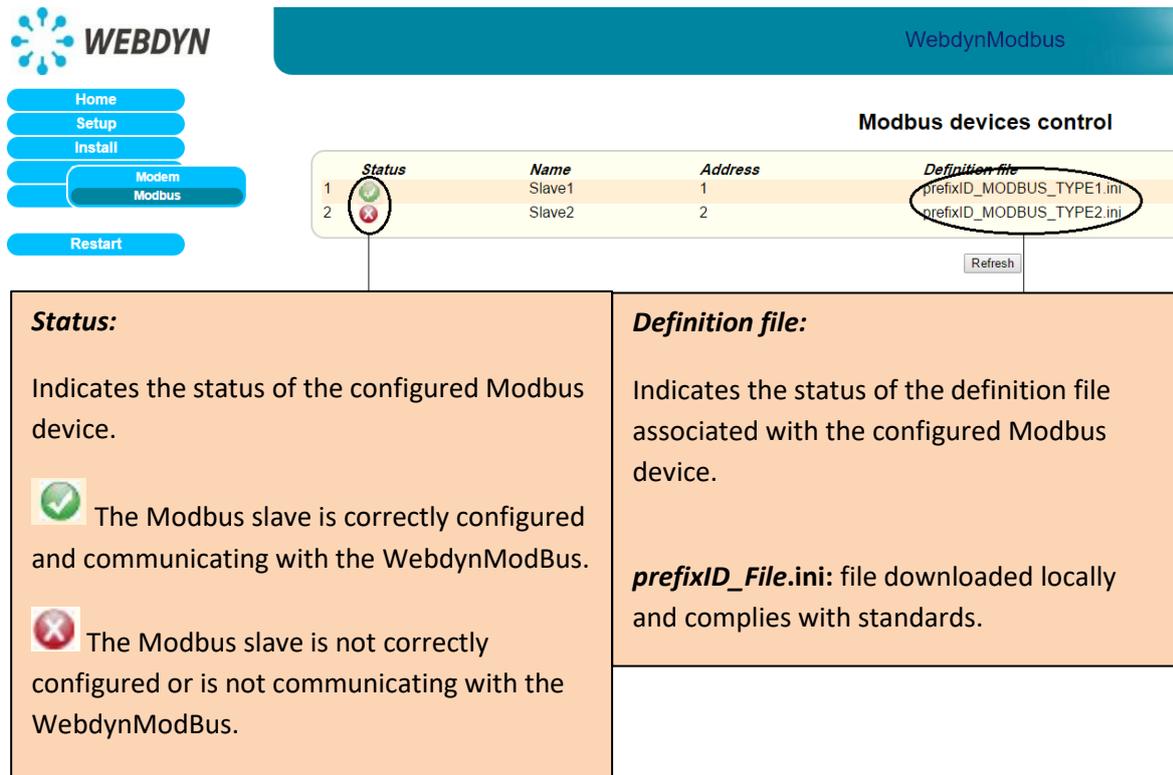
	Variable 1	Variable 2	...	Variable 12
<b>index</b>	1	2		12
<b>indexRequest</b>	1 ("counter" request)	1 ("counter" request)		3 ("input" request)
<b>name</b>	counter1	counter2		input4
<b>type</b>	3 (word)	3 (word)		1 (bit)
<b>signed</b>	2 (unsigned)	2 (unsigned)		2 (unsigned)
<b>position</b>	1 (1st packet item).	2 (2nd packet item).		4 (4th packet item).
<b>option1</b>	Not supplied	Not supplied		Not supplied
<b>option2</b>	Not supplied	Not supplied		Not supplied
<b>coeffA</b>	1	1		1
<b>coeffB</b>	0	0		0
<b>unit</b>	pulse	pulse		Not supplied
<b>action</b>	4 (instantaneous value)	4 (instantaneous value)		8 (alarm)

```

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
}
    
```

## 7.4 Checking that Modbus devices are operating correctly

It is advisable to check that Modbus devices are operating correctly after they have been installed and configured. This can be done via the built-in Web server by going to the “Control/Modbus” menu:



**Modbus devices control**

	Status	Name	Address	Definition file
1		Slave1	1	prefixID_MODBUS_TYPE1.ini
2		Slave2	2	prefixID_MODBUS_TYPE2.ini

Refresh

**Status:**

Indicates the status of the configured Modbus device.

-  The Modbus slave is correctly configured and communicating with the WebdynModBus.
-  The Modbus slave is not correctly configured or is not communicating with the WebdynModBus.

**Definition file:**

Indicates the status of the definition file associated with the configured Modbus device.

**prefixID\_File.ini:** file downloaded locally and complies with standards.

You can also look at the RS485/232(B) LED on the front panel of the unit to check on the activity over the Modbus bus. This LED flashes rapidly on reception of Modbus packets.

## 7.5 Modbus data

Once it has been configured, the *WebdynModBus* constantly collects data from the Modbus devices, then writes it to a text file in CSV format. This file is compressed in GZ format, then uploaded periodically to the FTP server for subsequent operations.

### 7.5.1 Filename syntax

The data file uploaded to the FTP server complies with the following format:

```
prefixID_MODBUS_YYMMDD_hhmmss.csv.gz
```

**Where:**

*prefixID*: gateway identifier.

*YYMMDD\_hhmmss*: timestamp for the archive in the format “year-month-day-hour-minute-second”.

### 7.5.2 Data format

The file format is as follows: (fields in green are optional data that can be enabled or disabled in *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
```

**Where:**

*slaveAddr\_N*: address of the Modbus slave (1 to 247).

*NumDeviceN*: index of the device in DDD format (001 to 247)

*fileDefinitionName\_N*: name of the definition file associated with the device.

*nbVariableDeviceN*: number of variables collected for each device.

*indexVariable\_x\_DeviceN*: index of the variable collected.

*date-time\_n*: timestamp of the data capture in YY/MM/DD-hh:mm:ss format.

*variable\_x\_value\_n*: value n of variable x captured at date-time n.

**With the definition file being:**

```
# Definition of packets
#
Id;Name;ReadFctCode;WriteFctCode;StartReg;NbReg;EnableReading;EnableWriting;Option1;Option2
Modbus_RequestsTables={
1;request;3;0;0;4;1;2
}
# Definition of 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
}
```

**Special case: averaged data item:**

If a data item is configured as Min/Max/Average in the definition file, it will appear in the data file in the following manner:

```
nbVariableDeviceN;indexVariable_1(min);indexVariable_1(max);indexVariable_1(avg);
date-time_n;variable_x_value_n_min;variable_x_value_n_max;variable_x_value_n_avg
```

### 7.5.3 Example

**2 Modbus devices, with acquisition every 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 (avg);2 (min);2 (max);2 (avg)
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
```

**Where the two definition files are:**

*prefixID\_MODBUS\_TYPE1.ini*

```

# Definition of packets
#
Id;Name;ReadFctCode;WriteFctCode;StartReg;NbReg;EnableReading;EnableWriting;Option1;Option2
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
}
# Definition of 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*

```

# Definition of packets
#
Id;Name;ReadFctCode;WriteFctCode;StartReg;NbReg;EnableReading;EnableWriting;Option1;Option2
Modbus_RequestsTables={
1;voltage;3;0;0;4;1;2
}
# Definition of 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
}

```

On the server side, a link must be set up between the data received and the corresponding definition files.

**After the data are formatted, we obtain the following results:**

Device at address 1:



Counter values:

	counter1	counter2	counter3	counter4
27/03/13-09:45:00	32 pulses	52 pulses	5 pulses	102 pulses
27/03/13-10:00:00	35 pulses	57 pulses	5 pulses	108 pulses

Status of outputs:

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

Status of inputs:

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

Device at address 2:

Measured tension values:

	U1			U2		
	min	max	avg	min	max	avg
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

## 7.6 Modbus alarms

A variable declared as an alarm (action field = 8 for the variable in the definition file) causes an alarm to be triggered if its' status changes. This alarm is written to a file in CSV format. This file is compressed into GZ format, then uploaded to the FTP server at the next acquisition time.

### 7.6.1 Alarm filename syntax

The alarm file uploaded to the FTP server complies with the following format:

```
prefixID_AL_YYMMDD_hhmmss.csv.gz
```

**Where:**

*prefixID*: gateway identifier.

*YYMMDD\_hhmmss*: timestamp of the archive in the format "year-month-day-hour-minute-second".

### 7.6.2 Format of alarms

The uploaded CSV alarm file may contain several alarms from different sources. It complies with the following format:

```
date-time_1;AlarmSource1;fileDefinitionName;deviceName;indexVariable,value
```

*date-time\_N*;AlarmSource*N*;fileDefinitionName*N*;deviceName*N*;indexVariable*N*,value*N*

**Where:**

*date-time\_N*: timestamp when the alarm was triggered, in format YY/MM/DD-hh:mm:ss  
*AlarmSourceN*: source of alarm triggering: here MODBUS.  
*fileDefinitionName\_N*: name of the definition file associated with the triggering device.  
*deviceNameN*: name of the triggering device.  
*indexVariableN*: index of the variable raising this alarm.  
*valueN*: value of the variable that raised this alarm.

### 7.6.3 Example of alarm files

The alarm file *prefixID\_AL\_130327\_100305.csv.gz* was received after a change in the status of inputs input4 and input2 of the device Slave 1.

The file contains the following information:

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

With the definition file *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
}
# Definition of 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
}
```

In the configuration file *prefixID\_daq.ini*:

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

**After analysis of the alarm file and the definition files, we obtain the following result:**

On 27/03/13 at 10:01:16 the variable input4 indexed at 12 on the device named Slave1 switched to 1.

On 27/03/13 at 10:01:18 the variable input2 indexed at 10 on the device named Slave1 switched to 0.

## **7.7 Writing Modbus variables via a command file**

Some tasks, known as “commands”, may be remotely requested from the *WebdynModBus*. These commands are transmitted to the gateway in the form of files downloaded from the FTP server (*prefixID\_CMD.csv*). This file can contain several types of commands, including the command to write Modbus variables. It is deleted from the server by the gateway after downloading. After the commands are executed, an acknowledgement file is sent to the server (*prefixID\_ACK\_YYMMDD\_hhmmss.csv*).

**Command file:** *prefixID\_CMD.csv*.

The parameters of the commands depend on the type of command sent, as indicated below:

<b>index;MODBUS;indexDevice;indexVariable;value</b>
---

**Where:**

index	1 to N: Unique identifier providing command identification
indexDevice	Index of the Modbus device to be configured. This index corresponds to the index used in the file daq.ini to declare the device (0 to N).
indexVariable	Index of the Modbus variable to be configured. This index corresponds to the index identifying the variable that is found in the definition file of the device to be configured.
value	Value of the parameter, in decimal or ASCII.

**Acknowledgement file:** *prefixID\_ACK\_YYMMDD\_hhmmss.csv*.

The acknowledgement file mirrors the command file, with timestamps added, and the acknowledgement:

<b>Date-time;index;MODBUS;indexDevice;indexVariable;value;ack</b>
---

Where ack=OK or ERROR.

## **8 Updating the unit**

The gateway can be updated locally via the built-in Web server, or remotely via the FTP server.

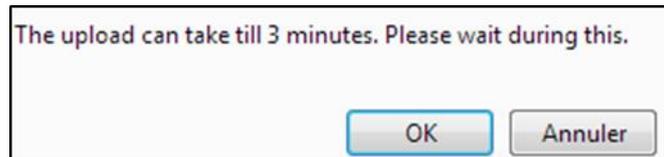
Only “.pak” files containing firmware supplied by Webdyn should be used. For updates done via the FTP server, a “.txt” file containing the checksum of the “.pak” file is also necessary.

### **8.1 Updating via the Web server**

To update your *WebdynModBus* gateway from the “Upgrade” menu of the built-in Web server, go through the following steps:



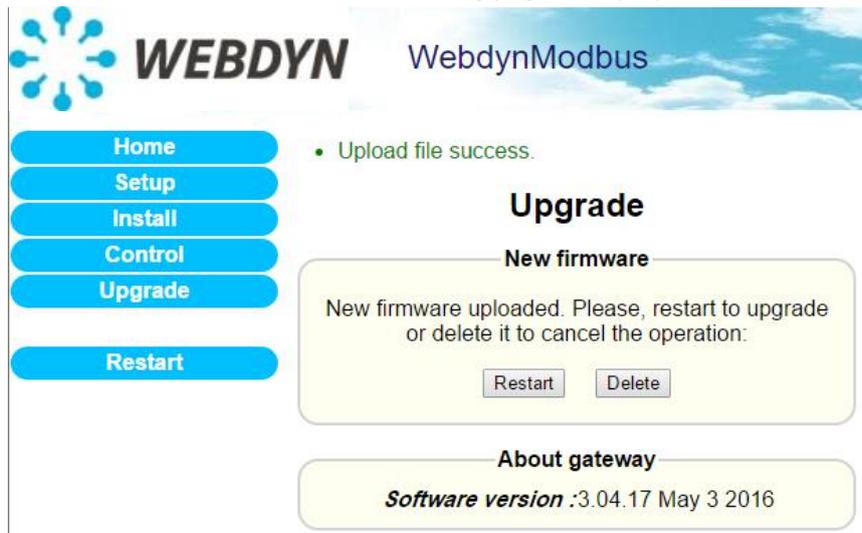
1. Click on Choose file and select the “.pak” file provided by Webdyn.
2. Click on OK.
3. The following message is displayed.



4. Click on “OK” and wait until the file download onto the gateway is complete



5. Once the download has finished, the following page is displayed:



6. Click on “Restart”.
7. The gateway should restart. After a few seconds, all the LEDs should flash, indicating that the update is in progress.
- Warning: do not disconnect the unit from the power supply during this phase.
8. Wait until the update has finished (10–15 minutes) and check the version number on the homepage.

Your *WebdynModBus* gateway is now up to date.

## 8.2 Updating remotely via the FTP server

Proceed as follows for remote updating:

1. Upload the new firmware supplied by Webdyn onto the FTP server.
2. Modify the following variables in the configuration file *prefixID\_config.ini* for the gateway to be updated:
  - BIN\_FileName= name of the new firmware (supplied by Webdyn).
  - BIN\_Checksum= checksum of the new firmware (supplied by Webdyn).
  - FTP\_DirBin= name of the directory containing the new firmware.

The gateway will download its new configuration file, then its new firmware, when it next connects to the FTP server.

## 9 Using web services

The *WebdynModbus* can access the Web Services of an HTTP server in order to retrieve information or to inform the front-end application that an action has been performed on the FTP server, such as upload or download of files. These calls are optional.

## 9.1 Enabling and configuring

The use of Web Services can be enabled via the variables “WebService\_Enable” and “WebService\_Url” in the configuration file *prefixID\_config.ini*.

Variable	Definition	Default value
WebService_Enable	Enable/Disable Web Services: 0=Disabled 1=Enabled	0
WebService_Url	http address of the Web Service (up to 29 characters)	

## 9.2 Format of HTTP requests

There are two types of Web Services:

- initialisation.php => called to obtain a site number, the site ID.
- confirmation.php => called to give notification that an action has been performed on the FTP server.

The POST method is used for requests to the HTTP server.

The format of requests for Web Services complies with the following syntax:

*URL of HTTP server/name of Web Service*

POST data: parameters of the Web Service in the format:

*parameter1&parameter2... parameterN.*

The URL is configured via the variable “WebService\_Url” in the configuration file.

The name of the Web Service can be “initialisation.php” or “confirmation.php”.

The possible parameters are:

MAC-ADR=“MAC address of the gateway” in the format 00:05:F3:XX:XX:XX  
 NSITE=“site number of the gateway”  
 ACTION=“action performed”  
 ACTION-COMP=“supplementary information about the action described by ACTION”  
 RC=“return code”  
 RC-COMP= “supplementary information in ASCII about the return code”

The list of Web Services and the available actions is described below:

### Web Service “initialisation.php”:

Name	POST data	Description
initialisation.php	MAC-ADR=MAC <i>address</i>	Automatic attribution of the site number MAC address format: XX:XX:XX:XX:XX:XX

HTTP response:

ReturnCode##IDsite##

The return codes of the responses from the HTTP server may be:

00: OK
13: MAC-ADR absent
-1: Internal server problem

**Web Service “confirm.php”:**

Name	POST data	Description
confirm.php	NSITE=IDsite& ACTION=UPLOADDATA& RC=0& RC-COMP=	Informs the HTTP server that an inverter, meter, Modbus or input/output data file has been uploaded to the FTP server.
confirm.php	NSITE=IDsite& ACTION=UPLOADALARM& RC=0& RC-COMP=	Informs the HTTP server that an alarms file has been uploaded to the FTP server.
confirm.php	NSITE=IDsite& ACTION=UPLOADGLOBAL& ACTION-COMP= <i>list of files involved</i> & RC=0& RC-COMP=	Informs the HTTP server that configuration files have been uploaded to the FTP server.
confirm.php	NSITE=IDsite& ACTION=CONFIGGLOBAL& ACTION-COMP= <i>list of files involved</i> & RC=0& RC-COMP=	Informs the HTTP server that configuration files have been downloaded from the FTP server.
confirm.php	NSITE=IDsite& ACTION=UPLOADDEF& ACTION-COMP= <i>list of files involved</i> & RC=0& RC-COMP=	Informs the HTTP server that inverter, meter, Modbus or input/output definition files have been uploaded to the FTP server.
confirm.php	NSITE=IDsite& ACTION=CONFIGDEF& ACTION-COMP= <i>list of files involved</i> & RC=0& RC-COMP=	Informs the HTTP server that inverter, meter, Modbus or input/output definition files have been downloaded from the FTP server.
confirm.php	NSITE=IDsite& ACTION=CONFIGINV&	Informs the HTTP server that the file INV.ini has been uploaded to the FTP server.

	RC=0& RC-COMP=	
confirm.php	NSITE=IDsite& ACTION=CMD& RC=0& RC-COMP=	Informs the HTTP server that a CMD file has been downloaded from the FTP server.
confirm.php	NSITE=IDsite& ACTION=CONFIGBIN& RC=0& RC-COMP=	Informs the HTTP server that a firmware file has been downloaded from the FTP server.
confirm.php	NSITE=IDsite& ACTION=VERSION& ACTION-COMP=2.03.01 Aug 1 2011& RC=0& RC-COMP=	Informs the HTTP server of the current version of the gateway firmware. This Web Service is sent on the first connection following a gateway restart.

HTTP response:

ReturnCode

The return codes of the responses from the HTTP server may be:

00: OK  
 10: Site unknown  
 11: Action code unknown  
 12: RC received unknown  
 13: MAC-ADR absent  
 -1: Internal server problem

### 9.3 Examples of Web Services requests

*Upload of a data file:*

URL/confirm.php;NSITE=IDsite&ACTION=UPLOADDATA&RC=0&RC-COMP=

*Download of the configuration files IDsite\_config.ini and IDsite\_var.ini:*

URL/confirm.php;NSITE=IDsite&ACTION=CONFIGGLOBAL&ACTION-COMP=IDsite\_config.ini;IDsite\_var.ini;&RC=0&RC-COMP=

*Download of a new firmware release:*

URL/confirm.php;NSITE=IDsite&ACTION=CONFIGBIN&RC=0&RC-COMP=

## 10 Tools and diagnostics

### 10.1 Events journal

On every connection, the gateway uploads a journal of events to the /LOG directory of the remote FTP server. This indicates the actions it has performed since the previous connection. It is compressed into GZ format GZ and bears the name *prefixID\_YMMMDD\_hhmmss.log.gz*.

**List of messages that can appear in the events journal:**

Message	Description
Error config file <i>[filename]</i> on variable <i>[variable name]</i>	Error in a variable in a configuration file.
FTP connection failed	Error connecting to the FTP server
GPRS signal: <i>[RSSI]</i>	Level of the GSM signal (1 to 31).
Firmware version: <i>[version]</i>	Current version of the firmware.
Restart Gateway	The gateway was restarted.
WAN connection opened	Beginning of the WAN connection.
WAN connection terminated	End of the WAN connection.
FTP get command file OK: <i>[filename]</i>	Command file downloaded OK.
FTP delete command file failed: <i>[filename]</i>	Deletion of command file failed.
FTP send ack command file failed: <i>[filename]</i>	Command acknowledgement file upload failed.
FTP get command file failed: <i>[filename]</i>	Command file download failed.
FTP send config file OK: <i>[filename]</i>	Upload of configuration file <i>[filename]</i> successful.
FTP send config file failed: <i>[filename]</i>	Upload of configuration file <i>[filename]</i> unsuccessful.
FTP get config file OK: <i>[filename]</i>	Download of configuration file <i>[filename]</i> successful.
FTP get config file failed: <i>[filename]</i>	Download of configuration file <i>[filename]</i> failed.
FTP send definition file OK: <i>[filename]</i>	Upload of definition file <i>[filename]</i> successful.
FTP send definition file failed: <i>[filename]</i>	Upload of definition file <i>[filename]</i> unsuccessful.
FTP get definition file OK: <i>[filename]</i>	Download of definition file <i>[filename]</i> successful.
FTP get definition file failed: <i>[filename]</i>	Download of definition file <i>[filename]</i> unsuccessful.
FTP get firmware OK	Firmware successfully downloaded from FTP server.
FTP get firmware failed	Firmware not downloaded from FTP server
FTP send alarm file OK: <i>[filename]</i>	Upload of alarm file <i>[filename]</i> successful.
FTP send alarm file failed: <i>[filename]</i>	Upload of alarm file <i>[filename]</i> unsuccessful.

FTP send data file OK: <i>[filename]</i>	Upload of data file (Inverters, TIC, IO, MODBUS, etc.) named <i>[filename]</i> successful.
FTP send data file failed: <i>[filename]</i>	Upload of data file (Inverters, TIC, IO, MODBUS, etc.) named <i>[filename]</i> unsuccessful.
FTP send log file OK: <i>[filename]</i>	Upload of events journal <i>[filename]</i> successful.
FTP send log file failed: <i>[filename]</i>	Upload of events journal <i>[filename]</i> unsuccessful.
FTP send debug file OK: <i>[filename]</i>	Upload of debug trace file <i>[filename]</i> successful.
FTP send debug file failed: <i>[filename]</i>	Upload of debug trace file <i>[filename]</i> unsuccessful.
FTP send parameters file OK: <i>[filename]</i>	Upload of parameter file <i>[filename]</i> successful.
FTP send parameters file failed: <i>[filename]</i>	Upload of parameter file <i>[filename]</i> unsuccessful.
NTP synchronization OK	Synchronisation of gateway time via Network Time Protocol successful.
NTP synchronization failed	Synchronisation of gateway time via NTP failed.
SMS received:Request reboot	A restart command was received via SMS.
SMS received:Request factory	A factory reset command was received via SMS.
SMS received:Request connection	A connection request was received via SMS.
SMS received:Request Version	A firmware version request was received via SMS.
SMS received:Change FTP parameters Server: <i>[server]</i> Login: <i>[login]</i> Password: <i>[password]</i>	A request to change FTP parameters was received via SMS.
SMS received:Change GPRS number <i>[number]</i>	A request to change the GPRS call number was received via SMS.
SMS received:Change GPRS APN <i>[apn]</i>	A request to change the APN was received via SMS.
SMS received:Change GPRS login <i>[login]</i>	An SMS to change the APN login was received.
SMS received:Change GPRS password <i>[password]</i>	An SMS to change the APN password was received.

**Example:**

```

<0>Apr 04 13:27:10 Application: FTP send debug file OK: /LOG/prefixID_130404_132611_debug.log.gz
<0>Apr 04 14:26:16 Application: Firmware version: 2.05.10 Mar 25 2013
<0>Apr 04 14:26:16 Application: WAN connection opened
<0>Apr 04 14:26:17 Application: NTP synchronization OK
<0>Apr 04 14:26:17 Application: FTP send data file OK: /DATA/INV/prefixID_INV_1_1_130404_142614.csv.gz
<0>Apr 04 14:26:17 Application: FTP send data file OK: /DATA/TIC/prefixID_TIC_130404_142614.csv.gz
<0>Apr 04 14:26:18 Application: FTP send data file OK: /DATA/IO/prefixID_IO_130404_142614.csv.gz
    
```

## 10.2 Modem information

It is advisable to examine the information from the modem to check that it is operating correctly. This is done by going to the page “Control/Modem” of the built-in Web server. The following page is displayed:



- **Modem type:** GPRS.
- **Modem ID:** IMEI number of the modem.
- **SIM ID:** ICCID number of the SIM card.
- **SIM status:** Status of the SIM card. The message may be:
  - *READY:* The SIM card has been inserted and the PIN code is correct. The gateway is ready to open a GPRS connection.
  - *SIM PIN:* The SIM card is waiting for a PIN code.
  - *SIM PUK:* The SIM card is waiting for a PUK code (after three unsuccessful attempts to enter a PIN code).
  - *SIM ERROR:* No SIM card has been inserted.
- **Signal level:** Quality of the GSM signal; range 1 to 31.



**For an operational GPRS connection, this level must be at least 20. It is strongly disadvised to use the connection GPRS with a level lower than 12.**

- **Modem status:** The message may be:
  - *Standby:* modem not connected.
  - *Initialisation:* modem establishing a connection.
  - *Connected:* modem connected.

## 10.3 Detecting power connection

The *WebdynModbus* constantly monitors the status of its 24 V power supply. This enables it to detect a prolonged power cut, and so warn the operator by uploading an alarm file to the FTP server. An alarm indicating that power has been restored is also uploaded.

### 10.3.1 **Syntax of the alarm file name:**

The name of the alarm file uploaded to the FTP server complies with the following format:

```
prefixID_AL_YYMMDD_hhmmss.csv.gz
```

Where:

*prefixID*: gateway identifier.

*YYMMDD\_hhmmss*: timestamp for the archive in the format “year-month-day-hour-minute-second”.

### 10.3.2 Format of alarms

The uploaded CSV alarm file may contain several alarms from different sources. The entry for a 24 V power loss alarm has the following format:

*date-time;GATEWAY;info*

**Where:**

*date-time\_N*: timestamp when the alarm was triggered, in format YY/MM/DD-hh:mm:ss

*info*: *Power OFF* to indicate loss of the 24 V supply

*Power ON* to indicate loss of the 24 V supply

**Example:**

09/10/09-09:09:36;GATEWAY;Power OFF

### 10.4 LED indicators

LED	Function	Status	Explanation
ACTIVITY	Operational status of the gateway	Power on	Power on
		Flashing rapidly	Initialisation
		Flashing slowly	Operational
TX	Activity LED	Flashing rapidly	Initialisation ModBus
		Power on	Initialisation ends
		Flickering	Traffic with Modbus equipments
RX	Activity LED	Flashing rapidly	initialisation Modbus
		Power on	Initialisation ends
		Flickering	Traffic with Modbus equipments
WAN	WAN connection via Ethernet	Flashing rapidly	Ethernet connection being initialised
		On continuously	Initialisation complete
		Flashing slowly	Connection with remote server in progress
	WAN connection via GPRS	Flashing rapidly	GPRS modem being initialised
		Flashes periodically 1 to 5 times	Initialisation complete. Signal strength (number of flashes)

	Connexion WAN (PSTN)	Flashing slowly	Connection with remote server in progress
		Flashing rapidly	PSTN modem being initialised
		On continuously	Initialisation complete
		Flashing slowly	Connection with remote server in progress

**Ethernet socket:**

LED	Function	Status	Explanation
<b>Green LED: speed</b>	Connection speed	Off	10 Mbps
		On	100 Mbps
<b>Orange LED: link activity</b>	Connection	Off	No connection
		On	Connection made
		Flickering	Data being sent or received

## 10.5 Installation button

The *INSTALL* button, fitted on the front panel of the unit, enables forced connection or restart of the gateway as explained in the table:

ACTION	CONSEQUENCE
Press the <i>INSTALL</i> button for about 1 second until the <i>SERVICE</i> LED flashes	Besides the actions defined in §10.3.3 <i>Connection</i> , the gateway uploads the inverter parameter file.
Press the <i>INSTALL</i> button for about 10 seconds until all the LEDs on the gateway go out	The gateway restarts (the LEDs light up again about 1 minute after the restart begins)

## 10.6 Diagnostic SMSs

Besides the configuration SMSs described in previous chapters, some SMSs can enable initial diagnosis of a problem with the *WebdynModbus*:

SMS	Description
connect	Requests a connection to the remote server
version (*)	Requests the current software version of the unit
reboot	Initiates a restart of the product
status (*)	Requests information on the current configuration of the unit: <ul style="list-style-type: none"> <li>• Unit type: <i>WebdynModBus</i></li> <li>• Unit identifier (prefixID)</li> <li>• Software version</li> <li>• Connection mode (GPRS or LAN)</li> </ul>

	<ul style="list-style-type: none"> <li>• Information on the APN configured</li> <li>• SIM card identifier</li> <li>• GSM signal strength (RSSI)</li> <li>• Information on the Ethernet interface (IP, router, DNS, etc.)</li> <li>• Information on the remote FTP server</li> </ul>
diag (*)	Requests diagnostics on the unit interfaces: <ul style="list-style-type: none"> <li>• WAN: status of the WAN connection (OK or ERR)</li> <li>• FTP: status of the connection to the FTP server (OK or ERR)</li> <li>• NTP: NTP synchronisation status (OK or ERR)</li> <li>• WS: status of Web Services (OK or ERR)</li> <li>• MODBUS: status of the Modbus link (OK or ERR)</li> </ul>
Factory	Initiates a factory reset of the unit

(\*) Operational only if the operator contract includes the option to send SMSs.

## 10.7 Debug traces

Occasionally, it is necessary to enable debug traces, so as to be able to diagnose a problem.

This is done by providing values for the parameters listed below, in the configuration file prefixID\_config.ini:

Variable	Definition	Default value
Log_Enable	Enable/Disable debugging logs: 0=Disabled 1=Enabled  Reserved for use by Webdyn in support mode	0
Log_Level	Level of detail in debugging logs: 0 Emerg (emergency) 1 Alert 2 Crit (critical) 3 Err (error) 4 Warning 5 Notice (default value) 6 Info (informational) 7 Debug Reserved for use by Webdyn in support mode	5
Log_RemoteIpAddr	Syslog destination address Reserved for use by Webdyn in support mode (up to 15 characters)	<i>empty</i>

Log_Port	Syslog destination port Reserved for use by Webdyn in support mode	2000
CFG_Debug	Enable/disable configuration traces: 0=Disabled 1=Enabled	0
MODBUS_Debug	Enable/disable Modbus device traces. 0=Disabled 1=Enabled 2=Verbose mode	0
MODEM_Debug	Enable/disable modem traces. 0=Disabled 1=Enabled	0



**Enabling debug traces is likely to generate much greater GPRS traffic.**

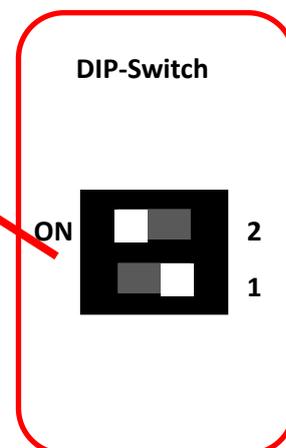
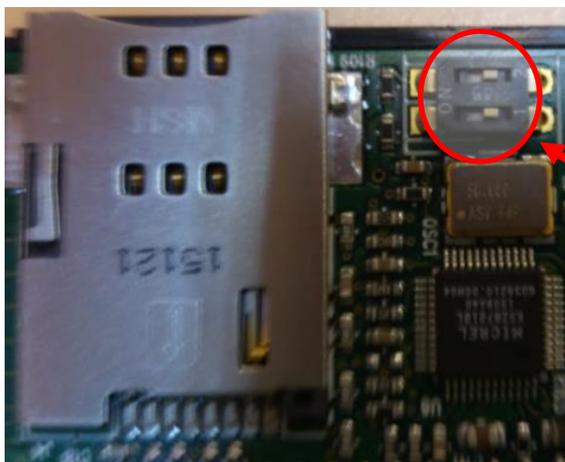
The traces are subsequently sent out via UDP and on every connection, stored in the directory /LOG of the remote FTP server in the form of a compressed file named *prefixID\_YYMMDD\_hhmmss\_debug.log.gz*.

## 10.8 Factory reset procedure

A mechanism to reset the gateway to its factory default parameter settings is provided to deal with problems of access to the *WebdynModbus*.

To carry out this factory reset, please proceed as shown below:

- 1) Turn off the power supply to the unit.
- 2) Open the casing of the *WebdynModbus* to gain access to the configuration DIP switch.
- 3) Set DIP switch 2 to ON.



- 4) Turn on the power supply to the unit

- 5) Wait for the unit to restart automatically, after all the LEDs flash simultaneously (around 2 minutes)
- 6) Reset DIP switch 2 to OFF
- 7) Close the unit.



**A factory reset does not delete the collected data**

## 10.9 Support

If there is a technical problem related to our products, contact Webdyn support:

Webdyn SA

26 Rue des Gaudines,

78100 Saint-Germain-en-Laye,

France

Fax: +33 1 3904 2941

E-mail: [support@webdyn.com](mailto:support@webdyn.com)

Websites: <http://www.webdyn.com>

<http://www.webdynsun.com>

We shall require the following details:

- Serial number of the gateway.
- Hardware and software version numbers of the gateway.