

**WEBDYN**  
flexitron group

## USER MANUAL



WebdynEasy W M-Bus 868MHz

Collect Wireless M-Bus data

# Contents

	Additions on Field L for B format frames.....	7
	Details on the data parameter F and on the diagnostic mode, adjustment of the / scheduleRadio / data range .....	7
1	Presentation .....	8
1.1	General description .....	9
1.2	Operating principle .....	9
1.3	The interfaces .....	11
1.4	Supported equipment.....	12
1.5	Product and accessory references .....	12
1.6	Technical Specifications.....	13
1.6.1	General specifications.....	13
1.6.2	Technical specifications .....	14
1.6.3	Software specifications .....	15
1.6.4	Battery life.....	15
1.7	Safety instructions.....	16
1.8	Regulations .....	17
2	Installation .....	18
2.1	Prerequisite .....	19
2.2	Unpacking .....	19
2.2.1	Contents .....	19
2.2.2	Identification.....	19
2.3	Assembly.....	21
2.3.1	Opening/closing the box .....	21
2.3.2	Wall mounting .....	21
2.4	Interface description.....	21
2.4.1	Product power supply .....	21
2.4.1.1	External power supply .....	21
2.4.1.2	Battery Pack.....	22
2.4.2	Radio .....	23
2.4.2.1	Antenna.....	23
2.4.3	Cellular Network.....	24
2.4.3.1	Antenna.....	24
2.4.3.2	SIM card .....	25
2.4.4	Indicators & buttons .....	26
2.4.4.1	Hall Effect Sensor.....	27
2.4.4.2	RESET button .....	28
2.4.4.3	Modem LED .....	28
2.4.5	USB interface.....	28
2.4.6	Bluetooth interface (BLE).....	29
3	Product operation .....	30
3.1	Starting the product.....	31
3.1.1	Operating mode .....	31
3.1.1.1	FACTORY SETTINGS mode .....	33
3.1.1.2	STORAGE mode.....	33

3.1.1.3	RUNmode.....	34
3.1.1.4	RESET .....	35
3.1.2	Application operation .....	35
3.1.2.1	WM-BUS task.....	36
3.1.2.2	Modem task (connection to the IS).....	36
3.1.2.3	Monitoring task .....	36
4	Configuration .....	37
4.1	Local configuration .....	38
4.1.1	USB .....	38
4.1.2	Bluetooth (BLE).....	38
4.2	Configuration file .....	40
4.2.1	System.....	41
4.2.1.1	Local.....	41
4.2.1.2	Security .....	42
4.2.2	WM-Bus radio .....	43
4.2.2.1	White List.....	44
4.2.2.2	Field L for B format frames .....	45
4.2.3	Hub connectivity.....	46
4.2.3.1	Modem .....	46
4.2.3.2	FTP/FTPS .....	47
4.2.3.3	NTP .....	48
4.2.4	Alarms.....	48
4.2.5	Schedule and Monitoring .....	49
4.2.5.1	Schedules.....	49
4.2.5.2	Monitoring.....	52
5	Operation.....	54
5.1	The remote server.....	55
5.1.1	The FTP server .....	55
5.1.1.1	Configuration .....	55
5.1.1.2	Server tree structure.....	55
5.1.1.3	Operation.....	57
5.1.1.4	File Format .....	57
5.2	The configuration.....	58
5.3	The data .....	59
5.4	Alarms .....	61
5.5	ACK acknowledgements .....	62
5.6	Commands.....	63
5.6.1	"Request" command .....	65
5.6.2	"Factory" command.....	65
5.6.3	"Update" command .....	65
5.6.4	"Status" command.....	65
5.6.5	"Diag" command .....	66
5.6.6	"ConfigGet" command.....	66
5.6.7	"whiteListErase" command .....	67
5.6.8	"operatorInit" command.....	67
5.6.9	"logGet" command .....	67
5.6.10	"logClean" command .....	68
5.6.11	"supervisionClean" command .....	68
5.7	Supervision .....	68
5.8	The Log .....	73

6	Update.....	75
6.1	Local.....	77
6.2	Remote.....	77
7	Tools & diagnostics .....	78
8	Support.....	80
9	FAQ.....	82
10	APPENDICES.....	84
10.1	Appendix A: Configuration - Variable list .....	85

## Glossary

Name	Description
<b>APN</b>	<b>Access Point Name</b>  The name of the access point the gateway uses to connect to the Internet via a mobile connection.
<b>AES</b>	<b>Advanced Encryption Standard</b> – Synthetic encryption algorithm.
<b>BLE</b>	<b>Bluetooth Low Energy</b>  Wireless transmission technique in the form of a Bluetooth-based open standard, characteristically having much lower power consumption
<b>Base64</b>	Encoded data using an alphabet with 64 characters.
<b>BSON</b>	<b>Binary JSON</b>  Binary version of JSON: encoding used to reduce the size of transmitted data to save stand-alone system batteries.
<b>FTP</b>	<b>File Transfer Protocol</b>  Communication protocol used to exchange files over a TCP/IP network.
<b>FTPS</b>	<b>File Transfer Protocol Secure</b>  Communication protocol intended for the computer exchange of files on a TCP / IP network, variant of FTP, secured with SSL or TLS protocols. It allows visitors to verify the identity of the server they are accessing using an authentication certificate. It also makes it possible to encrypt the communication.
<b>2G</b>	<b>Second generation</b>  Second generation digital standard (2G) for mobile telephony including GSM, GPRS and EDGE.
<b>HTTP</b>	<b>HyperText Transfer Protocol</b>  Client-server communication protocol developed for the Web.
<b>IP</b>	<b>Internet Protocol</b>  Message protocol in charge of addressing and sending TCP packets over the network.
<b>JSON</b>	<b>JavaScript Object Notation</b>

	JSON is an easily interpretable data exchange format.
<b>LTE-M/CAT-M1</b>	<b>Long Term Evolution - Machine</b> 4G mobile network specific to connected objects, i.e. low consumption and long range.
<b>NTP</b>	<b>Network Time Protocol</b> An NTP server makes it possible to time synchronise equipment.
<b>IS</b>	<b>Information System</b> Server with which the hub exchanges (configuration, data, alarms, etc.)
<b>TCP</b>	<b>Transmission Control Protocol</b> An Internet-based connection-oriented protocol that provides data packet segmenting services that the IP protocol sends over the network. This protocol provides a reliable data transfer service. See also IP.
<b>TCP/IP</b>	<b>Transmission Control Protocol/Internet Protocol</b> A set of network protocols that provide interconnection services between computers of different hardware architectures and operating systems. TCP/IP includes standards for communication between computers and conventions for network interconnection and routing.
<b>W M-BUS</b>	<b>Wireless M-Bus</b> Evolution of the European Mbus standard in a radio frequency adaptation. This connectivity is specific to metering applications such as: water, gas and electricity meters.

## Notes on this manual

This guide describes all features of the WebdynEasy W M-Bus W M-Bus product.

Its purpose is to help operators install and configure their WebdynEasy W M-Bus W M-Bus and to allow operating entities to include the collected data in their IS.

This manual is split into six separate sections:

- Section 1: General presentation
- Section 2: Installation
- Section 3: Configuration
- Section 4: Operation
- Section 5: Tools & diagnostics
- Section 6: FAQ

## Scope

This technical description is valid for WebdynEasy W M-Bus W M-Bus hubs from hardware version V1 and software version V1.0 onwards.

## Target group

This guide is intended for on-site installers who will cable and configure the installations, for those in charge of local or remote maintenance of the installations, and for developers of portals to use the sent data.

## Change history

Version	Content
V1.0	Creation
V1.1	Modification of the functioning of the product regards the magnet and the number of bips. Adding of a scheme concerning file management.
V1.2	Addition and Modification on the supervision part. Adding autonomy.
V1.3	Additions in the glossary, modification of the command file
V1.4	Additions on Field L for B format frames
V1.5	Details on the data parameter F and on the diagnostic mode, adjustment of the / scheduleRadio / data range

# 1 Presentation



## 1.1 General description

The purpose of the WebdynEasy W M-Bus W M-Bus hub is to collect data from Wireless M-Bus sensors such as meters (water, gas and electricity) and sensors (temperature, humidity, etc...). This technology is energy-efficient, the hub battery service life can exceed 10 years.

Data collection is by radio on the 868MHz frequency. This frequency is free of charge and its use is harmonised in many EU countries.

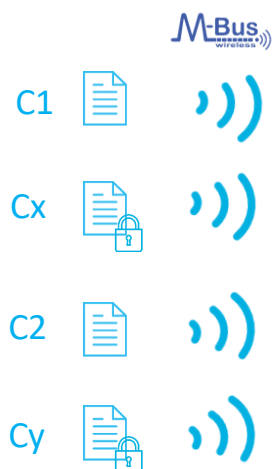
The collected information (data, parameters, alarms, ...) is formatted to BSON format before being sent to an information system (IS) by modem.

## 1.2 Operating principle

WebdynEasy W M-Bus W M-Bus allows you to build your private network and is part of a package. WM-Bus communication is point-to-point in a non-operated way. The meters or sensors regularly send data frames, which are received and the raw data (encrypted or transparent) contained in these frames are stored by the WM-Bus hub. WebdynEasy W M-Bus W M-Bus will regularly send a file in BSON format containing the whole raw data of the different sensors to a server via FTP using its 2G/LTE-M or modem. It is possible to secure the exchanged files between the WebdynEasy W M-Bus W M-Bus and the FTP server by activating the AES encryption (*see chapter 4.2.1.2 : "security"*).

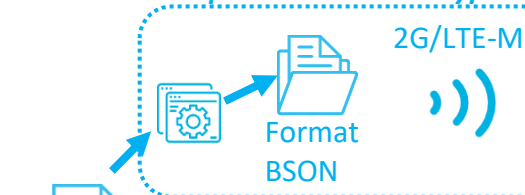
WebdynEasy W M-Bus W M-Bus can be configured by connecting it to a computer via USB or by BLE using a smartphone app.

## Meters or sensors

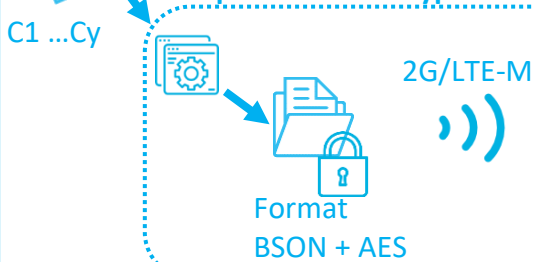


## WebdynEasy

## Option without encryption



## Option with encryption

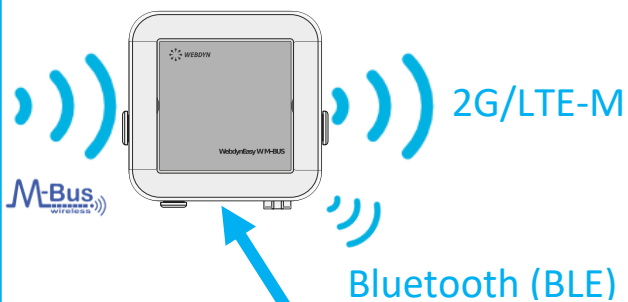


## FTP Server

## Exploitation

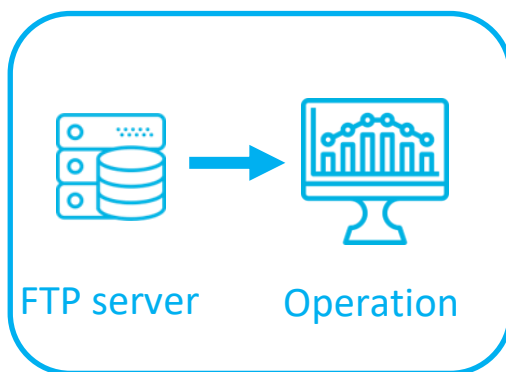

 Meters  
or  
sensors

## WebdynEasy



Local configuration

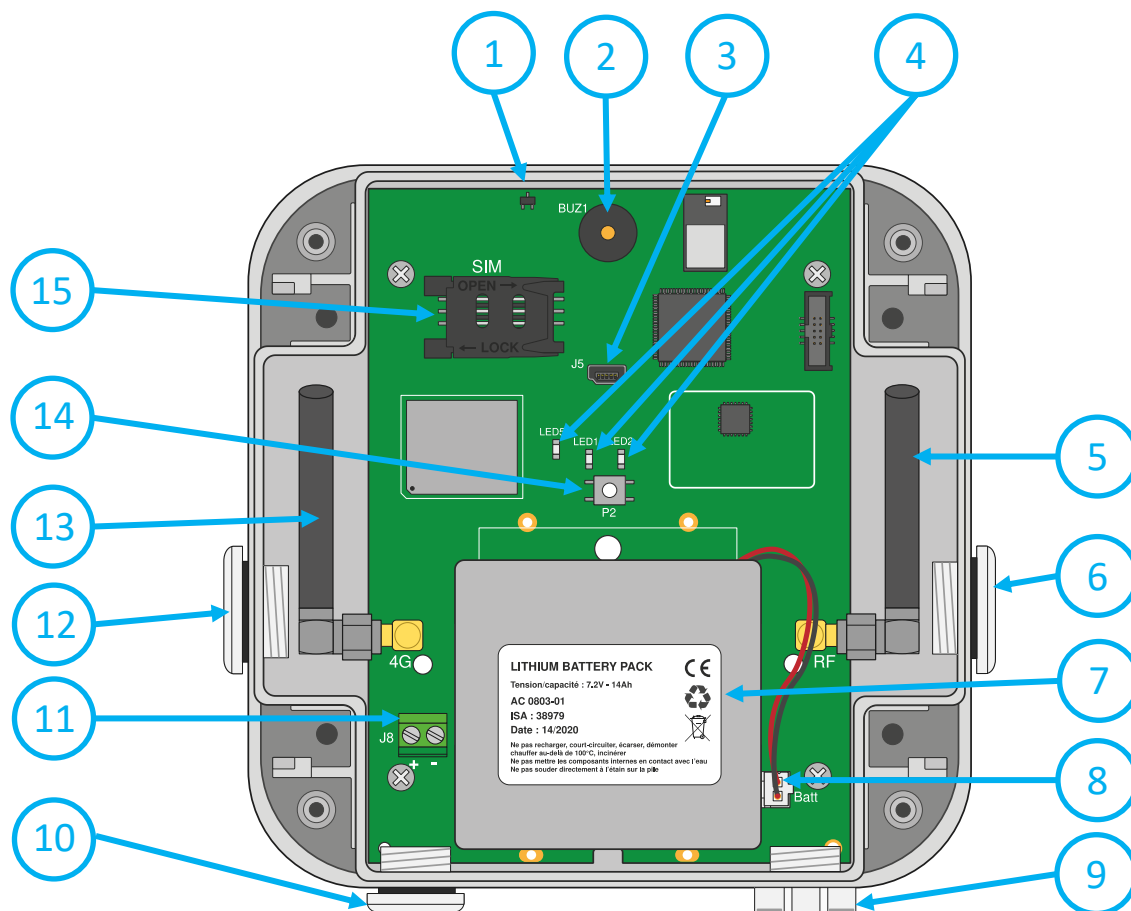
IS



FTP server

Operation

### 1.3 The interfaces



1 HALL effect sensor

2 Buzzer

3 Mini-USB connector

Indicators:

- 4
  - LED 1: reserved
  - LED 2: reserved
  - LED 5: Modem status

5 868MHz radio SMA antenna

6 Box output for the 868MHz radio external antenna (optional)

7 Battery Pack

8 Battery pack connector

- 9 Aerator
- 10 Box output for external power supply (optional)
- 11 Terminal block for external 12 V power supply ---
- 12 Box output for the 2G/LTE-M modem external antenna (optional)
- 13 2G/LTE-M Modem SMA Antenna
- 14 Reset button
- 15 SIM card holder

## 1.4 Supported equipment

The hub supports all equipment compliant with the 868MHz Wireless M-Bus standard and using one of the following modes: T1, S1 and T1+C1.

## 1.5 Product and accessory references

Product:

References	Descriptions
WG0612-A12	WebdynEasy W M-Bus W M-Bus




Accessories:

References	Descriptions
AC0803-01	Battery pack 7.2V / 14Ah

Webdyn proposes factory preconfiguration services for your WebdynEasy W M-Bus W M-Bus. Please contact our sales department for further information.

## 1.6 Technical Specifications

### 1.6.1 General specifications

<i>Specifications</i>	<i>Descriptions</i>
<i>Power supply</i>	+12V $\equiv$ 1A
<i>Battery</i>	7.2V 14Ah lithium battery pack (non-rechargeable) - in optimal configuration, the battery can last up to 10 years
<i>Consumption</i>	P: 2 W Pmax: 5 W
<i>Dimensions</i>	160 x 150 x 55 mm
<i>Box</i>	ASA Protected against dust and bad weather.
<i>Weight</i>	0.600 kg with battery pack 0.450 kg without battery pack
<i>Operating temperature</i>	-20 °C / +55 °C
<i>Storage temperature</i>	-20 °C / +70°C (without battery) -20 °C / +25°C (with battery)
<i>Humidity</i>	25% - 75%
<i>Pollution rating</i>	2
<i>Certification</i>	RED ROHS REACH
<i>Regulation</i>	 : CE marking created in the framework of European technical harmonisation legislation. It is mandatory for all products covered by one or more European regulatory texts (directives or regulations)
	 : Symbol indicating that the waste must be collected via a specific channel and must not be disposed of as household waste.
	 : Symbol indicating that the product must be recycled.

## 1.6.2 Technical specifications

Specifications	Descriptions
Memory capacity for data	<ul style="list-style-type: none"> <li>Flash: 1.5MB (circular operation)</li> <li>2000 frames per listening window</li> </ul>
Radio interference	Frequency: 868 MHz Antenna: Internal SMA (external as an option)
Cellular Interface Modem	GSM/GPRS/EDGE/LTE-M modem <ul style="list-style-type: none"> <li>2G (GSM, GPRS): 850/900/1800/1900 MHz</li> <li>LTE-M1: B1/B2/B3/B4/B5/B8/B12/B13/B18/B19/B20/B25/B26/B27/B28/B66/B85</li> </ul> Antenna: Internal SMA (external as an option)
BLE interface	Bluetooth 4.2 compliant
SIM format	Standard SIM (mini SIM) 2FF format 1.8V compatible



Webdyn does not supply any SIM cards. Please contact an M2M operator that supports the 2G and LTE-M network.

### Connectivity data:

RF band	Emission frequencies	Max Power
2G GSM 900	880 MHz – 915 MHz	33 dBm (class 4)
2G GSM 850	824 MHz – 849 MHz	
2G DCS 1800	1710 MHz – 1785 MHz	30 dBm (class 1)
2G PCS 1900	1850 MHz – 1910 MHz	
LTE-M - Band 1	1920 MHz – 1980 MHz	21 dBm (class 5)
LTE-M - Band 2	1850 MHz – 1910 MHz	
LTE-M - Band 3	1710 MHz – 1785 MHz	
LTE-M - Band 4	1710 MHz – 1755 MHz	
LTE-M - Band 5	824 MHz – 849 MHz	
LTE-M - Band 8	880 MHz – 915 MHz	
LTE-M - Band 12	699 MHz – 716 MHz	
LTE-M - Band 13	777 MHz – 787 MHz	
LTE-M - Band 18	815 MHz – 830 MHz	
LTE-M - Band 19	830 MHz – 845 MHz	
LTE-M - Band 20	832 MHz – 862 MHz	
LTE-M - Band 25	1850 MHz – 1915 MHz	
LTE-M - Band 26	814 MHz -849 MHz	

<i>LTE-M - Band 27</i>	807 MHz – 824 MHz	
<i>LTE-M - Band 28</i>	703 MHz – 748 MHz	
<i>LTE-M - Band 66</i>	1710 MHz – 1780 MHz	
<i>LTE-M - Band 85</i>	698 MHz – 716 MHz	
<i>BLE</i>	2402 MHz – 2480 MHz	0 dBm

### 1.6.3 Software specifications

<i>Specifications</i>	<i>Descriptions</i>
<i>Wireless M-Bus</i>	T1, S1 and T1+C1 mode
<i>BSON</i>	Specification 1.1

### 1.6.4 Battery life

Battery life may vary depending on product use and environmental conditions.

The tables below are estimates taking into account:

- standard environmental conditions
- 1 radio listening window per day
- 20KB of data recorded per listening window (ie 300 frames of 64 bytes or 500 frames of 40 bytes)

**With LTE-M:**

<i>Length of listening window in minutes</i>	<i>2 uploads/day</i>	<i>1 upload/day</i>	<i>1 upload/week</i>
2	> 15 years	> 15 years	> 15 years
4	13 years	> 15 years	> 15 years
5	12 years	14 years	> 15 years
6	11 years	13 years	13 years
8	9 years	10 years	11 years
10	8 years	9 years	9 years
12	7 years	7 years	8 years
15	6 years	6 years	6 years
20	5 years	5 years	5 years
30	3 years	3 years	3 years

**With 2G:**

<i>Length of listening window in minutes</i>	<i>2 uploads/day</i>	<i>1 upload/day</i>	<i>1 upload/week</i>
2	> 15 years	> 15 years	> 15 years
4	12 years	> 15 years	> 15 years

5	11 years	13 years	14 years
6	10 years	12 years	12 years
8	8 years	10 years	10 years
10	7 years	8 years	8 years
12	6 years	7 years	8 years
15	5 years	6 years	6 years
20	4 years	5 years	5 years
30	3 years	3 years	3 years

## 1.7 Safety instructions

Follow all the safety instructions in this guide.

Failure to follow these instructions can damage equipment and endanger people.



Electric connection:

- All wiring work must be carried out by a specialised qualified electrician.
- Please follow all the safety instructions featured in the equipment documentation.



The WebdynEasy W M-Bus W M-Bus product can be damaged by electrostatic discharges (ESD)

When the equipment is open, do not carry out any operations other than those described in this manual. Avoid any contact with the components.



Class 3 equipment: the device operates on safety extra-low voltage (SELV) (50V maximum). The voltage reduction must be obtained using a safety transformer providing safe galvanic isolation between primary and secondary.



Do not install the equipment near a heat source or at a height greater than 2m.



To clean the product, only use a slightly damp cloth to gently clean and wipe the surfaces.

Never use aggressive chemical agents or solvents that could alter the plastic material or corrode the metal parts.



Never insert a battery other than the one recommended by Webdyn.

Never recharge the battery.





To optimise radio and cellular modem reception sensitivity, it is imperative to leave 20 cm free space around the antennas.

## 1.8 Regulations

The product complies with the European directives according to the EU Declaration of Conformity available from Webdyn or on website: [www.webdyn.com](http://www.webdyn.com)

### **Recycling:**



The European directives enacted into national law covering battery waste and electric and electronic equipment provide the framework for the actions needed to limit the negative impact of the product's end of life.

These products are collected separately. Use an authorised battery collection and processing centre or contact Webdyn.

## 2 Installation

## 2.1 Prerequisite

Since the role of the WebdynEasy W M-Bus W M-Bus hub is to send the data it collects to an IS, the installation requires sufficient knowledge of the hub, but also of the information system to which it sends its data.

The following is **required** to ensure proper installation:

- To have this user manual to hand.
- To have a screwdriver suitable for the types of connectors and screws available on the WebdynEasy W M-Bus W M-Bus hub.
- To have knowledge of the parameters to connect to the IS information system.
- To have a SIM card with an activated M2M (data) subscription and knowledge of the provider's APN.
- To have a magnet.

It is also **strongly recommended** to have the elements described below for any intervention on site and to install the product.

- Use a remote antenna if radio or cellular modem reception is deteriorated.
- Have an Android smartphone with the WebdynEasy W M-Bus W M-Bus app installed to facilitate product configuration.
- Have a PC for product configuration or update via USB.

## 2.2 Unpacking

### 2.2.1 Contents

The WebdynEasy W M-Bus W M-Bus hub comes standard with:

- a curved SMA antenna for the modem (internal)
- a curved SMA antenna for the radio (internal)
- a battery pack

### 2.2.2 Identification

The commercial reference differs depending on whether or not the hub includes a customer configuration. The commercial reference is composed as follows:

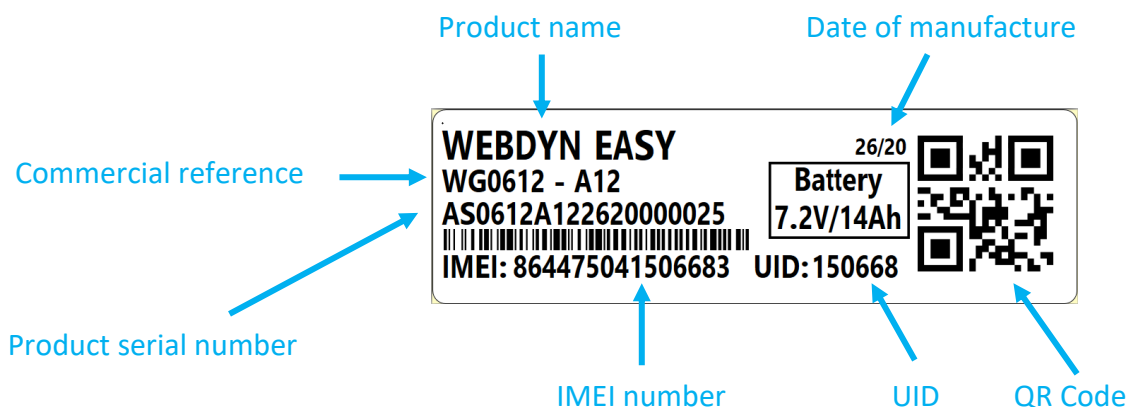
- **Without configuration:** the commercial reference is WG0612-A12.
- **With configuration:** the commercial reference is WG0612-A12 to which the customer identifier is added.

For example, the following reference can be obtained: WG0612-A12-WG

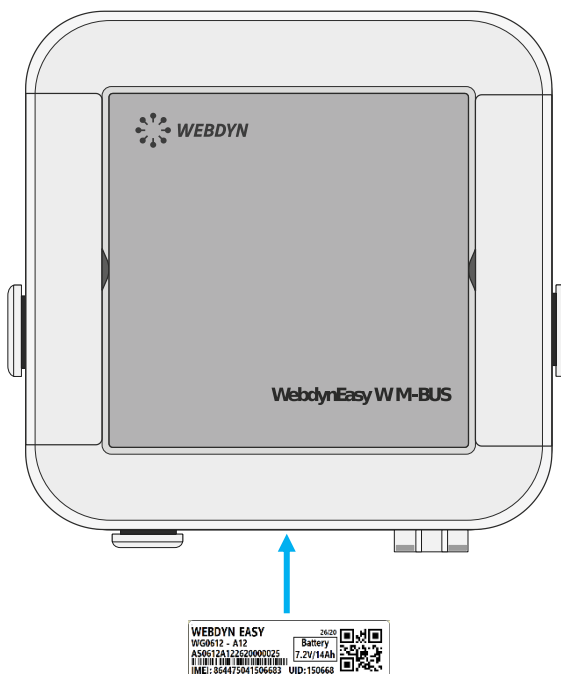
A unique identifier called **UID** is assigned to each WebdynEasy W M-Bus W M-Bus. It is used to configure the product and recognise it when you have several WebdynEasy W M-Bus W M-Bus products.

The **QR Code** is used by the mobile app for synchronisation with the hub.

Each product is labelled with the following information:



This label is accessible on the underside of the product:



## 2.3 Assembly

### 2.3.1 Opening/closing the box

***Follow these steps to open the hub box:***

If the box is wall-mounted:

1. Open the 2 doors on the front panel.
2. Unscrew the 4 wall mounting screws in the recesses under the doors.

Then follow these steps:

1. Unscrew the 4 screws behind the box.
2. Remove the cover

***Follow these steps to close the hub box:***

1. Place the cover on the box base, make sure the seal is properly fitted.
2. Screw in the 4 screws on the back of the box.

### 2.3.2 Wall mounting

The WebdynEasy W M-Bus W M-Bus can be wall-mounted. Before wall-mounting, first close the box (see section 2.3.1: “Opening/closing the box”).



Screws and anchors are not included in the kit. You must choose the correct type of screw for the type of wall you are fixing the hub to.  
(Screw with a diameter of 4mm and a length of 25mm minimum)

***Follow the steps below to fix the hub to a wall:***

1. Open the 2 doors on the front panel.
2. Screw the 4 wall mounting screws into the recesses under the doors.
3. Close both doors on the front.

## 2.4 Interface description

### 2.4.1 Product power supply

#### 2.4.1.1 External power supply

The WebdynEasy W M-Bus W M-Bus hub can be powered using 12V DC with or without a battery pack connected. Power is supplied from terminal block J8 on the left side of the battery pack.



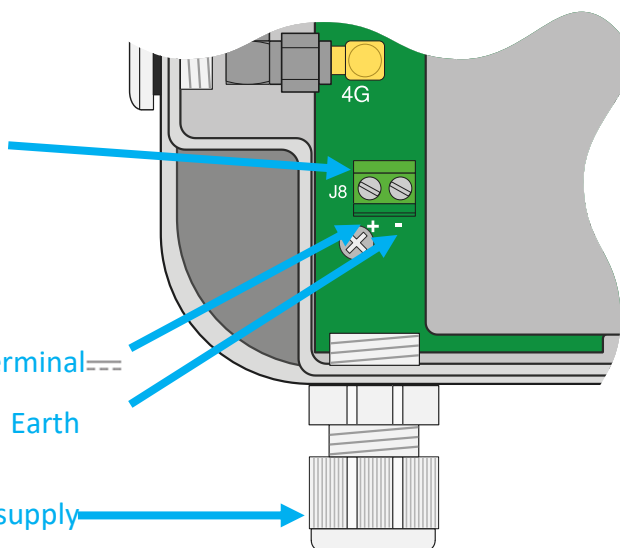
End users must use a CE certified power supply of less than 15 watts. The distance between the power supply and the product must not exceed **3 metres**. End users must make sure their installation meets applicable EMC standards.

Terminal block for  
external 12 V power  
supply ===

12 V terminal ===

Earth

External power supply  
cable gland



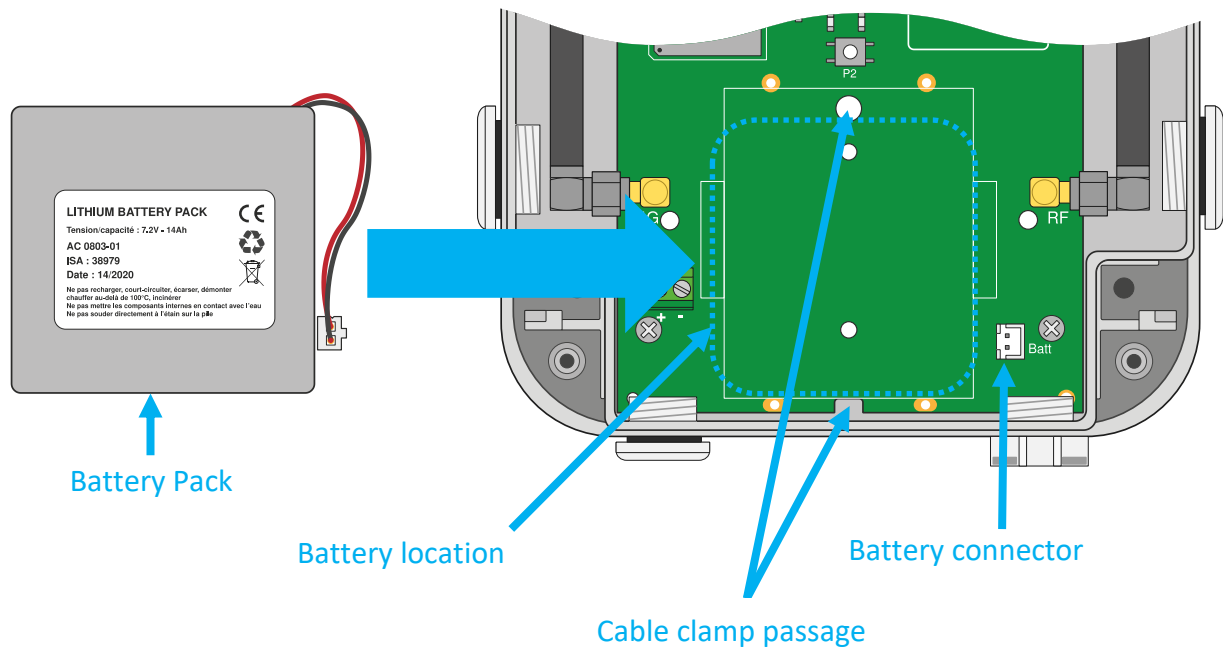
Make sure the power supply wires are connected to the proper terminals.

Product power consumption varies depending on its configuration. Make sure the power supply used can provide a minimum power of 10 watts.

#### 2.4.1.2 Battery Pack

The WebdynEasy W M-Bus W M-Bus hub has a Lithium battery that can keep the product running for 10 years. The hub includes a gauge that provides a battery consumption estimate as soon as it is installed.

The battery pack is fixed to the board in a reserved slot using a cable clamp and plugs into the connector marked "Batt" on the board.



Never install battery packs that are not new.  
Once a battery pack is installed, never disconnect it except to replace it.



CAUTION, THERE IS A RISK OF EXPLOSION IF THE BATTERY PACK IS REPLACED BY A BATTERY PACK OTHER THAN THAT RECOMMENDED BY WEBDYN.  
DISPOSE OF USED BATTERY PACKS ACCORDING TO THE INSTRUCTIONS.

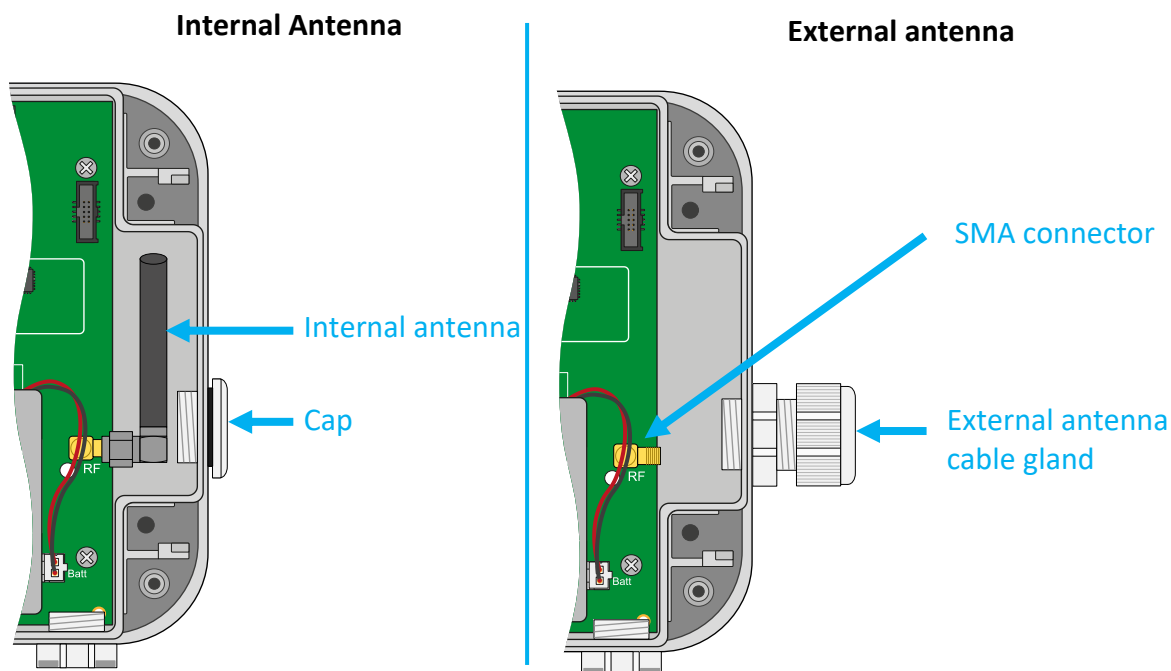
## 2.4.2 Radio

The hub has an 868MHz radio to receive Wireless M-Bus frames from the surrounding sensors.

### 2.4.2.1 Antenna

The hub has a female SMA connector labelled "RF" on the board to connect a radio antenna. The product is delivered with an internal antenna. An external antenna can be connected to the product. To do this, unscrew the cap on the box and fit a M16\*1.5 cable gland (not included).

To optimise the radio range, it is important to install the radio antenna as high as possible and to place it carefully, avoiding obstacles as far as possible. As a priority, move it away from any metal (cupboard, beams...) or concrete (reinforced concrete, walls...) obstacles as they greatly attenuate radio waves.



End users must make sure their installation using **remote antennas** meets applicable EMC standards.

### 2.4.3 Cellular Network

The WebdynEasy W M-Bus W M-Bus hub includes a 2G and LTE-M network compatible modem.

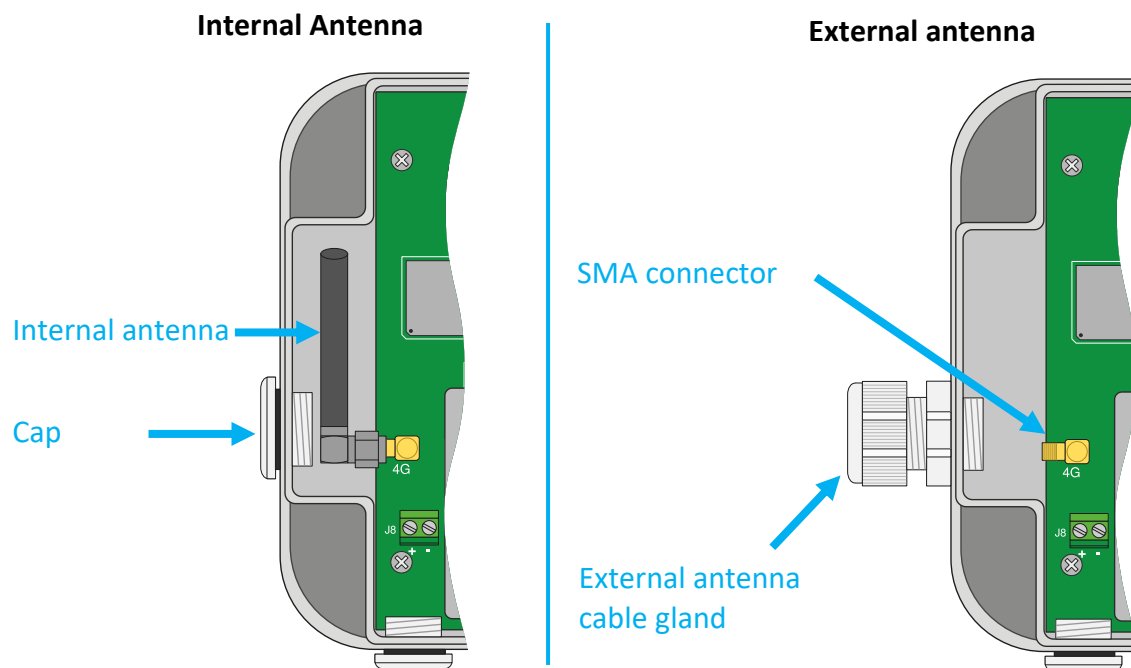
#### 2.4.3.1 Antenna

The hub has a female SMA connector labelled "4G" on the board to connect a modem antenna. The product is delivered with an internal antenna. An external antenna can be connected to the product. To do this, unscrew the cap on the box and fit a M16\*1.5 cable gland (not included).



If the WebdynEasy W M-Bus W M-Bus hub were to be installed in a metal box or in a location that does not have proper signal reception, the use of a remote antenna is strongly recommended.  
Be careful to use an antenna compatible with the connector and frequencies used.





End users must make sure their installation using **remote antennas** meets applicable EMC standards.

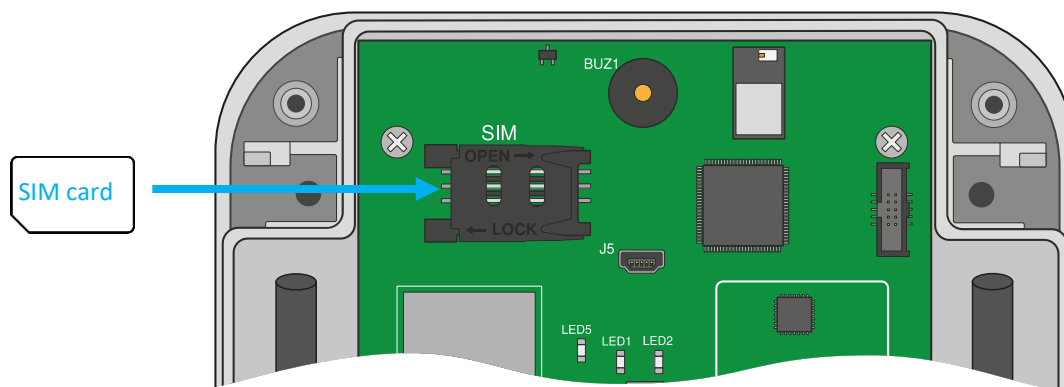
#### 2.4.3.2 SIM card

To use the LTE-M or 2G modem connection to allow the hub to communicate with the remote server, the box must be opened (see section 2.3.1: "Opening/closing the box") and a mini SIM card inserted into the SIM card housing inside the hub.

The hub is compatible with all market operators as well as with all mini SIM 2FF 25 x 15mm format SIM cards.

To check that the WebdynEasy W M-Bus W M-Bus is operating properly, insert a SIM card with the following specifications:

- 2G or LTE-M communication included.



To insert the SIM card into the product, slide the holder flap to the right (in the OPEN direction). Slide the SIM card into the flap. Then close the flap by sliding it to the left (in the LOCK direction)



Webdyn does not supply any SIM cards. Please contact an M2M operator that supports the 2G and LTE-M network.



Please contact your SIM card provider to find out what information to enter to configuration the modem.

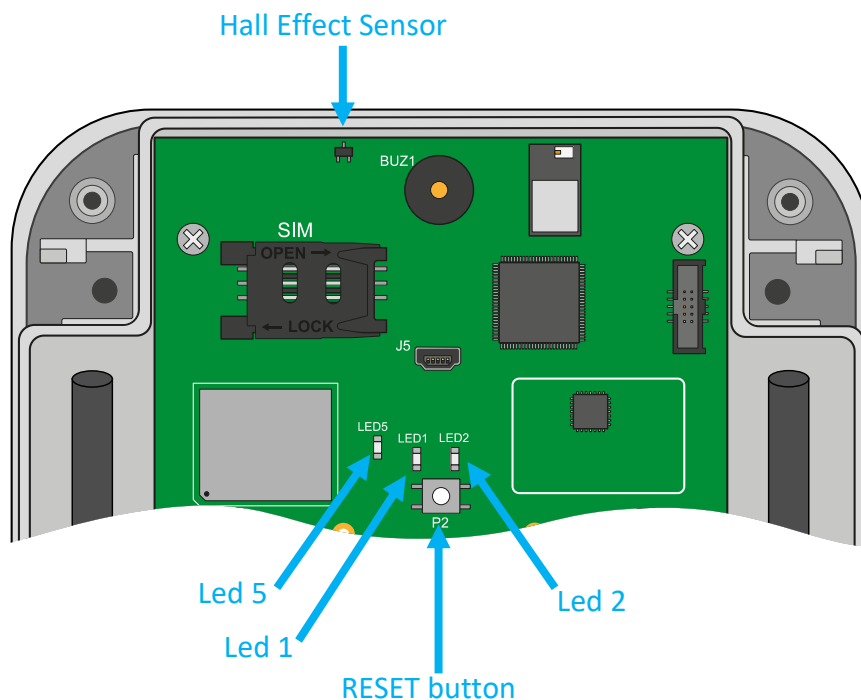


If the SIM card has an activated PIN code and it is incorrect the first time the hub is started, it will be blocked after 3 attempts. It can be unlocked using a mobile phone using the PUK code provided by the operator.

#### 2.4.4 Indicators & buttons

The hub is equipped with:

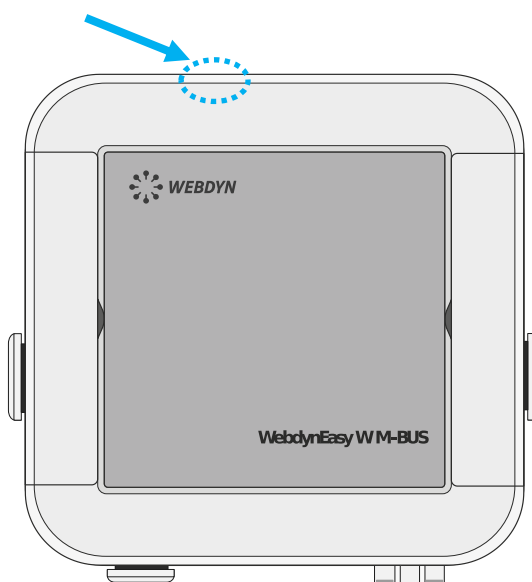
- 1 Hall Effect sensor
- 1 push button
- 1 modem indicator.



#### 2.4.4.1 Hall Effect Sensor

The WebdynEasy W M-Bus W M-Bus hub is equipped with a Hall effect sensor that activates when a magnet is passed over the top of the box.

Magnet passage area



The Hall effect sensor is used for various actions such as:

- Requesting the current operating mode and activating the interfaces for local configuration.
- Changing the operating mode.

- Re-applying factory settings (combined with the RESET button)

#### 2.4.4.2 RESET button

The RESET button has two functions:

- **Restarting** the product: a short press on the button
- **Re-applying factory settings** by following this procedure:
  1. Place the magnet on the top of the box opposite the Hall effect sensor and keep it in place.
  2. Briefly press the RESET button.
  3. A long beep (1 second) is issued, informing that the factory settings have been re-applied.
  4. Remove the magnet.

#### 2.4.4.3 Modem LED

The Modem led lights when the modem is activated to upload files to the remote server.

#### 2.4.5 USB interface

The hub USB interface can be used for:

- configuration
- commands
- product updates.

For proper operation, it is essential to strictly follow the steps below:

1. Switch the WebdynEasy W M-Bus W M-Bus to RUN mode.
2. Connect the hub to a computer using a USB cord (A male - mini B male type).
3. A new drive called "WebdynEasy W M-Bus W M-Bus" will appear on the computer (like a thumb drive)
4. Open this drive
5. Copy the configuration, command or update file to the hard drive
6. Eject the removable drive and wait for confirmation from the system.
7. Press the RESET button.
8. The product restarts.  
*The following steps are optional and are based on the commands issued.*
9. A new drive called "WebdynEasy W M-Bus W M-Bus" will appear on the computer (like a thumb drive)

10. Open this drive
11. Open the ACK file and check the result of the command  
Or  
Retrieve the files requested by the command.
12. Eject the removable drive and wait for confirmation from the system before disconnecting the USB cord.



Never disconnect the USB cord or restart the board before having the system eject the removable drive. You could damage the file system and corrupt the loaded files.



If the removable drive is inaccessible, you must format it as FAT32.

#### 2.4.6 Bluetooth interface (BLE)

The hub Bluetooth interface (BLE) is used by the "WebdynEasy W M-Bus W M-Bus" mobile app to:

- configure
- run a diagnostics command concerning its operating

The "WebdynEasy W M-Bus W M-Bus" app is available for Android and downloadable from Google Play.

## 3 Product operation

### 3.1 Starting the product

By default, the WebdynEasy W M-Bus W M-Bus hub is shipped in either:

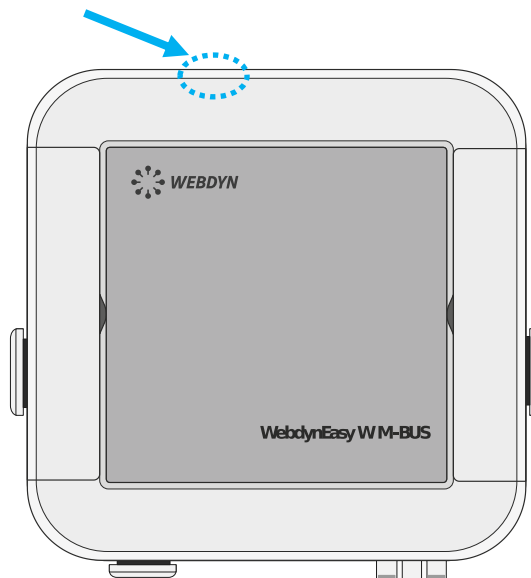
- **STORAGE** mode: if the product is delivered with a customer configuration
- **FACTORY SETTINGS** mode: if the product is delivered without a configuration.

These are low-power modes and allow products to be stored for long periods of time while limiting battery consumption.



To find out whether the WebdynEasy W M-Bus W M-Bus has a customer configuration, check the product's commercial reference. (see section 2.2.2: Identification)

Magnet passage area



To change mode and wake up the product, slowly pass a magnet over the top of the box in the indicated location (see location above) until the hub emits the first beep. The number of beeps issued by the WebdynEasy W M-Bus W M-Bus indicates the current mode.

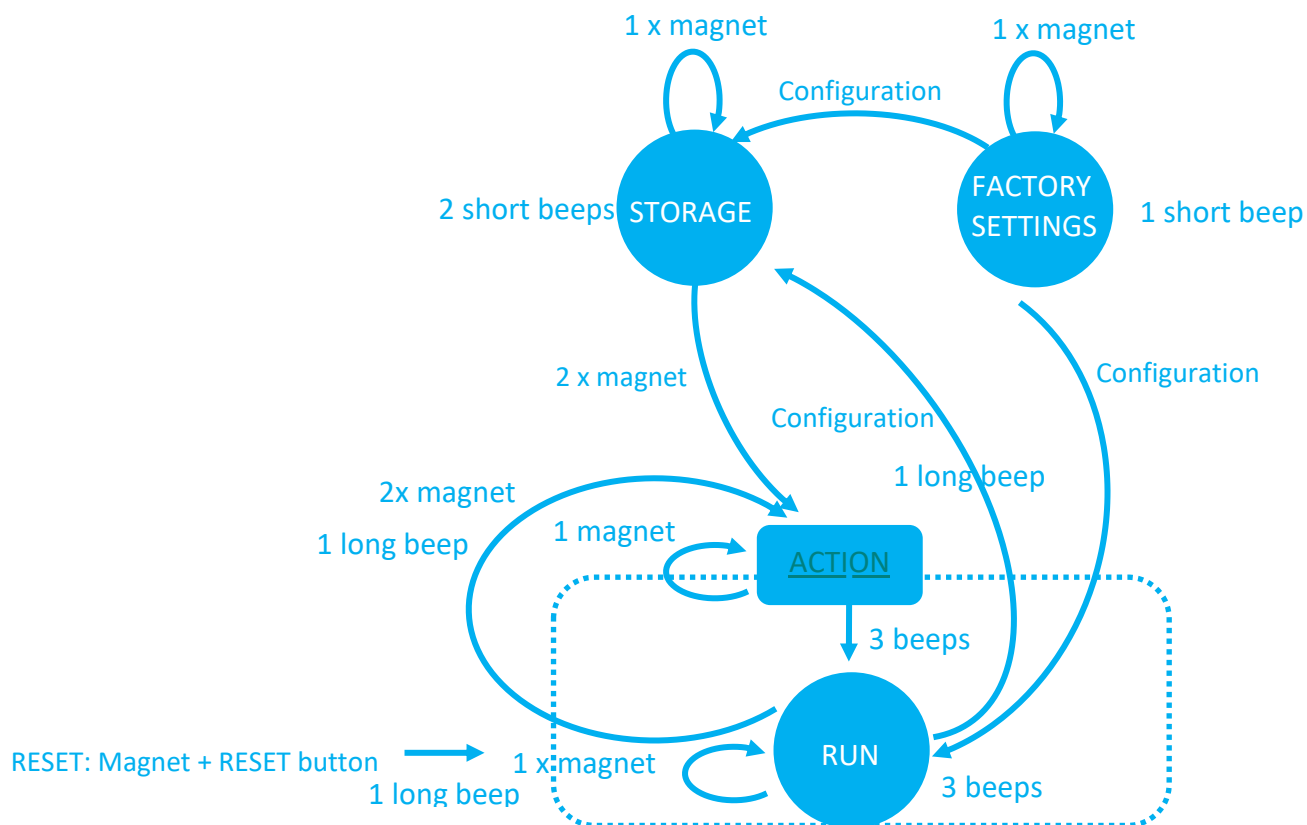
#### 3.1.1 Operating mode

The module has 3 operating modes:

- **FACTORY SETTINGS (no configuration)**
- **STORAGE (configuration)**
- **RUN**

Mode changes are **always** initiated by the installer.

The state machine below shows the 3 modes, the mode change conditions, and the beeps for each mode.



An action on the Hall Effect sensor is taken into account as soon as a magnet passes over it. For example, if a magnet passage is passed over the Hall Effect sensor, one or more beeps will sound informing of the current hub mode. When a magnet is passed a second time within 10 seconds of the first pass, the hub will switch to RUN mode.

Each mode (STORAGE, FACTORY SETTINGS and RUN) is associated with a number of beeps indicating the equipment status.

The mode can be changed by specifying it when sending a configuration to the hub.

From an installer's point of view:

- Passing a magnet once gives the current mode:
  - 1 short beep: the product is in **FACTORY SETTINGS** mode (no customer configuration).
  - 1 long beep: an action is underway, please wait for the action to end to ask for the function mode
  - 2 short beeps: the product is in **STORAGE** mode (customer configuration).
  - 3 short beeps: the product is in **RUN** mode and the last connection with the server was successful.



- 3 long beeps: the product is in **RUNmode** and the last connection with the server has failed.
- Passing a magnet twice in less than 10 seconds changes the current mode:
  - In **STORAGE** mode
    - 3 long beeps + 1 long beep: the product executes the action associated with the magnet (activation of BLE, Diagnosis command, request command) and switches to **RUNmode**.
  - In **RUNmode**
    - 1 long beep: the product executes the action associated with the magnet (activation of BLE, Diagnosis command, request command) and stays in **RUNmode**.
- Presence of a magnet + pressing the RESET button switches back to "factory settings":
  - 1 long beep: the product starts its "factory settings" process.

### 3.1.1.1 FACTORY SETTINGS mode

The product is delivered in FACTORY SETTINGS mode if it has no configuration. The hub is then on standby and its consumption is minimal. When a magnet is passed, a short beep will sound to indicate that the magnet has been detected, indicate the mode and activate the configuration.

Exiting FACTORY SETTINGS mode is only possible by configuring the product.

The hub can be configured using:

- **USB:** Connecting the USB cord automatically wakes up the product.
- **Bluetooth (BLE):** BLE is activated when the magnet is passed over the product.

To optimise battery consumption, the interfaces will automatically shut down after 4 minutes without any activity. To reactivate the interfaces, a new configuration phase will need to be launched.

The required mode can be set when configuring the product.



To find out whether the WebdynEasy W M-Bus W M-Bus has a customer configuration, check the product's commercial reference. (*see section 2.2.2: Identification*)

### 3.1.1.2 STORAGE mode

The product is delivered in STORAGE mode if it has a factory pre-configuration. The hub is then on standby and its consumption is minimal. When a magnet is passed, a short beep will sound to indicate that the magnet has been detected and indicate the mode.

Exit from STORAGE mode requires passing a magnet over the hub twice in less than 10 seconds and switches the hub to RUN mode.



To find out whether the WebdynEasy W M-Bus has a customer configuration, check the product's commercial reference. (see section 2.2.2: *Identification*)

### 3.1.1.3 RUNmode

RUN mode runs the product in its end use mode. It is used to regularly run the collection of WM-BUS data from the sensors and to connect the modem to upload the data to the server. When the product is inactive, it switches to standby mode to optimise its battery. When a magnet is passed over it, three short beeps will sound to indicate the magnet has been detected, indicate its mode and informs about the last connection with the server:

- 3 short beeps: the last connection with the server has been successful
- 3 long beeps: the last connection with the server has failed.

A second passing of the magnet within 10 seconds permits the execution of an action (activation of BLE, diagnosis command or request command) and returns to **RUNmode**.

The hub can be configured using:

- **USB:** Connecting the USB cord automatically wakes up the product.
- **Bluetooth (BLE):** BLE is activated when the magnet is passed over the product for the second time and if the preconfigured action is “Bluetooth BLE+Modem” (see chapter 4.2.1.1: “Local”).

To optimise battery consumption, the interfaces will automatically shut down after 4 minutes without any activity. A new configuration phase will need to be launched to reactivate it if necessary.

The required mode can be set when configuring the product. When the hub applies the configuration, users will be notified of the current mode by the buzzer.

In STORAGE or RUNmode, an action is started upon the second passage of the magnet over the product before it starts the RUNmode (see chapter 4.2.1.1. “Local”).

There are 3 possibilities:

- **Activation of Bluetooth BLE + Modem:** The product waits for a configuration or an update via Bluetooth or Modem within 4 minutes before starting the RUNmode and deactivating the Bluetooth and Modem.
- **Diagnosis Command:** Allows the recovery of WM-Bus data of the sensors over a configured period of time (see the radio parameters “config→radio→duration ») and starts a connection via Modem to deposit them onto the server. At the end of the diagnosis, the concentrator will inform regarding the result by:
  - 3 short beeps if everything went well
  - 3 long beeps if an error occurred.
- **Request command:** Allows the connection via Modem to deposit or recover a file (configuration, data, supervision,...) onto the FTP server. At the end of the connection, the concentrator will inform of the result by:

- 3 long beeps if the connection with the server has failed
- 3 short beeps if the connection with the server was successful.

### 3.1.1.4 RESET

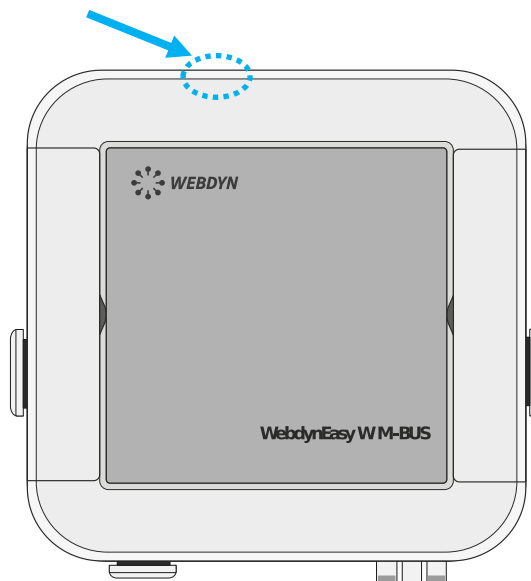
The WebdynEasy W M-Bus W M-Bus has 2 RESET functions that can be triggered using the RESET button (see section 2.4.4.2.: “RESET button”).

- **RESTART** of the product
- Hard **RESET** of the product configuration

#### Hall Effect Sensor

The WebdynEasy W M-Bus W M-Bus hub is equipped with a Hall effect sensor that activates when a magnet is passed over the top of the box.

Magnet passage area



The Hall effect sensor is used for various actions such as:

- Requesting the current operating mode and activating the interfaces for local configuration.
- Changing the operating mode.
- Re-applying factory settings (combined with the RESET button)

**RESET button** If custom settings are pre-configured at the factory, they are considered to be the default settings and are therefore applied if a factory setting mechanism is used.



To find out whether the WebdynEasy W M-Bus W M-Bus has a customer configuration, check the product's commercial reference. (*see section 2.2.2: Identification*)

### 3.1.2 Application operation

In RUN mode, the product carries out 3 tasks which are:

- WM-BUS
- Modem
- Monitoring

These tasks are regular and not synchronised with each other.

#### 3.1.2.1 WM-BUS task

The WM-BUS task is used to collect WM-BUS data from the sensors. It has its own scheduler to set its run frequency, date and time. Listening time can also be configured in the WM-BUS task.

##### Frame recording principle:

During a WM-Bus listening window, only the first frame of each size sent by a module is recorded.

Example: When a meter emits a 20 byte frame every 8 seconds and a 45 byte frame every 60 seconds. During a 2 minute listening window, only the first 20-byte frame and the first 45-byte frame are recorded.

#### 3.1.2.2 Modem task (connection to the IS)

The Modem task is used to:

- Set the hub time using NTP
- Upload and download files to and from the FTP server.

It has its own scheduler to set its run frequency, date and time. The type of file to be uploaded to the server can also be configured.

#### 3.1.2.3 Monitoring task

The Monitoring task is used to monitor:

- product temperature: temperature measurements are taken and recorded in the supervision file. The maximum temperature not to be exceeded can also be configured.
- the product battery level: the minimum battery level percentage can be defined.

If a defined threshold is exceeded, an alarm is triggered and sent to the server. For as long as a temperature alarm is active, the WM-BUS and Modem task frequencies are stopped.

The Monitoring task has its own frequency in minutes.

## 4 Configuration

The WebdynEasy W M-Bus W M-Bus hub can be configured in different ways, either by:

- Configuration file: To be uploaded to the FTP server or on the removable drive by USB connection,
- Bluetooth (BLE): Using the mobile app.

For a remote configuration with a hub delivered without a customer configuration, the cell interface and access to FTP servers must be pre-configured locally so that the hub regularly connects to the IS.

If the hub is delivered with a customer configuration, the product will retrieve its configuration directly from the FTP server without the need for a local configuration beforehand.

## 4.1 Local configuration

To configure the WebdynEasy W M-Bus W M-Bus locally, the hub must be in RUN or FACTORY SETTINGS mode. Local configuration can be by:

- USB
- Bluetooth (BLE)

### 4.1.1 USB

Connecting the USB cord to the product wakes it up. A new drive called "WebdynEasy W M-Bus W M-Bus" appears as a thumb drive. A configuration file placed on this drive is automatically taken into account when the hub is restarted.

Please follow the steps detailed in section 2.4.5: *"USB interface"*.


The configuration file is described in the section 4.2: *"Configuration file"*.

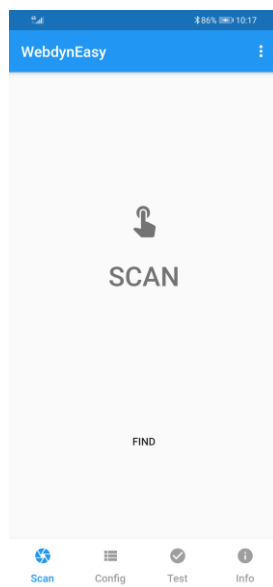
Each time a configuration file or an update file is applied, an acknowledgement file (*see section 5.5: "ACK acknowledgements"*) is placed on the hub hard drive.

### 4.1.2 Bluetooth (BLE)

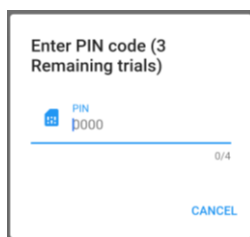
To use the local configuration with Bluetooth (BLE), an Android smartphone is required with the "WebdynEasy W M-Bus W M-Bus" app installed downloadable from Google Play ([https://play.google.com/store/apps/details?id=com.webdyn.WebdynEasy\\_W\\_M-Bus\\_W\\_M-Bus](https://play.google.com/store/apps/details?id=com.webdyn.WebdynEasy_W_M-Bus_W_M-Bus))

To synchronise the mobile app with the hub using Bluetooth, follow these steps:

- 1 Pass the magnet over the product. (see section 2.4.4.1: “Hall Effect Sensor”)
- 2 Activate Bluetooth on the smartphone.
- 3 Launch the "WebdynEasy W M-Bus W M-Bus"  smartphone app
- 4 Click "SCAN"

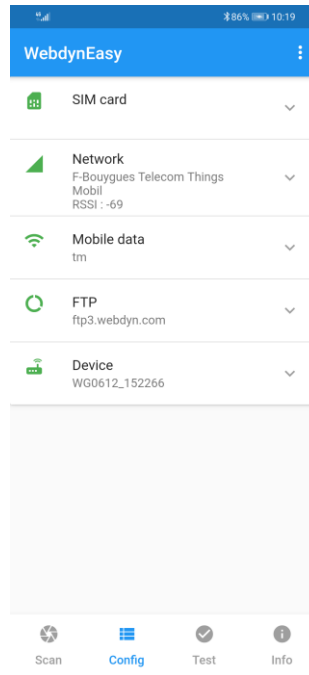


- 5 Flash the QR Code on the product label (see section 2.2.2: “Identification”).
- 6 Enter the Bluetooth Authentication PIN (by default: “1234”)



- 7 Now that you are connected to the hub, you can configure the mobile data and choose the network type as well as enter the FTP configuration. You can also run various tests such as getting the number of WM-Bus meters detected by the hub.





The hub will remain in synchronisation standby mode for 4 minutes. If that time is exceeded, the magnet will need to be passed over the product again.

## 4.2 Configuration file

The configuration files exchanged with the hub are in BSON format, which is a binary version of JSON. The format described in the document is in JSON format. To convert JSON format to BSON format or vice versa, a library available on the official website is needed: <http://bsonspec.org/implementations.html>

To avoid file alteration during exchanges, a CRC32 is included in each file.

*See section 5.1.1.4: “File Format”*

The WebdynEasy W M-Bus W M-Bus hub configuration file is in BSON format and its format is as follows:

`"<uid>-config.bson"`

### Presentation

The hub has a default configuration. It is therefore not necessary to put all the parameters in the configuration file. A configuration file can be complete or partial. A configuration file containing only one variable can therefore be sent.



JSON file parameters are case sensitive. Please respect upper and lower case characters.



It is possible to reuse the configuration file deposited by the concentrator.  
To do this, it is important to rename the “<uid> -cfg.bson” file to “<uid> -config.bson”.

### 4.2.1 System

The product operating mode, the hub name and the log level can be modified.

Parameters	Description
<i>mode</i>	Hub operating mode: <i>factorySettings</i> : FACTORY SETTINGS mode <i>storage</i> : STORAGE mode <i>run</i> : RUN mode
<i>name</i>	Hub name
<i>logLevel</i>	Event log level: 0: Error 1: Warning 2: Info (default) 3: Debug



If the functioning mode of the product is changed to FACTORY SETTINGS or STORAGE via FTP configuration, the gateway can only be put into RUN mode locally. (see chapter 3.1 “Start-up of the product”).

Example:

- **Using a JSON file** (to be converted to BSON format for the hub):

```
{
  "config":
  {
    "mode": "run",
    "name": "WebdynEasy",
    "logLevel": 2
  },
  "crc": 0
}
```

#### 4.2.1.1 Local

All the hub local variables are in the "local" object in JSON.

Parameters	Description
<i>magnet</i>	Configuration of the magnet action in RUN mode: 0: Bluetooth BLE+ Modem (default) 1: diagnosis command

	2: request command
<i>blePin</i>	Bluetooth BLE identification code (default: "1234")
<i>whiteList</i>	List of phone numbers authorised to issue commands (default: NULL, all numbers are authorised)
<i>testCount</i>	Number of diagnostic command sequences in a row between 1 and 30 (default 1)



If the number of diagnostic command sequences in a row is greater than 1, then there will be no Beep to notify of the result at the end of the diagnosis.

Example:

- **Using a JSON file** (to be converted to BSON format for the hub):

```
{
  "config":
  {
    "local":
    {
      "magnet": 1,
      "blePin": "1234",
      "whiteList": [
        { "num": "+336000000000"},
        { "num": "+8800000000000"},
      ],
      "testCount": 1
    }
  },
  "crc": 0
}
```

#### 4.2.1.2 Security

All the hub security variables are in the "security" object in JSON.

Parameters	Description
<i>crcMode</i>	BSON file CRC check: 0: No CRC check 1: Partial check (no check if CRC is 0) (default) 2: CRC check enabled
<i>encryption</i>	BSON file AES encryption: <i>true</i> : file encryption enabled <i>false</i> : file encryption disabled (default)
<i>key</i>	16-byte AES128 CBC key (to be put in Base64 in the JSON file)



If the verification of the CRC of the file (cfg, cmd, ...) fails, the reading of the file is stopped and an error is announced the acknowledgement file ACK deposited in the directory « /ALARM » of the FTP.

### **Coding in AES128 CBC:**

All the BSON files can be encrypted using an AES128 CBC with the following key as "Initial Vector" (IV):

C0 50 3E CD E3 DB 6B 2E 52 F5 9B 95 B6 F1 9B 58

The padding standard used is as follows: "PKCS#5 padding"

If the files are encrypted, a ".aes" extension is added to each BSON file.

For example, the following configuration file "123456-config.bson" becomes "123456-config.bson.aes" after enabling file encryption.

### **Example of a AES encryption conversion:**

Format	Key
Key AES128 de 16 bits in hexadecimal	44:00:AA:F7:83:78:04:9C:AB:13:EE:4E:35:0E:28:1B
Key AES128 in Base64 (JSON file)	RACq94N4BJyrE+5ONQ4oGw==

Example:

- **Using a JSON file (to be converted to BSON format for the hub):**

```
{
  "config":
  {
    "security":
    {
      "crcMode": 1,
      "encryption": true,
      "key": { "$type": "80", "$binary": "RACq94N4BJyrE+5ONQ4oGw==" }
    },
    "crc": 0
  }
}
```

## **4.2.2 WM-Bus radio**

All the hub WM-Bus radio variables are in the "radio" object in JSON.

Parameters	Description
<i>mode</i>	WM-Bus mode used: 0: for S1 1: for T1 2 : for T1 + C1 (default)

<i>duration</i>	WM-Bus listening window duration in seconds (1 to 3600) The default is: 60
<i>manufFilter</i>	List of manufacturer codes (2 bytes: M-field) authorised in binary format (type 0x00 in JSON) (maximum 8 codes)
<i>mediumFilter</i>	List of the medium codes (byte 1 of the A-field) of the authorised WM-BUS sensors or meters in binary format (type 0x00 in JSON) (max. 8 codes)
<i>whiteListAutoStop</i>	Automatic end of the listening window relative to the equipment whiteList: <i>true</i> : listening window closed after all the equipment in the whiteList is detected. <i>false</i> : listening window closed at the end of the "duration" time (default)
<i>BFormatLFieldAdaptation</i>	Modification of field L in B format WM-Bus frames (see 0)

Example:

- **Using a JSON file** (to be converted to BSON format for the hub):

```
{
  "config":
  {
    "radio":
    {
      "mode": 2,
      "duration": 60,
      "manufFilter": { "$type": "00", "$binary": "TDARpQ==" },
      "mediumFilter": { "$type": "00", "$binary": "BwE=" },
      "whiteListAutoStop": false
    }
  },
  "crc": 0
}
```

#### 4.2.2.1 White List

Sensor or meter acquisition management can be activated according to the WM-Bus identifiers entered in the white list. To do that, the " *whiteListAutoStop* " parameter must be activated by setting it to " *true* ". During a listening window, the window may close before the programmed time if the data for all the equipment in the white list has been recorded.

The white list is a BSON file that is uploaded as a configuration file for the hub. The white list accepts up to 2000 WM-Bus identifiers.

The WebdynEasy W M-Bus W M-Bus hub white list file is in BSON format and its format is as follows:

"<uid>-wlFilter.bson"

The hub white list table is in the "*wlFilter*" object, then each device is in the "*id*" object stored in binary format (type 0x00 in JSON) in JSON. The equipment contains 8 bytes for the WM-BUS sensor or meter identification and consists of:

M-field	A-field
2 bytes	6 bytes

The following can be used to remove equipment from the hub:

- either send a "whitelisterase" command (see section 5.6.7: *""whiteListErase" command"*),
- or send a configuration file with a "null" value for the "*wlFilter*" object.

Example of a white list file converted to JSON:

```
{
  "wlFilter":
  [
    { "id": { "$type": "80", "$binary": "TDDy/JtB6wE=" } },
    { "id": { "$type": "80", "$binary": "TDDX/JtB6wE=" } },
    { "id": { "$type": "80", "$binary": "TDDS/JtB6wE=" } },
    { "id": { "$type": "80", "$binary": "TDDM/JtB6wE=" } },
    { "id": { "$type": "80", "$binary": "EaWWWEBBwewc=" } },
    { "id": { "$type": "80", "$binary": "IyQRA0AQZA4=" } },
    { "id": { "$type": "80", "$binary": "IySVakAQZA4=" } },
    { "id": { "$type": "80", "$binary": "EaUXERBwewc=" } },
    { "id": { "$type": "80", "$binary": "aklIAQBBAQC=" } }
  ],
  "crc": 0
}
```

Example of a white list erasure file converted to JSON:

```
{
  "wlFilter": null,
  "crc": 0
}
```

#### 4.2.2.2 Field L for B format frames

There are two possible types of frame format for C1 mode in the WM Bus EN13757-4 standard: A and B.

- A format: field L (first byte of the frame) contains the number of bytes in the frame without the CRCs
- B format: field L (first byte of the frame) contains the number of bytes in the frame with the CRCs (2 bytes of CRC if the frame length is less than 128 bytes, otherwise 4).

The CRCs are not present in the WM Bus frames recorded by the gateway. Depending on the frame format, frame processing is not the same, so the format used by the meter must be known.

By default, the gateway modifies the B format frame L field so that it is the same as for an A format frame. Frame processing then becomes identical regardless of the format.

However, this behaviour can be disabled by applying the “false” value to the “BFormatLFieldAdaptation” parameter.

### 4.2.3 Hub connectivity

All the hub connectivity variables are in the "remote" object in JSON. They are split into 3 families:

- modem
- ftp
- ntp

#### 4.2.3.1 Modem

All the hub modem variables are in the "modem" object in JSON.

Parameters	Description
<i>mode</i>	Connection type selection: <i>2g</i> : forces the modem to 2G <i>lte</i> : Forces the modem to LTE-M <i>auto</i> : The modem manages the network automatically (default)
<i>cpin</i>	SIM card PIN code (max. 8 characters) <i>null</i> : disables the PIN code (default)
<i>apn</i>	Network Access Point (APN) identifier (32 characters max) <i>null</i> : no APN (default)
<i>use</i>	PPP connection identifier (32 characters max) <i>null</i> : no identifier (default)
<i>pass</i>	PPP connection password (32 characters max) <i>null</i> : no password (default)
<i>timeout</i>	Maximum time to connect to a network in seconds (between 60 and 1800) 300 (default)
<i>randomDelay</i>	Random wake-up time in minutes (between 0 and 60) 0: no time (default)

The variable “randomDelay” permits to randomly spread the FTP connections of all gateways with equal configuration.



When the modem mode is in auto mode, the modem connects by default in LTE-M then if this one is unavailable, it falls back to 2G.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{
  "config": {
    "remote": {
      "modem": {
        "mode": "auto",
        "cpin": null,
        "apn": null,
        "user": null,
        "pass": null,
        "timeout": 240,
        "randomDelay": 0
      }
    }
  },
  "crc": 0
}
```

#### 4.2.3.2 FTP/FTPS

All the hub FTP/FTPS variables are in the "ftp" object in JSON.

Parameters	Description
<i>addr</i>	IP address or remote FTP server name (64 characters max) (Default port: 21) The FTP port can be changed by adding ": " then the port number (between 1 and 65535)
<i>mode</i>	FTP protocol choice: 0: FTP (default) 1: FTPS (FTP over TLS)
<i>user</i>	FTP account username (32 characters max)
<i>pass</i>	FTP account password (32 characters max)
<i>dir</i>	FTP server root directory (64 characters max) <i>null</i> : FTP account root directory

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*



```
{
  "config":
  {
    "remote":
    {
      "ftp":
      {
        "mode": 0,
        "addr": "ftp.webdyn.com:60",
        "user": "webdyn",
        "pass": "1234",
        "dir": "/site1234"
      }
    }
  },
  "crc": 0
}
```

#### 4.2.3.3 NTP

The hub regularly synchronises its time with an NTP server. WebdynEasy W M-Bus W M-Bus uses the date and time in UTC+0 format.

Parameters	Description
<i>ntp</i>	IP address or NTP server name (64 characters max) (Default address: "time.google.com" and default port: 123) The NTP port can be changed by adding ":" then the port number (between 1 and 65535)



Data recording and file name generation is relative to the date and time retrieved from the NTP server in UTC+0 format. If you want to change the time zone, you will have to develop processing on the files uploaded to the FTP server.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{
  "config":
  {
    "remote":
    {
      "ntp": "time.google.com"
    }
  },
  "crc": 0
}
```

#### 4.2.4 Alarms

All the hub alarm variables are in the "alarms" object in JSON.

Parameters	Description
<i>oneAlarmPerDay</i>	Only one alarm transmission per modem per day: <i>true</i> : sends at most one alarm transmission per day (default) <i>false</i> : transmission of all alarms during a day.
<i>temperature</i>	Maximum temperature in degrees (°C) before an alarm is triggered (between 10 and 50) <i>null</i> or 0: disables the temperature alarms. (default)
<i>timeGap</i>	Monitoring of the hub clock drift compared to the time retrieved using NTP in seconds before an alarm is triggered (between 2 and 3600) <i>null</i> or 0: disables the clock drift alarms. (default)
<i>battery</i>	Monitoring of the remaining battery level in percent (%) before an alarm is triggered (between 10 and 99) <i>null</i> or 0: disables battery alarms. (default)

The temperature and battery acquisitions are made each time monitoring is triggered and when the radio Schedule or FTP Schedule is triggered (see section 4.2.5: “Schedule and”).

Example:

- **Using a JSON file** (to be converted to BSON format for the hub):

```
{
  "config":
  {
    "alarms" :
    {
      "oneAlarmPerDay" : true,
      "temperature" : null,
      "timeGap" : null,
      "battery" : 10
    }
  },
  "crc": 0
}
```

## 4.2.5 Schedule and Monitoring

The hub has:

- **A Radio Schedule:** which is used to retrieve data from WM-Bus meters and sensors,
- **An FTP Schedule:** which is used to upload files and synchronise the product clock,
- **Monitoring:** which is used to monitor the battery and the product temperature.

### 4.2.5.1 Schedules

Schedules are timer triggers that are used to schedule the running of tasks. Each Schedule can manage up to 8 timer triggers. In JSON, timer triggers are called "cron".

A "cron" timer trigger is defined as follows:

*mm hh dd MM DD*

Where:

- mm: represents the minutes (from 0 to 59)
- hh: represents the hour (from 0 to 23)
- dd: represents the day of the month (from 1 to 31)
- MM: represents the month (from 1 to 12)
- DD : represents the day in the week:
  - 0 = Sunday
  - 1 = Monday
  - 2 = Tuesday
  - ...
  - 6 = Saturday

Each field can be associated with the following value types:

Type	Example	When triggered
A specific value	"5 * * * *"	at hh:05 where hh stands for every hour (once per hour)
All values (*)	"0 * * * 1"	at hh:00 every Monday, where hh stands for every hour (24 times on Mondays)
A range (operator -)	"30 11-13 * * *"	at 11:30, 12:30 and 13:30 every day (3 times a day)
A set of values (operator , )	"5,12,47 * * * *"	at hh:05, hh:12 and hh:47 where hh stands for every hour (3 times per hour)
An interval value (operator /)	"0 */2 * * *"	at 00:00, 02:00, 04:00 and so on until 24:00 (12 times a day)



If both Schedules are triggered at the same time, the Radio Schedule will always be run in priority over the FTP Schedule.



If two timer triggers are triggered at the same time in the same Schedule, then only the first timer trigger in the file will be run.

#### 4.2.5.1.1 Radio Schedule

The Radio schedule is used to collect the data from WM-Bus meters and sensors.

All the Radio Schedule timer triggers are in the "scheduleRadio" object in JSON. The maximum number of timer triggers is 8.

Parameters	Description
<i>cron</i>	Timer trigger:

	<i>mm hh dd MM DD</i>
<i>data</i>	Radio listening window duration in seconds (between 30 and 3600)



If the "data" listening window is not entered, then the listening window duration will be the one defined in "config→radio→duration" (see section 4.2.2: "WM-Bus radio")

Example:

- **Using a JSON file** (to be converted to BSON format for the hub):

```
{
  "config":
  {
    "scheduleRadio": [
      { "cron" : "* / 30 * * * * ", "data": 30 },
      { "cron" : "15 23 * * * * ", "data": 120 }
    ],
    "crc": 0
  }
}
```

Explanation:

In the example above, we have a listening window of 30 seconds every 30 minutes and a window of 120 seconds every day at 11:15 pm.

#### 4.2.5.1.2 FTP Schedule

The FTP schedule is used to upload files to the FTP server and synchronise the product clock.

The FTP Schedule timer triggers are in the "scheduleFTP" object in JSON. The maximum number of timer triggers is 8.

Parameters	Description
<i>cron</i>	Timer trigger:  <i>mm hh dd MM DD</i>
<i>data</i>	Type of file to be sent: Bit 0: Data (1=enabled, 0=disabled) Bit 1: supervision (1=enabled, 0=disabled) Bit 2: log (1=enabled, 0=disabled)

Example:

- **Using a JSON file** (to be converted to BSON format for the hub):

```
{
  "config": {
    "scheduleFTP": [
      { "cron": "0 0 * * *", "data": 7 },
      { "cron": "0 * * * *", "data": 1 }
    ],
    "crc": 0
  }
}
```

Explanation:

In the example above, we have the data being sent every hour and the supervision and log files sent at midnight every day.

#### 4.2.5.2 Monitoring

Monitoring is a simple timer that starts at a fixed interval defined in minutes.

The hub includes battery level and product temperature monitoring, but also triggers data acquisition for statistics that will be recorded in the supervision files (see section 5.7: "Supervision").

Parameters	Description
<i>monitoringPeriod</i>	Hub monitoring period in minutes (between 15 and 1440) (60 by default) <i>null</i> or <i>0</i> : disables monitoring



If monitoring is disabled, this greatly impacts temperature and battery alarm monitoring, as temperature and battery data acquisition will only be carried out when a radio or FTP Schedule is triggered.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{
  "config":
  {
    "monitoringPeriod ": 60
  },
  "crc": 0
}
```

## 5 Operation

## 5.1 The remote server

The hub communicates with a remote server using the FTP protocol. This server is used to manage the hub remotely.

The remote server has several roles:

- **Report data and alarms collected locally by the hub:**  
Each time a connection is made to the server, whether by manual request, the triggering of an alarm or the triggering of the Connection Schedule, the hub takes advantage of the connection to the server to upload its stored data.
- **Save a copy of the configuration:**  
A backup of the hub configuration is available in the "/CONFIG" directory of the server. Each time the hub configuration is changed (locally or remotely), the hub sends a copy of its configuration to this directory.
- **Reconfigure the hub or trigger actions on it:**  
The configuration or command files must be uploaded to the server in an INBOX directory associated with the hub.
- **Monitor the hub and assist in diagnosis:**  
The hub can upload hub status files and logs for diagnostic purposes.

### 5.1.1 The FTP server

#### 5.1.1.1 Configuration

The FTP server is defined by the following parameters:

- **An address:** This can be an IP address or a domain name.
- The FTP connection port (default 21) can be changed by adding the port to be used after the ':' character to the end of the address. The format to be used is as follows: "address:port" (e.g. "192.168.1.2:8021").
- **A login and a password:** The parameters are used to define the FTP account to be used.
- **A root directory:** The root directory can be the FTP server root " / " or a series of subdirectories (for example "/WebdynEasy W M-Bus W M-Bus/123456/").

You can configure your hub remotely from your FTP server. This is only possible if your WebdynEasy W M-Bus W M-Bus hub is correctly configured to upload and synchronise its configuration on an FTP server.

#### 5.1.1.2 Server tree structure

The FTP server must have a tree structure specific to the WebdynEasy W M-Bus W M-Bus product.



Below the root directory, the FTP server must have the following directories:

Name	Rights	Description
/CONFIG	Write	Contains the configuration image. The configuration uploaded by the hub is in the following format: “<uid>-<timestamp>-config.bson”
/DATA	Write	Contains the collected data. The data file name is in the following format: “<uid>-<timestamp>-data.bson”
/ALARM	Write	Contains the alarms and acknowledgements (ACK) The alarm file name is in the following format: “<uid>-<timestamp>-alarm.bson” The acknowledgement file (ACK) name is in the following format: “<uid>-<timestamp>-ack.bson”
/SUPERVISION	Write	Contains the supervision files and the logs. The supervision file name is in the following format: “<uid>-<timestamp>-supervision.bson” The log file name is in the following format: “<uid>-<timestamp>-log.bson”
/INBOX/<uid>	Read/Write	Mailbox to send a configuration or a command to the hub.  The configuration for the hub is in the following format: “<uid>-cfg.bson” The command for the hub is in the following format: “<uid>-cmd.bson”
/BIN	Read	Contains the update files

Where:

- <uid>: Hub identifier
- <timestamp>: Number of seconds elapsed since January 1<sup>st</sup>, 1970 at midnight UTC precisely.

The minimum access rights to the different directories must be defined as specified in the table above.

If the directories on the FTP server do not exist the first time the hub connects, it will create them.



If the directories are not created at the hub connection, or if the rights are not sufficient to upload or download files, contact the server administrator.

### 5.1.1.3 Operation

The hub always uploads files to the FTP server using a 2-step process:

1. At the start of the transfer the file has an additional ".tmp" extension.
2. When the file transfer is complete, it is renamed by removing the ".tmp" extension.

This process allows the remote server to easily differentiate between files being uploaded and files that are completely uploaded.

### 5.1.1.4 File Format

The files exchanged with the remote server are in BSON format which is a binary version of JSON. The format described in the document is in JSON format. To convert JSON format to BSON format or vice versa, a library available on the official website is needed: <http://bsonspec.org/implementations.html>

To avoid file alteration during exchanges, a CRC32 is included in each file.

The "crc" field must always be the last field in the JSON and BSON files.

Example JSON configuration file:

```
{
  "config": {
    "remote": {
      "ntp": ["ntp.google.com", ""]
    }
  },
  "crc": 0
}
```

Example BSON configuration file (converted from JSON):

```
50 00 00 00 03 43 6F 6E 66 69 67 00 3A 00 00 00
03 52 65 6D 6F 74 65 00 2D 00 00 00 04 4E 54 50
00 23 00 00 00 02 30 00 0F 00 00 00 6E 74 70 2E
67 6F 6F 67 6C 65 2E 63 6F 6D 00 02 31 00 01 00
00 00 00 00 00 00 10 43 52 43 00 00 00 00 00
```

The CRC is then calculated on the whole file and the result of the CRC32 in Little Endian overwrites the 4 CRC 0 bytes. In the example, the CRC32 for the corresponding file is 0x2A0C7BA3.

Example of the BSON file with the updated CRC32:

50	00	00	00	03	43	6F	6E	66	69	67	00	3A	00	00	00
03	52	65	6D	6F	74	65	00	2D	00	00	00	04	4E	54	50
00	23	00	00	02	30	00	0F	00	00	00	00	6E	74	70	2E
67	6F	6F	67	6C	65	2E	63	6F	6D	00	02	31	00	01	00
00	00	00	00	00	00	10	43	52	43	00	A3	7B	0C	2A	00

## 5.2 The configuration

The hub allows configurations using a .

### Configuration file:

The WebdynEasy W M-Bus W M-Bus hub configuration file is in BSON format and the file name format is as follows:

`"<uid>-cfg.bson"`

A backup of the current configuration is available on the remote server in the "/CONFIG" directory. Whether after a local or remote modification of the configuration, the hub sends its new configuration to the remote server. The configuration file saved on the hub is in BSON format and the file name format is as follows:

`"<uid>-<timestamp>-config.bson"`

A configuration file can be sent locally via USB, or remotely via the FTP "INBOX" directory.

- **Locally:**

Connect the USB to the hub. Then upload the configuration file ("`<uid>-cfg.bson`").

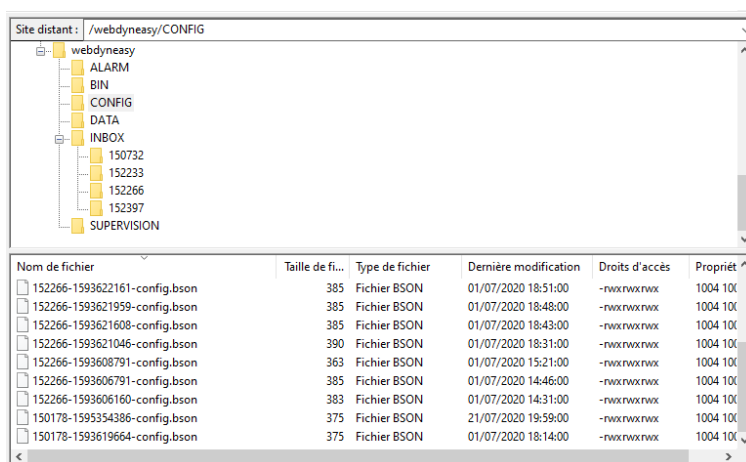
Follow the steps detailed in section 2.4.5: "USB interface".

- **Remotely:**

Upload the configuration file ("`<uid>-cfg.bson`") to the "INBOX" FTP directory for your hub ("`/INBOX/<uid>/`", with `<uid>` being your hub identifier). On the next connection to the FTP server, the hub will carry out 4 steps:

1. Download the configuration file available on the server,
2. Delete the server configuration file,
3. Apply the new configuration,

#### 4. Upload the acknowledgement file in the "ALARM" directory.



Nom de fichier	Taille de fi...	Type de fichier	Dernière modification	Droits d'accès	Propriét
152266-1593622161-config.bson	385	Fichier BSON	01/07/2020 18:51:00	-rwxrwxrwx	1004 100
152266-1593621959-config.bson	385	Fichier BSON	01/07/2020 18:48:00	-rwxrwxrwx	1004 100
152266-1593621608-config.bson	385	Fichier BSON	01/07/2020 18:43:00	-rwxrwxrwx	1004 100
152266-1593621046-config.bson	390	Fichier BSON	01/07/2020 18:31:00	-rwxrwxrwx	1004 100
152266-1593608791-config.bson	363	Fichier BSON	01/07/2020 15:21:00	-rwxrwxrwx	1004 100
152266-1593606791-config.bson	385	Fichier BSON	01/07/2020 14:46:00	-rwxrwxrwx	1004 100
152266-1593606160-config.bson	383	Fichier BSON	01/07/2020 14:31:00	-rwxrwxrwx	1004 100
150178-1593534386-config.bson	375	Fichier BSON	21/07/2020 19:59:00	-rwxrwxrwx	1004 100
150178-1593619664-config.bson	375	Fichier BSON	01/07/2020 18:14:00	-rwxrwxrwx	1004 100

A pre-defined name must be used for the configuration file "*<uid>-cfg.bson*".

Once the new configuration has been applied, an acknowledgement file containing the result of the new configuration application is uploaded to the server.

If there is an error in the configuration file (corrupt file, incorrect value, ...), the file will not be applied and the acknowledgement file reports an error.

There is no need to send the entire configuration back to your hub. A configuration file can be complete or partial. A configuration file containing only one variable can therefore be sent.

By default, the configuration sent to the hub overwrites the current configuration. Only the variables in the configuration file will be overwritten.



Refer to *section 4.2: "Configuration file"* or to "*Appendix A – Configuration – Variable list*" to see the list of variables and their possible values.

## 5.3 The data

The data is uploaded to the "/DATA" directory of the FTP server in BSON format files.

Below is the data file name format:

*<uid>-<timestamp>-data.bson*

Where:

- *<uid>*: Hub identifier
- *<timestamp>*: Number of seconds elapsed since January 1<sup>st</sup>, 1970 at midnight UTC precisely.

Example:

123456-1591083697-data.bson

The frequency at which files are sent to the remote server can be defined by an FTP Schedule (see section 4.2.5.1.2: “FTP Schedule”). However, when connecting to the server following the launch of a diagnostic, the hub takes advantage of the connection to upload the data in its memory.

The data file consists of the following elements:

Parameters	Description
<i>uid</i>	Hub identifier
<i>source</i>	Source that triggered the file upload (schedule, magnet, usb, ...)
<i>ts</i>	File creation timestamp (Number of seconds elapsed since January 1 <sup>st</sup> , 1970 at midnight UTC precisely)
<i>framecount</i>	The number of frames recorded in the <i>data</i> section
<i>data</i>	Table of recorded WM-Bus frames

The table of recorded WM-Bus frames is split up as follows:

Parameters	Description
<i>T</i>	Frame receipt timestamp (Number of seconds elapsed since January 1 <sup>st</sup> , 1970 at midnight UTC precisely)
<i>R</i>	Frame RSSI level
<i>F</i>	Content of the received frame (header + payload without CRC) in binary format. (type 0x00 in JSON) (Base64 format in the JSON file)

Example of a data file converted to JSON:

```
{
  "uid": "WE_1234",
  "source": "schedule",
  "TS": 1560068897,
  "frameCount": 2,
  "data": [
    {
      "T": 1560066602,
      "R": 199,
      "E": {
        "$type": "80",
        "$binary": "fAviGIskafDwA4E0UkO26w=="
      }
    },
    {
      "T": 15600645887,
      "R": 178,
      "E": {
        "$type": "80",
        "$binary": "+plAPliAwE0df2KBY1N7Iodr/LAFS2CrXTrUsY3wy731VN113/9UmO\CiglwR"
      }
    }
  ],
  "crc": 0
}
```

## 5.4 Alarms

Alarms are uploaded to the " /ALARM " directory on the FTP server in BSON format.

The alarm file name format is:

`<uid>-<timestamp>-alarm.bson`

Where:

- `<uid>`: Hub identifier
- `<timestamp>`: Number of seconds elapsed since January 1<sup>st</sup>, 1970 at midnight UTC precisely.

Example:

123456-1591083697-alarm.bson

Alarms can be configured to be uploaded only once per day (*see section 4.2.4: "Alarms"*).

The alarm file consists of the following elements:

Parameters	Description
<i>uid</i>	Hub identifier
<i>source</i>	Source that triggered the (alarm) file upload
<i>ts</i>	File creation timestamp (Number of seconds elapsed since January 1 <sup>st</sup> , 1970 at midnight UTC precisely)
<i>alarm</i>	Alarm Information

The "alarm" alarm information is as follows:

Parameters	Description
<i>type</i>	Alarm type: <i>temperature</i> , <i>battery</i> or <i>ntp</i>
<i>value</i>	Value of the item of data that triggered the alarm.

Example of an alarm file converted to JSON:

```
{
  "uid": "WE_1234",
  "source": "alarm",
  "TS": 1560068897,
  "frameCount": 2,
  "alarm": {
    "type": "temperature",
    "value": 45
  },
  "crc": 0
}
```

## 5.5 ACK acknowledgements

ACK acknowledgements are uploaded to the "/ALARM" directory on the FTP server as BSON format files.

The acknowledgement file name format is:

*<uid>-<timestamp>-ack.bson*

Where:

- *<uid>*: Hub identifier
- *<timestamp>*: Number of seconds elapsed since January 1<sup>st</sup>, 1970 at midnight UTC precisely.

Example:

123456-1591083697-ack.bson

Acknowledgements are sent following a command received by the hub.

The acknowledgement file consists of the following elements:

Parameters	Description
<i>uid</i>	Hub identifier
<i>source</i>	Source that triggered the file upload (ftp, usb, ...)
<i>ts</i>	File creation timestamp (Number of seconds elapsed since January 1 <sup>st</sup> , 1970 at midnight UTC precisely)
<i>ack</i>	Command acknowledgement

The "ack" command acknowledgement consist of the following:

Parameters	Description
<i>type</i>	Command type corresponding to the acknowledgement <i>cmd</i> : command <i>config</i> : configuration update
<i>cid</i>	Command identifier corresponding to the acknowledgement
<i>result</i>	Command result: <i>ok</i> : command or update successfully processed <i>ko</i> : error processing the command or the update
<i>data</i>	Data associated with the acknowledgement (for example: the new firmware name, the configuration file name, ...)

Example of an acknowledgement file converted to JSON:

```
{
  "uid": "WE_1234",
  "source": "ftp",
  "TS": 1560068897,
  "ack": {
    "type": "cmd",
    "cid": "c_1234",
    "result": "ok",
    "data": "/f10/152233-cmd.bson"
  },
  "crc": 0
}
```

## 5.6 Commands

Actions can be run on the hub remotely. To do this, the hub must be send a command. This command can be sent using a BSON command file.

- BSON command file:

The command file should be named as follows:

*<uid>-cmd.bson*

The command file can be uploaded by:

- **FTP** to the remote server “/INBOX” directory for the hub (“/INBOX/<uid>/”, where <uid> is the hub identifier). In the same way as the configuration files. All the files in this directory will be downloaded before being deleted and run. An acknowledgement file will be uploaded to the remote server notifying of the result of the command processing.
- **USB** on the removable drive. After the product is restarted (for example by a simple press on the RESET button), the files on the removable drive will be processed and an acknowledgement will be uploaded to the drive notifying of the result of the command.



All commands accept two optional parameters "uid" and "cid":

- uid: unique hub identifier
- cid: command identifier

Commands will be rejected if the included uid parameter does not match the hub uid. The cid can be freely chosen by the command issuer. It will be included with any associated download.

A command acknowledgement is created and uploaded for the:

- FTP: in the ALARM directory of the remote server,
- USB: on the hub removable drive.

Below is a list of the commands available on the hub:

Command	Description
request	Immediate connection to the remote server
factory	Back to factory settings
update	Hub software update
status	Hub status retrieval
diag	Launch a duration
configGet	Force the upload of the config to the remote server
whiteListErase	Erase the radio white list
operatorInit	Reset the modem network operator selection
logGet	Force logs to be uploaded to the remote server
logClean	Erase the event log
supervisionClean	Resets counters to 0



If several command files are sent at the same time, the "factory" and "update" commands can result in the commands following them being lost. Also, if there is an error on a previous command, the following commands will not be run.

Example:

- **Using a JSON file** (to be converted to BSON format for the hub):

```
{
  "cmd": {
    "type": "request",
  },
  "crc": 0
}
```

### 5.6.1 “Request” command

The "request" command triggers the immediate connection of the product to the remote server, making it possible to upload data, the configuration and the supervision, but also to download the files present in its INBOX.

No subcommands or parameters are required for this command.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{  
  "cmd": {  
    "type": "request",  
  },  
  "crc": 0  
}
```

### 5.6.2 “Factory” command

The "factory" command is used to restore the hub factory settings.

No subcommands or parameters are required for this command.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{  
  "cmd": {  
    "type": "factory",  
  },  
  "crc": 0  
}
```

### 5.6.3 “Update” command

The "update" command is used to update the hub software.

*(see section 6: “Update”)*

### 5.6.4 “Status” command

The "status" command is used to retrieve product status information. When the request is made using a file, a status file is uploaded to the remote server "/SUPERVISION" directory.

No subcommands or parameters are required for this command.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{
  "cmd": {
    "type": "status",
    "cid": "status cmd 1"
  },
  "crc": 0
}
```

### 5.6.5 “Diag” command

The "diag" command can be used to trigger a hub diagnosis by collecting the data sent by the WM-Bus sensors over a configured period of time (see the « config→radio→duration » radio settings) and then launching a connection to the remote server, thus making it possible to upload data, the configuration and the supervision, but also to download the files from in its INBOX.

No subcommands or parameters are required for this command.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{
  "cmd": {
    "type": "diag",
    "uid": "123456"
  },
  "crc": 0
}
```

### 5.6.6 “ConfigGet” command

The "configGet" command forces the current configuration to be uploaded to the remote server in the "/CONFIG" directory.

No subcommands or parameters are required for this command.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{  
  "cmd": {  
    "type": "configGet"  
  },  
  "crc": 0  
}
```

### 5.6.7 “whiteListErase” command

The "whiteListErase" command is used to erase the WM-Bus radio white list.  
No subcommands or parameters are required for this command.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{  
  "cmd": {  
    "type": "whiteListErase"  
  },  
  "crc": 0  
}
```

### 5.6.8 “operatorInit” command

The "operatorInit" command is used to reset the modem network operator.  
No subcommands or parameters are required for this command.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{  
  "cmd": {  
    "type": "operatorInit"  
  },  
  "crc": 0  
}
```

### 5.6.9 “logGet” command

The "logGet" command is used to force the upload of the logs to the remote server in the  
"/SUPERVISION" directory.  
No subcommands or parameters are required for this command.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{  
  "cmd": {  
    "type": "logGet"  
  },  
  "crc": 0  
}
```

### 5.6.10 "logClean" command

The "logClean" command is used to clear the log files.  
No subcommands or parameters are required for this command.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{  
  "cmd": {  
    "type": "logClean"  
  },  
  "crc": 0  
}
```

### 5.6.11 "supervisionClean" command

The "supervisionClean" command is used to reset the hub counter values to 0.  
No subcommands or parameters are required for this command.

Example:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{  
  "cmd": {  
    "type": "supervisionClean"  
  },  
  "crc": 0  
}
```

## 5.7 Supervision

The supervision information is uploaded to the "/SUPERVISION" directory on the FTP server in BSON format.

Below is the supervision file name format:

<uid>-<timestamp>-supervision.bson

Where:

- **<uid>**: Hub identifier
- **<timestamp>**: Number of seconds elapsed since January 1<sup>st</sup>, 1970 at midnight UTC precisely.

Example:

123456-1591083697-supervision.bson

Supervision files are uploaded when connecting to the FTP server.

The supervision file consists of the following elements:

Parameters	Description
<i>uid</i>	Hub identifier
<i>source</i>	Source that triggered the file upload (schedule, ftp, usb, ...)
<i>TS</i>	File creation timestamp (Number of seconds elapsed since January 1 <sup>st</sup> , 1970 at midnight UTC precisely)
<i>supervision</i>	Supervision information

The information from the "supervision" supervision is as follows:

Section	Parameters	Description
<i>identity</i>	<i>uid</i>	Hub identifier
	<i>name</i>	Hub name
	<i>SN</i>	Serial number
<i>version</i>	<i>hw</i>	Hardware Version
	<i>sw</i>	Software version of the application
	<i>swBle</i>	Software version of the Bluetooth BLE module
	<i>swModem</i>	Software version of the Modem
<i>uptime</i>	<i>run</i>	Total operating time in RUN mode in seconds
	<i>radio</i>	Total radio listening time in seconds
	<i>modem</i>	Total modem time in seconds
	<i>lowPower</i>	Total standby time in seconds
<i>measures</i>	<i>vAlim</i>	Supply voltage in mV
	<i>temperature</i>	Hub temperature in °C
	<i>histogram</i>	Temperature bar chart
<i>modem</i>	<i>rsi</i>	RSSI for the last connection
	<i>imei</i>	Modem IMEI
	<i>num</i>	SIM card phone number
	<i>operator</i>	Preferred operator name
	<i>mode</i>	Connection type (2G, LTEM, auto)
<i>counters</i>	<i>wakeup</i>	Total number of wake-ups
	<i>magnetWakeup</i>	Total number of wake-ups from the magnet
	<i>bleWakeup</i>	Total number of wake-ups from the Bluetooth BLE
	<i>ftp</i>	Total number of wake-ups by the FTP Schedule
	<i>alarms</i>	Total number of alarms sent

	<i>radio</i>	Total number of wake-ups by the radio Schedule
	<i>usb</i>	Total number of wake-ups by USB
	<i>wd</i>	Number of watchdog triggers
	<i>fault</i>	Number of fatal errors
<i>dates</i>	<i>lastConfig</i>	Timestamp of the last configuration update (optional)
	<i>lastUpdate</i>	Timestamp of the last software update (optional)
	<i>lastPowerOn</i>	Timestamp of the last power-up (optional)
	<i>lastCnxOk</i>	Timestamp of the last successful connection (optional)
	<i>lastCnxKo</i>	Timestamp of the last connection failure (optional)
<i>battery</i>	<i>remainTime</i>	Estimated remaining battery life in months ( <i>null</i> : impossible to estimate the duration as a minimum of 1 month of use is required before the first estimate)
	<i>capaPrcent</i>	Percentage of remaining battery capacity
	<i>capaUsed</i>	Number of mAh used on the battery
<i>WmBus</i>	<i>CptRxPreamb</i>	Number of radio preambles detected in radio reception
	<i>CptRxSync</i>	Number of radio synchronization codes detected during radio reception
	<i>CptRxFrames</i>	Number of radio frames decoded in radio reception
	<i>CptRxFramesCrcKo</i>	Number of CRC errors in radio reception
	<i>CptRxFramesDecodeKo</i>	Number of decoding errors in radio reception
	<i>CptRxFramesLenKo</i>	Number of length errors in radio reception
	<i>CptRxOverrun</i>	Number of buffer overruns in radio reception
	<i>CptRxTimeOut</i>	Number of times exceeded in radio reception
	<i>CptRecFrames</i>	Number of recorded frames



The “WM Bus” statistical data is intended for Webdyn support.



Sometimes the “num” phone number is empty because some mobile operators do not enter the phone number on the SIM. This does not impact hub operation.

The temperature “*histogram*” bar chart is as follows:

Parameters	Description
<i>below -10°C</i>	Number of times the hub temperature was below -10°C
<i>-10°C to 0°C</i>	Number of times the hub temperature was between -10°C and 0°C
<i>0°C to 10°C</i>	Number of times the hub temperature was between 0°C and 10°C
<i>10°C to 20°C</i>	Number of times the hub temperature was between 10°C and 20°C
<i>20°C to 30°C</i>	Number of times the hub temperature was between 20°C and 30°C
<i>30°C to 40°C</i>	Number of times the hub temperature was between 30°C and 40°C
<i>above 40°C</i>	Number of times the hub temperature was above -40°C

Example of a supervision file converted to JSON:





```
{
  "uid": "WE_1234",
  "source": "schedule",
  "TS": 1598349735,
  "supervision": {
    "identity": {
      "uid": "WE_1234",
      "name": "1234",
      "SN": "SN20200423_9876543210"
    },
    "version": {
      "hw": 11,
      "sw": "01.04",
      "swBle": "0.1",
      "swModem": "BG95M3LAR02A03_01.005.01.005"
    },
    "uptime": {
      "run": 85484,
      "radio": 69090,
      "modem": 5873,
      "lowPower": 336583
    },
    "measures": {
      "vAlim": 7100,
      "temperature": 15,
      "histogram": {
        "below -10°C": 0,
        "-10°C to 0°C": 0,
        "0°C to 10°C": 63,
        "10°C to 20°C": 73,
        "20°C to 30°C": 36,
        "30°C to 40°C": 0,
        "above 40°C": 0
      }
    },
    "modem": {
      "rssi": -57,
      "imei": "864475041522664",
      "num": "",
      "operator": "Orange F Things Mobile",
      "mode": "LTEM"
    },
    "counters": {
      "wakeup": 2281,
      "magnetWakeup": 4,
      "bleWakeup": 0,
      "ftp": 137,
      "radio": 2094,
      "alarms": 0,
      "usb": 14,
      "wd": 0,
      "fault": 0
    },
    "dates": {
      "lastConfig": 1598364963,
      "lastUpdate": 0,
      "lastPowerOn": 0,
      "lastCnxOk": 1610968521,
      "lastCnxKo": 1610752526
    },
    "battery": {
      "remainTime": null,
      "capaPrcent": 95,
      "capamAh": 13362
    },
    "WmBus": {
      "CptRxPreamb": 18443,
      "CptRxSync": 11639,
      "CptRxFrames": 5632,
      "CptRxFramesCrcKo": 221,
      "CptRxFramesDecodeKo": 5786,
      "CptRxFramesLenKo": 0,
      "CptRxOverrun": 0,
      "CptRxTimeOut": 0,
      "CptRecFrames": 253
    }
  },
  "crc": 0
}
```



## 5.8 The Log

The logs are uploaded to the " /SUPERVISION " directory on the FTP server, in BSON format.

Below is the supervision file name format:

`<uid>-<timestamp>-log.bson`

Where:

- `<uid>`: Hub identifier
- `<timestamp>`: Number of seconds elapsed since January 1<sup>st</sup>, 1970 at midnight UTC precisely.

Example:

`123456-1591083697-log.bson`

Log files are uploaded when connecting to the FTP server.

The log file consists of the following elements:

Parameters	Description
<i>uid</i>	Hub identifier
<i>source</i>	Source that triggered the file upload (schedule, ftp, usb, ...)
<i>TS</i>	File creation timestamp (Number of seconds elapsed since January 1 <sup>st</sup> , 1970 at midnight UTC precisely)
<i>log</i>	List of logs

The list of "logs" is as follows:

Parameters	Description
<i>T</i>	Event timestamp (Number of seconds elapsed since January 1 <sup>st</sup> , 1970 at midnight UTC precisely)
<i>C</i>	Event Code (see "Appendix B - Log - Event List")
<i>D</i>	Event data (optional)

### Example of a log file converted to JSON:

```
{
  "uid": "WE_1234",
  "source": "schedule",
  "TS": 1560068897,
  "log": [
    {
      "T": 1598348100,
      "C": 33554432,
      "D": "[APP][appModeInit():133]mode: run"
    },
    {
      "T": 1598348100,
      "C": 33554432,
      "D": "[APP][appRun():27]run application"
    },
    {
      "T": 1598348100,
      "C": 3,
      "D": "[RADIO][appRadioTask():57]radio task"
    },
    {
      "T": 1598348100,
      "C": 3,
      "D": "[FRAME][Frame_FastFilter_Init():57]white list filter disable"
    },
    {
      "T": 1598348100,
      "C": 33554432,
      "D": "[RADIO][appRadioTask():98]Windowsduration: 30"
    },
    {
      "T": 1598348130,
      "C": 33554432,
      "D": "[FRAME][Frame_Process_Cmd():226]RX Frame count:5"
    },
    {
      "T": 1598348134,
      "C": 3,
      "D": "[PERIODIC][periodicTask():49]periodic task"
    },
    {
      "T": 1598348134,
      "C": 33554432,
      "D": "[RADIO][appRadioNextAlarm():144]Next Alarm: Tue Aug 25 09:40:00 2020\n data:30"
    }
  ],
  "crc": 0
}
```

## 6 Update

The WebdynEasy W M-Bus W M-Bus hub can be updated locally via USB or remotely via FTP. The product has 2 firmwares. One for the application part and the other for Bluetooth BLE management.

The latest version of the firmware ("WebdynEasy W M-Bus W M-Bus\_Vx.x.bson" and "WebdynEasy W M-Bus W M-Bus\_BLE\_Vx.x.bson") as well as the associated command are available for download on our site at the following address:  
<https://www.webdyn.com/en/support/webdyneasy/>

To update the hub, an "update" command must be issued in addition to the new firmware. In the command, the name of the firmware to be taken into account must be indicated in the "data" field.

A "firmware-cmd.bson" command file is supplied along with the firmware. Just rename it to "<uid>-cmd.bson".

With:

- <uid>: Username of the concentrator

The application firmware and Bluetooth (BLE) can be updated at the same time. In that case, the first update command must be for the Bluetooth (BLE) firmware.

An acknowledgement file is uploaded after the update.

Example of an application and Bluetooth (BLE) firmware update:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{
  "cmd": [
    {
      "type": "update",
      "cid": "firmwareBLE",
      "data": "WebdynEasy_BLE_V1.0.bson"
    },
    {
      "type": "update",
      "cid": "firmwareAPP",
      "data": "WebdynEasy_V1.0.bson"
    }
  ],
  "crc": 0
}
```

Example of an application firmware update:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{
  "cmd": {
    "type": "update",
    "data": "WebdynEasy_V1.0.bson"
  },
  "crc": 0
}
```

Example of Bluetooth (BLE) firmware update:

- **Using a JSON file** *(to be converted to BSON format for the hub):*

```
{
  "cmd": {
    "type": "update",
    "data": "WebdynEasy_BLE_V1.0.bson"
  },
  "crc": 0
}
```

## 6.1 Local

To update the hub locally, connect the USB cord between the hub and a computer and then follow the file download procedure *(see section 4.1.1: "USB")*.

The file or files containing the update and the associated command must be uploaded.

## 6.2 Remote

For a remote update, the file containing the update must be uploaded to the "/BIN" directory on the remote server, and an "update" command must be uploaded to the "INBOX" FTP directory for your hub ("/INBOX/<uid>/", with <uid> your hub identifier).

The command must include the name of the file containing the update ("data" field).

At the next FTP server connection, the hub will retrieve the files and run the update.

## 7 Tools & diagnostics

### BSON-JSON and JSON-BSON conversion tools:

To facilitate the conversion of JSON and BSON files, Webdyn has created a JSON-BSON and BSON-JSON converter. It can be used in silent mode (using a console) or through a HMI.

This tool is available free of charge for Windows and Linux from our website at this address:

<https://www.webdyn.com/download/JsonBsonConverter.zip>

### Example of Python Conversion Scripts:

An application note with concrete examples of scripts is available which allows you to convert and manipulate files.

It is available at the following address:

<https://www.webdyn.com/en/support>



## 8 Support

If there are technical problems related to our products, contact WEBDYN support:

**Webdyn SA**

26 Rue des Gaudines  
78100 Saint-Germain-en-Laye  
Phone: +33 1 39 04 29 40  
email [support@webdyn.com](mailto:support@webdyn.com)  
<https://www.webdyn.com>

Please have the following elements to hand:

- Product type
- Product serial number.
- Product hardware and software version.
- Hub logs
- Hub configuration



The user manual and firmware are available at this web address:  
<https://www.webdyn.com/support>

## 9 FAQ

**Why has the hub stopped uploading files to the FTP server?**

Please check these items in this order:

- the battery level: if the battery level is too low or empty, the product will not run properly or not run at all.
- Modem reception level: a bad signal at the modem may also prevent the hub from uploading files. Look to move the product or install an external antenna to improve signal quality.
- The last configuration file: a bad configuration file can block the product.

**How do I know if the product is started?**

Remotely, by checking the regularly uploaded files if the product configuration is correct.

On site, by passing the magnet over the product, you will hear 3 short beeps.

**How to replace a product by another?**

Replace the product and inject the configuration from the old product into the new one. If a white list is used, remember to inject it into the new product as well.

**Can the concentrator decrypt the encrypted data received from WM-Bus equipment?**

No, the concentrator is not able to decrypt data from WM-BUS equipment because it does not have a safe on board to guarantee the security of the encryption keys of your equipment. The recovered data is deposited without modification (without decryption) by the concentrator on your remote server.

## 10 APPENDICES

## 10.1 Appendix A: Configuration - Variable list



All the "names +tree structures" highlighted in blue are lists and can be created multiple times.

Name + tree structure	Description	Type	Default value	Information only
/version	Application version	Integer		•
/timestamp	Timestamp of the last configuration (Number of seconds elapsed since 1st January 1970 at midnight UTC precisely)	Text		•
/uid	Hub identifier	Whole number		•
/name	Hub name	Text		
/sn	Serial number			•
/mode	Hub operating mode	List: • factorySettings • storage • run		
/logLevel	Log level in the event log	List: • 0: Error • 1: Warning • 2: Info • 3: Debug	2	
/radio/mode	WM-Bus mode used	List: • 0: for S1 • 1: for T1 • 2: for T1+ C1	2	
/radio/BFormatLFieldAdaptation	Activation of the L field adaptation for B format frames	Boolean	true	
/radio/duration	Duration of the WM-Bus listening window in seconds	Integer (min 1 max 3600)	60	
/radio/manuffFilter	List of authorised manufacturer codes (maximum 8 codes)	List: 2 bytes in binary format (M-field)		
/radio/mediumFilter	List of authorised WM-BUS sensor or meter medium codes (maximum 8 codes)	List: byte 1 of the A-field in binary format		
/radio/whiteListAutoStop	Automatic end of the listening window relative to the equipment whiteList	Boolean: • true: listening window closed after detection of all the equipment in the whiteList. • false: listening window closed at the end of the "duration" time	false	
/security/crcMode	BSON file CRC check	List: • 0: No CRC check • 1: Partial check (no check if the CRC is 0) • 2: CRC check enabled	1	
/security/encryption	BSON file AES encryption	Boolean: • true: file encryption enabled	false	

		<ul style="list-style-type: none"> <li>false: file encryption disabled</li> </ul>		
/security/key	AES128 CBC Key	Hexadecimal 16 bytes	<b>null</b>	
/local/magnet	Configuration of the magnet action in RUN mode	List: <ul style="list-style-type: none"> <li>0: Bluetooth BLE + Modem</li> <li>1: diagnosis</li> <li>2: request</li> </ul>	<b>0</b>	
/local/blePin	Identification code for Bluetooth BLE	4-digit text	<b>1234</b>	
/local/whiteList	List of phone numbers authorised to send commands (maximum 4)	Text up to 16 digits	<b>null</b>	
/local/testCount	Number of diagnostic sequences in a row	Integer (min 1 max 30)	<b>1</b>	
/remote/modem/mode	Connection type selection	List: <ul style="list-style-type: none"> <li>2G: forces the modem to 2G</li> <li>LTEM: Forces the modem to LTE-M</li> <li>auto: The modem manages the network automatically</li> </ul>	<b>auto</b>	
/remote/modem/cpin	SIM card PIN code	Text (8 characters max)	<b>null</b>	
/remote/modem/apn	Network access point (APN) identifier	Text (32 characters max)	<b>null</b>	
/remote/modem/user	PPP connection login	Text (32 characters max)	<b>null</b>	
/remote/modem/pass	PPP connection password	Text (32 characters max)	<b>null</b>	
/remote/modem/timeout	Maximum time to connect to a network in seconds	Integer (min 60 max 1800)	<b>300</b>	
/remote/modem/randomDelay	Random wake-up time in minutes	Integer (min 0 max 60) 0: no delays	<b>0</b>	
/remote/ftp/mode	FTP protocol choice	List: <ul style="list-style-type: none"> <li>0: FTP</li> <li>1: FTPS (FTP over TLS)</li> </ul>	<b>0</b>	
/remote/ftp/addr	IP address or remote FTP server name The FTP port can be changed by adding ": " then the port number (between 1 and 65535)	Text (64 characters max)	<b>null</b>	
/remote/ftp/user	FTP account login	Text (32 characters max)	<b>null</b>	
/remote/ftp/pass	FTP account password	Text (32 characters max)	<b>null</b>	
/remote/ftp/dir	FTP server root directory	Text (64 characters max) null: FTP account root directory	<b>null</b>	
/remote/ntp	IP address or NTP server name The NTP port can be changed by adding ": " then the port number (between 1 and 65535)	Text (64 characters max)	<b>null</b>	
/alarms/oneAlarmPerDay	Only one alarm transmission per modem per day	Boolean: <ul style="list-style-type: none"> <li>true: sends at most one alarm transmission per day (default)</li> <li>false: transmission of all alarms during a day.</li> </ul>		
/alarms/temperature	Maximum temperature in degrees (°C) before an alarm is triggered	Integer (min 10 max 50) null or 0 : disables the temperature alarms.	<b>null</b>	
/alarms/timeGap	Monitoring of the hub clock drift compared to the time retrieved using NTP in seconds before an alarm is triggered	Integer (min 2 max 3600) null or 0 : disables the clock drift alarms.	<b>null</b>	
/alarms/battery	Monitoring of the remaining battery level in percent (%) before an alarm is triggered	Integer (min 10 max 99) null or 0 : disables battery alarms.	<b>null</b>	
/scheduleRadio	List of radio Schedule timer triggers			
/scheduleRadio/cron	Timer trigger	"cron" timer trigger: mm hh dd MM DD		

		<ul style="list-style-type: none"> <li>• <i>mm</i>: represents the minutes (from 0 to 59)</li> <li>• <i>hh</i>: represents the hour (from 0 to 23)</li> <li>• <i>dd</i>: represents the day of the month (from 1 to 31)</li> <li>• <i>MM</i>: represents the month (from 1 to 12)</li> <li>• <i>DD</i> : represents the day in the week: <ul style="list-style-type: none"> <li>▪ 0 = Sunday</li> <li>▪ 1 = Monday</li> <li>▪ = Tuesday</li> <li>▪ ...</li> <li>▪ 6 = Saturday</li> </ul> </li> </ul>		
/scheduleRadio/data	Duration of the Radio listening window in seconds	Integer (min 30 max 3600)		
/scheduleFTP	List of FTP Schedule timer triggers			
/scheduleFTP/cron	Timer trigger	<p>“cron” timer trigger: <i>mm hh dd MM DD</i></p> <ul style="list-style-type: none"> <li>• <i>mm</i>: represents the minutes (from 0 to 59)</li> <li>• <i>hh</i>: represents the hour (from 0 to 23)</li> <li>• <i>dd</i>: represents the day of the month (from 1 to 31)</li> <li>• <i>MM</i>: represents the month (from 1 to 12)</li> <li>• <i>DD</i> : represents the day in the week: <ul style="list-style-type: none"> <li>▪ 0 = Sunday</li> <li>▪ 1 = Monday</li> <li>▪ = Tuesday</li> <li>▪ ...</li> <li>▪ 6 = Saturday</li> </ul> </li> </ul>		
/scheduleFTP/data	File type to send	<p>Integer:</p> <ul style="list-style-type: none"> <li>• Bit 0: Data (1=enabled, 0=disabled)</li> <li>• Bit 1: supervision (1=enabled, 0=disabled)</li> <li>• Bit 2: log (1=enabled, 0=disabled)</li> </ul>		
/monitoringPeriod	Hub monitoring period in minutes	<p>Integer (min 15 max 1440)</p> <p><i>null</i> or 0: disables monitoring</p>	<b>60</b>	