webdyn
flexitron group

WebdynEasy W M-Bus 868MHz

User Manual

www.webdyn.com

Index

1. Glossary 5
2. About this Document 7
2.1 Scope 7
2.2 Target Audience 7
2.3 Product Versions 7
User Manual 9
1. Presentation 9
1.1 General Description 9
1.2 Operating Principle 9
1.3 Interfaces (old and new) 10
1.4 Supported Equipment 12
1.5 Product and Accessory References 12
1.6 Technical Specifications 12
1.6.1 General Specifications 12
1.6.2 Technical Specifications 13
1.6.3 Software Specifications 16
1.6.4 Battery Life 16
1.7 Safety Instructions 18
1.8 Regulations 19
2. Installation 20
2.1 Prerequisite 20
2.2 Unpacking 20
2.2.1 Contents 20
2.2.2 Identification 20
2.3 Assembly 22
2.3.1 Opening/Closing the Box 22
2.3.2 Wall Mounting 22
2.4 Interface Description 23
2.4.1 Product Power Supply 23
2.4.2 Radio 25

2.4.3 Cellular Network 26
2.4.4 Indicators & Buttons 28
2.4.5 USB Interface 29
2.4.6 Bluetooth Interface (BLE) 30
3. Product Operation 31
3.1 Starting the Product 31
3.1.1 Operating Mode 31
3.1.2 Application Operation 35
4. Configuration 37
4.1 Local Configuration 37
4.1.1 USB 37
4.1.2 Bluetooth (BLE) 37
4.2 Configuration File 39
4.2.1 System 39
4.2.2 W M-Bus radio 43
4.2.3 Hub Connectivity 48
4.2.4 Alarms 54
4.2.5 Schedule and Monitoring 55
5. Operation 61
5.1 The Remote Server 61
5.1.1 The FTP Server 61
5.2 The Configuration 64
5.3 The Data 66
5.4 Alarms 68
5.5 ACK Acknowledgements 69
5.6 Commands 71
5.6.1 “Request” Command 73
5.6.2 “Factory” Command 74
5.6.3 “Firmware” Command 74
5.6.4 “Diag” Command 74
5.6.5 “ConfigGet” Command 75
5.6.6 “whiteListErase” Command 75

5.6.7 “operatorInit” Command 76
5.6.8 “logGet” Command 76
5.6.9 “logClean” Command 77
5.6.10 “supervisionClean” Command 77
5.6.11 “supervisionGet” Command 78
5.6.12 “certificate” Command 78
5.6.13 “certClean” Command 79
5.7 Supervision 79
5.8 The Log 86
6. Update 88
6.1 Local 89
6.2 Remote 89
7. Tools & Diagnostics 90
8. FAQ 91
9. Appendix 92
9.1 Configuration - Variable list 92
Offices & Support Contact 96

1. Glossary

NAME	DESCRIPTION
APN	Access Point Name: the name of the access point the gateway uses to connect to the Internet via a mobile connection.
AES	Advanced Encryption Standard: synthetic encryption algorithm.
BLE	Bluetooth Low Energy: wireless transmission technique in the form of a Bluetooth-based open standard, characteristically having much lower power consumption.
Base64	Encoded data using an alphabet with 64 characters.
BSON	Binary JSON: binary version of JSON: encoding used to reduce the size of transmitted data to save stand-alone system batteries.
FTP	File Transfer Protocol: communication protocol used to exchange files over a TCP/IP network.
FTPS	File Transfer Protocol Secure: 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.
2G	Second generation: second generation digital standard (2G) for mobile telephony including GSM, GPRS and EDGE.
HTTP	HyperText Transfer Protocol: client-server communication protocol developed for the Web.
IP	Internet Protocol: message protocol in charge of addressing and sending TCP packets over the network.
JSON	JavaScript Object Notation: JSON is an easily interpretable data exchange format.
LTE-M/ CAT-M1	Long Term Evolution – Machine: 4G mobile network specific to connected objects, i.e. low consumption and long range.
NB-IoT	NarrowBand - Internet of Things: 4G mobile network dedicated to connected objects, i.e. low consumption and long range.
NTP	Network Time Protocol: an NTP server makes it possible to time synchronise equipment.

IS	Information System: server with which the hub exchanges (configuration, data, alarms, etc.).
TCP	Transmission Control Protocol: 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.
TCP/IP	Transmission Control Protocol/Internet Protocol: 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.
W M-BUS	Wireless M-Bus: 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.

2. About this Document

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

Its purpose is to help operators install and configure their WebdynEasy 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

2.1 Scope

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

2.2 Target Audience

This guide is 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.

2.3 Product Versions

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.

V1.5	Added multi listening mode. Changed example of “wlFilter”. Changed example command “diag”. Added “cid” in all commands. Added “supervisionGet” command. Changed the “update” command to “firmware”. Modification of the “configGet” command.
V2.0	Added NB-IoT Adding supervision variables Addition and presentation of the new case Added network unregistration by magnet Added certificate management commands Added radio configuration (longHeader, skipVersionField and skipMediumField) Addition of commercial reference without BLE Added clarification on the use of the whiteList

User Manual

1. Presentation

1.1 General Description

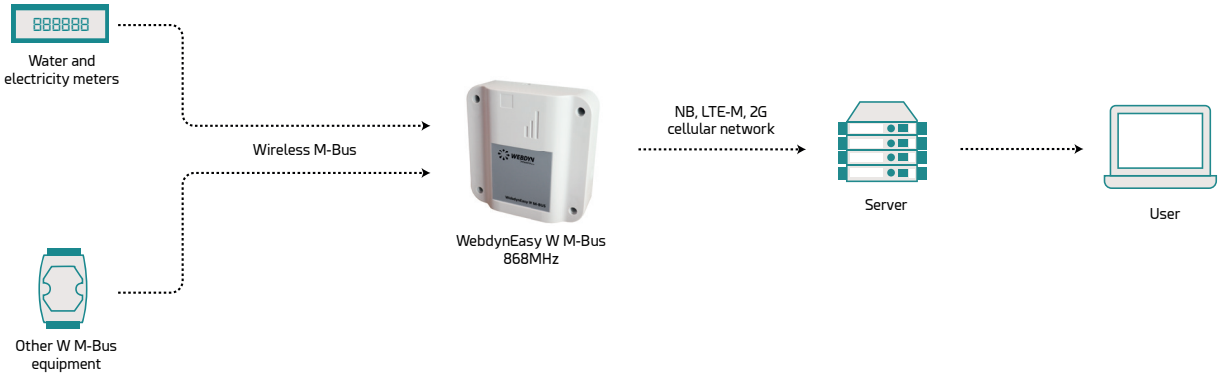
The purpose of the WebdynEasy W M-Bus hub is to collect data from Wireless M-Bus sensors such as meters (water, gas and electricity) and sensors (temperature, humidity, etc...). This technology is energy-efficient, the hub battery service life can exceed 10 years.

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

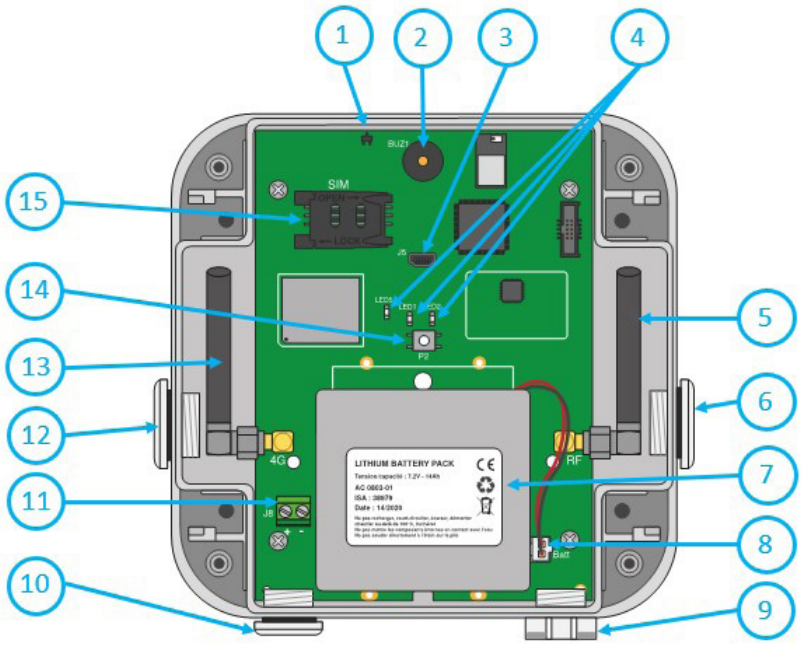
WebdynEasy W M-Bus allows you to build your private network and is part of a package. WM-Bus communication is point-to-point in a non-operated way. The meters or sensors regularly send data frames, which are received and the raw data (encrypted or transparent) contained in these frames are stored by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



It is possible to configure the WebdynEasy by connecting it by USB to a computer or by BLE using a smartphone application.

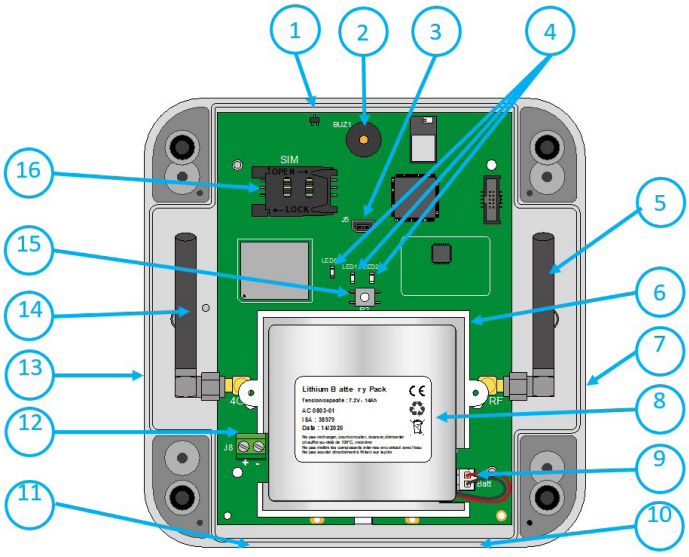
1.3 Interfaces (old and new)

Old Interfaces:



1. HALL effect sensor
2. Buzzer
3. Mini-USB connector
4. Indicators:
    - 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 LTE-M/NB-IoT/2G modem external antenna (optional)
13. LTE-M/NB-IoT/2G Modem SMA Antenna
14. Reset button
15. SIM card holder

New Interfaces:



1. HALL effect sensor

2. Buzzer

3. Mini-USB connector

4. Indicators:

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

5. 868MHz radio SMA antenna

6. Holder for holding the battery pack

7. Breakable output of the box for installation of M16 cable gland allowing the installation of an external 868MHz radio antenna (optional)

8. Battery Pack

9. Battery pack connector

10. Breakable output of the box for installation of M12 cable gland allowing installation of aerator optional

11. Breakable output of the box for installation of M12 cable gland allowing the installation of an external power supply (optional)

12. Terminal block for external 12 V power supply

13. Separable output of the box for installation of M16 cable gland allowing the installation of an external antenna LTE-M/NB-IoT/2G modem (optional)

14. LTE-M/NB-IoT/2G Modem SMA antenna

15. Reset button

16. 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-A02	WebdynEasy (without BLE)
WG0612-A12	WebdynEasy (with BLE)

Accessories:

REFERENCES	DESCRIPTIONS
AC0803-01	Battery pack 7.2V / 14Ah

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

1.6 Technical Specifications

1.6.1 General Specifications

REFERENCES	DESCRIPTIONS
Power supply	+12V 1A
Battery	7.2V 14Ah lithium battery pack (non-rechargeable) - in optimal configuration, the battery life is at least 10 years.
Consumption	P: 2 W Pmax: 5 W
Dimensions	Old case: 160 x 150 x 55 mm New case: 150 x 153 x 57 mm

Box	ASA Interior use
Weight	0.600 kg with battery pack 0.450 kg without battery pack
Operating temperature	-20°C/+55°C
Storage temperature	-20°C/+70°C (without battery) -20°C/+25°C (with battery)
Humidity	25% - 75%
Pollution rating	2
Certification	RED ROHS REACH
Regulation	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

REFERENCES	DESCRIPTIONS
Memory capacity for data	• Flash: 1.5MB (circular operation) • 2000 frames per listening window

Radio interference	Frequency: 868 MHz Antenna: Internal SMA (external as an option)
Cellular Interface Modem	GSM/GPRS/EDGE/LTE-M modem • 2G (GSM, GPRS): 850/900/1800/1900 MHz • LTE-M1: B1/B2/B3/B4/B5/B8/B12/B13/B18/B19/B20/ B25/B26/B27/B28/B66/B85 • NB-IoT : B1/B2/B3/B4/B5/B8/B12/B13/B18/B19/B20/ B25/B28/B66/B85 Antenna: Internal SMA (external as an option)
BLE interface	160 x 150 x 55 mm
SIM format	ASA Protected against dust and bad weather

Webdyn does not supply any SIM cards. Please contact an M2M operator that supports the 2G , NB-IoT 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/NB-IoT - Band 1	1920 MHz – 1980 MHz	
LTE-M/NB-IoT - Band 2	1850 MHz – 1910 MHz	
LTE-M/NB-IoT - Band 3	1710 MHz – 1785 MHz	
LTE-M/NB-IoT - Band 4	1710 MHz – 1755 MHz	
LTE-M/NB-IoT - Band 5	824 MHz – 849 MHz	
LTE-M/NB-IoT - Band 8	880 MHz – 915 MHz	
LTE-M/NB-IoT - Band 12	699 MHz – 716 MHz	
LTE-M/NB-IoT - Band 13	777 MHz – 787 MHz	
LTE-M/NB-IoT - Band 18	815 MHz – 830 MHz	21 dBm (class 5)
LTE-M/NB-IoT - Band 19	830 MHz – 845 MHz	
LTE-M/NB-IoT - Band 20	832 MHz – 862 MHz	
LTE-M/NB-IoT - Band 25	1850 MHz – 1915 MHz	
LTE-M - Band 26	814 MHz -849 MHz	
LTE-M - Band 27	807 MHz – 824 MHz	
LTE-M/NB-IoT - Band 28	703 MHz – 748 MHz	
LTE-M/NB-IoT - Band 66	1710 MHz – 1780 MHz	
LTE-M/NB-IoT - Band 85	698 MHz – 716 MHz	
BLE	2402 MHz – 2480 MHz	0 dBm

1.6.3 Software Specifications

SPECIFICATIONS	DESCRIPTIONS
Wireless M-Bus	T1, S1 and T1+C1 mode
BSON	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:

LENGTH OF LISTENING WINDOW IN MINUTES	2 UPLOADS/DAY	1 UPLOAD/DAY	1 UPLOAD/WEEK
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 NB-IoT:

LENGTH OF LISTENING WINDOW IN MINUTES	2 UPLOADS/DAY	1 UPLOAD/DAY	1 UPLOAD/WEEK
2	11 years	> 15 years	> 15 years
4	9 years	12 years	> 15 years
5	8 years	11 years	14 years
6	8 years	10 years	12 years
8	7 years	8 years	10 years
10	6 years	7 years	8 years
12	5 years	6 years	8 years
15	5 years	5 years	6 years
20	4 years	4 years	5 years
30	3 years	3 years	3 years

With 2G:

LENGTH OF LISTENING WINDOW IN MINUTES	2 UPLOADS/DAY	1 UPLOAD/DAY	1 UPLOAD/WEEK
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

When the mode of the modem is in auto mode, the modem connects by default in LTE-M then if this is unavailable, it falls back on 2G. If the first connection attempt fails, the modem will make a second connection attempt, but only in 2G.

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 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

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 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 hub.
- To have knowledge of the parameters to connect to the IS information system.
- To have a SIM card with an activated 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 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 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

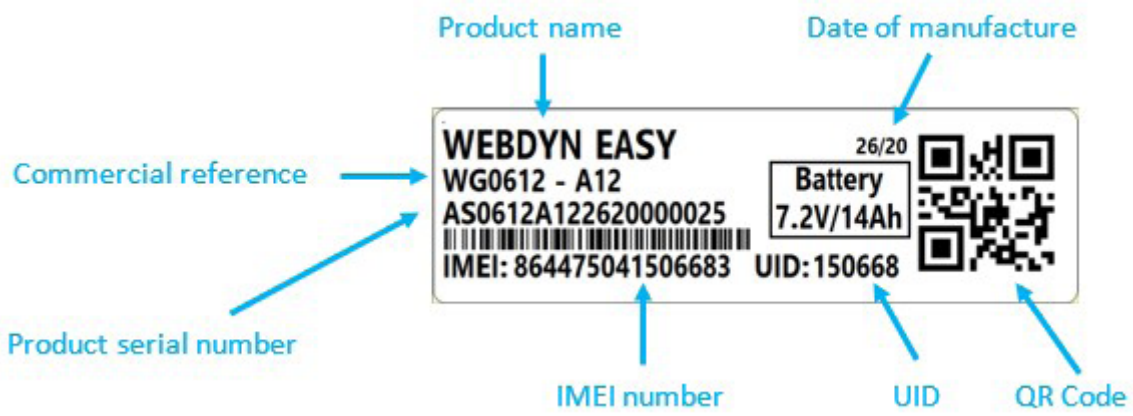
The commercial reference differs depending on whether or not the hub includes a customer configuration. The commercial reference is composed as follows:

- 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.

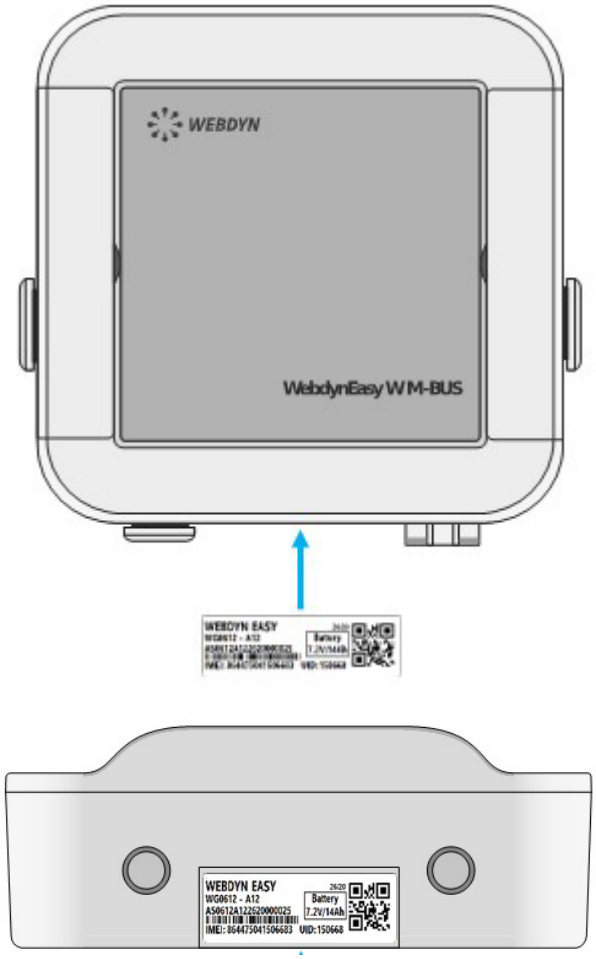
A unique identifier called UID is assigned to each WebdynEasy W M-Bus. It is used to configure the product and recognise it when you have several WebdynEasy W M-Bus products.

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:

The box can only be opened if it is not fixed to the wall. In the case where the box is fixed to a wall, it will have to be unhooked to be able to open it.

If the box is wall-mounted:

- Open the 2 doors on the front panel.
- For the old box, unscrew the 4 wall mounting screws in the recesses under the doors.
- For the new box, unscrew the 4 screws of the wall mounting in the slots on the front face.

Then follow these steps:

- Unscrew the 4 screws behind the box.
- Remove the cover.

Follow these steps to close the hub box:

- Place the cover on the box base.
- Screw in the 4 screws on the back of the box.

2.3.2 Wall Mounting

The WebdynEasy 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 diameter 4mm, head diameter 8mm maximum and length 25mm minimum).

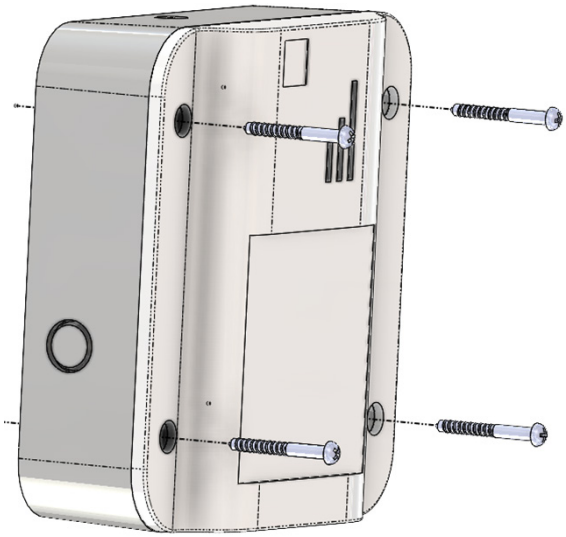
Follow the steps below to fix the hub to a wall:

Old box:

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

New box:

- Insert the 4 screws in the 4 accessible holes on the front panel. (see illustration below).
- Tighten the 4 screws.



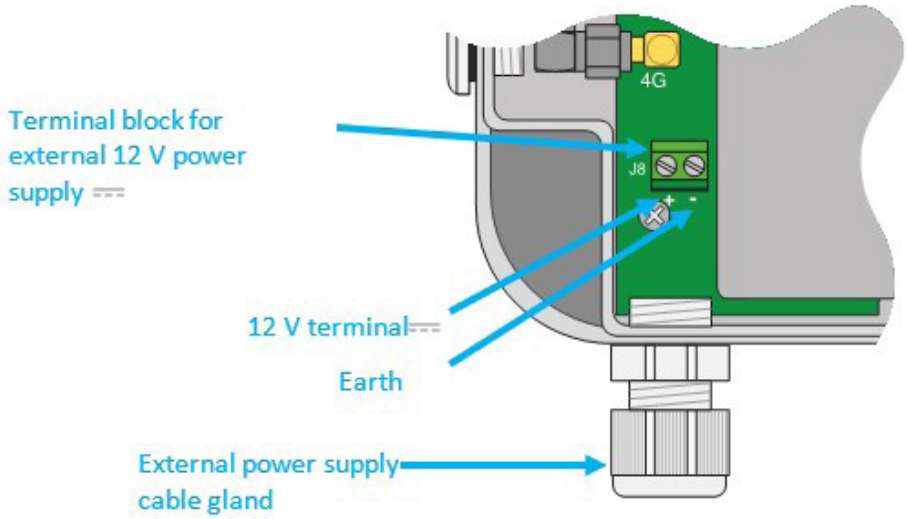
2.4 Interface Description

2.4.1 Product Power Supply

2.4.1.1 External Power Supply

The WebdynEasy 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.



In the new box, the cable gland is optional and is installed after breaking the breakable part provided in the box.

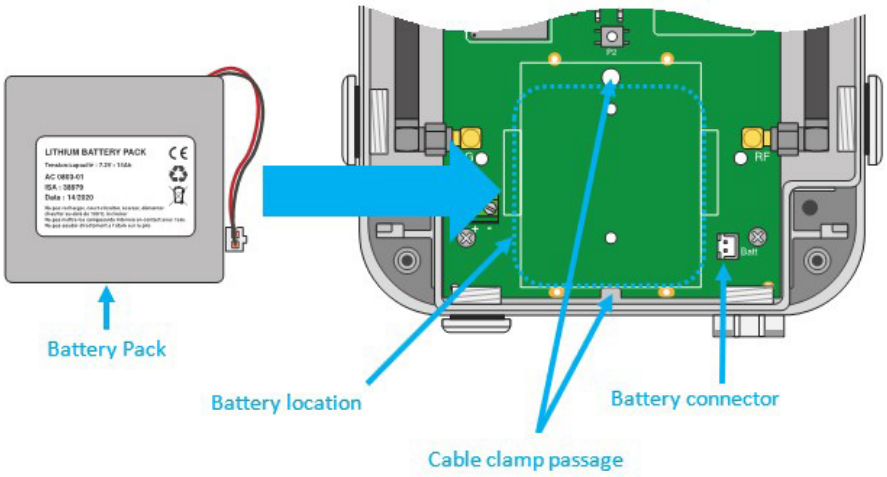
Make sure the power supply wires are connected to the proper terminals.

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

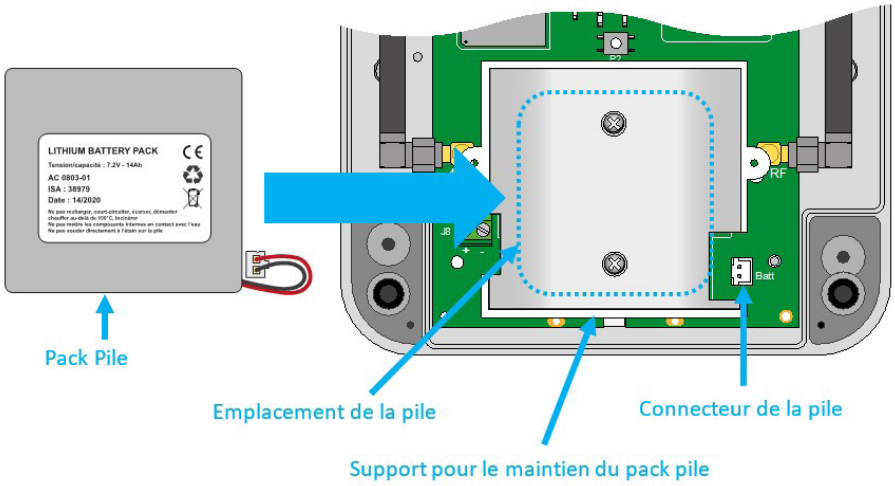
2.4.1.2 Battery Pack

The WebdynEasy W M-Bus 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.

In the old box, 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.



In the new box, the battery pack is installed in its holder and plugs into the connector identified “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 THAT RECOMMENDED BY WEBDYN. DISPOSE OF USED BATTERY PACKS ACCORDING TO THE INSTRUCTIONS.

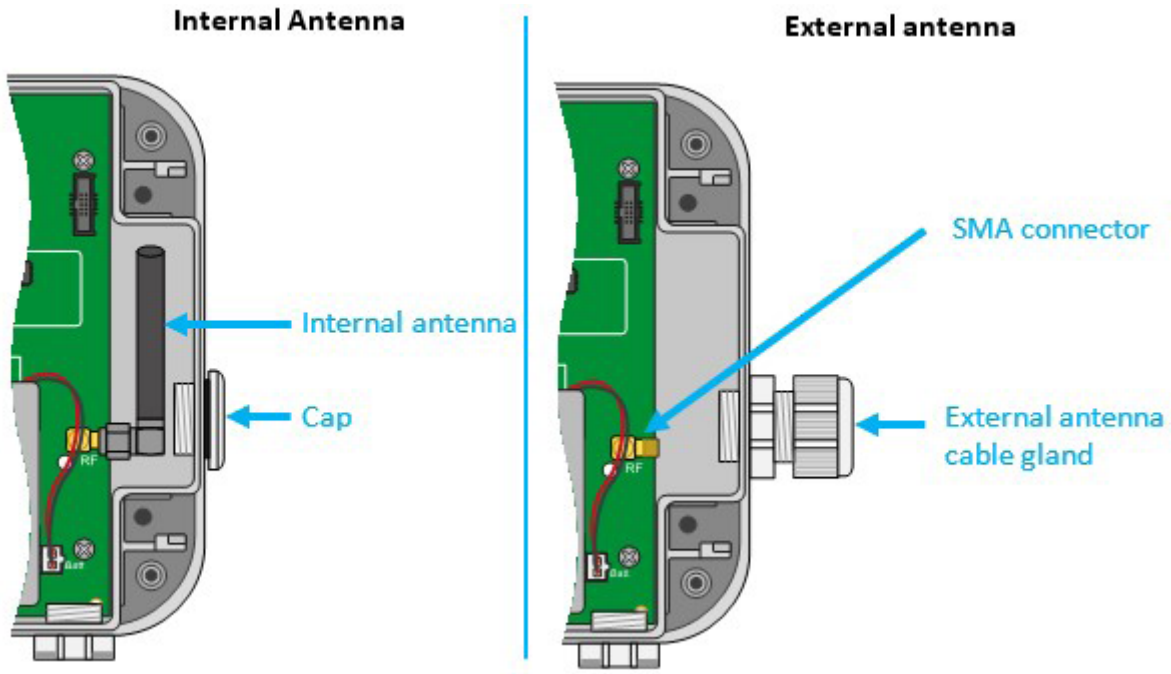
2.4.2 Radio

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.



In the new box, the cable gland is optional and is installed after breaking the breakable part provided in the box.

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

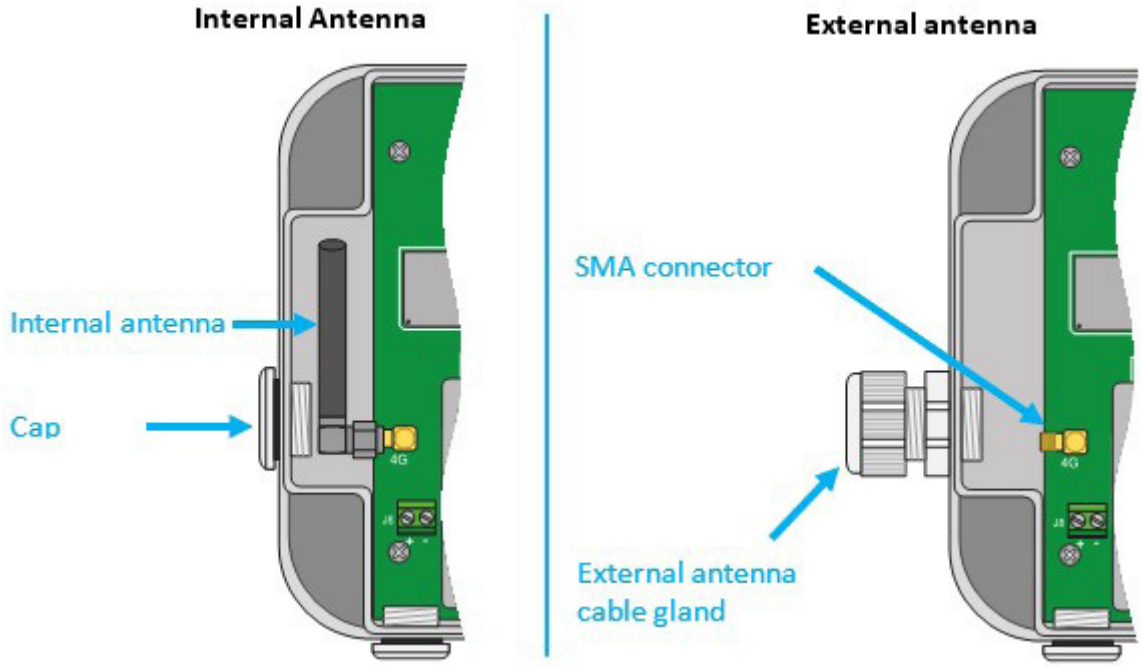
2.4.3 Cellular Network

The WebdynEasy W M-Bus hub includes a 2G, NB-IoT 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 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.



In the new box, the cable gland is optional and is installed after breaking the breakable part provided in the box.

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

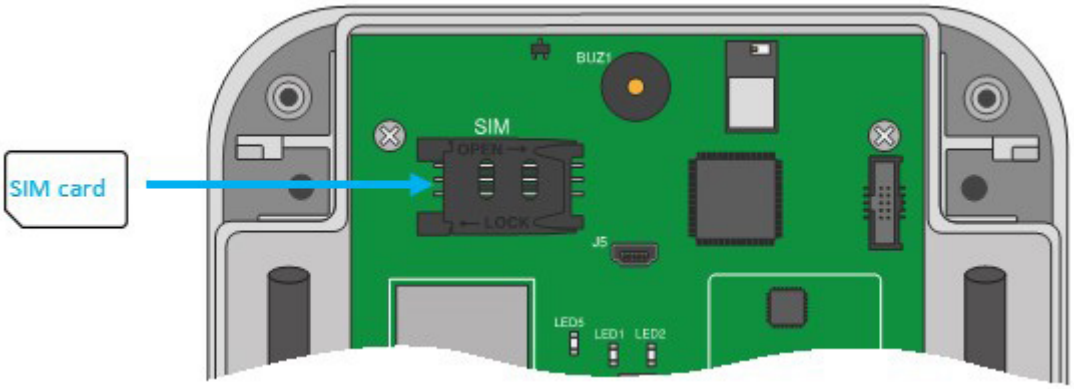
2.4.3.2 SIM Card

To use the LTE-M, NB-IoT or 2G modem connection to allow the hub to communicate with the remote server, the box must be opened(see section2.3.1: “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 is operating properly, insert a SIM card with the following specifications:

- 2G, NB-IoT 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, NB-IoT and LTE-M network.

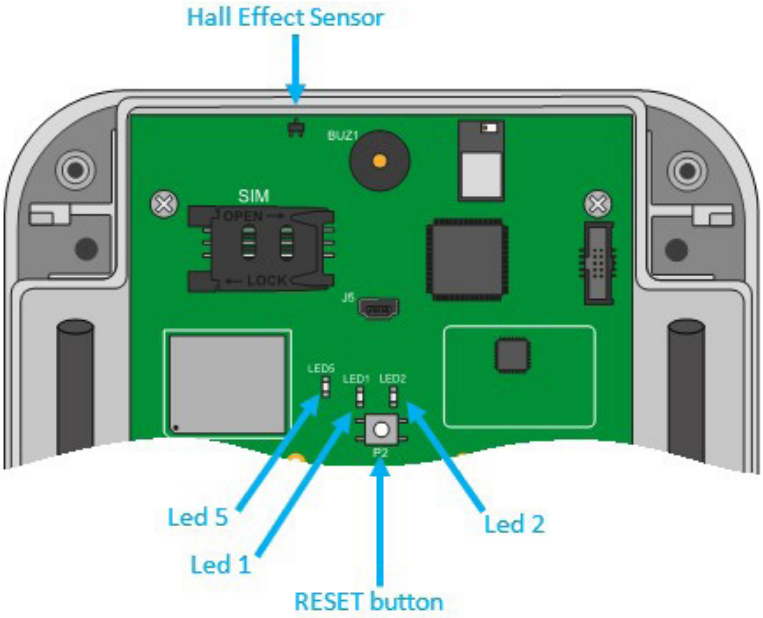
Please contact your SIM card provider to find out what information to enter to configuration the modem.

If the SIM card has an activated PIN code and it is incorrect the first time the hub isstarted , 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

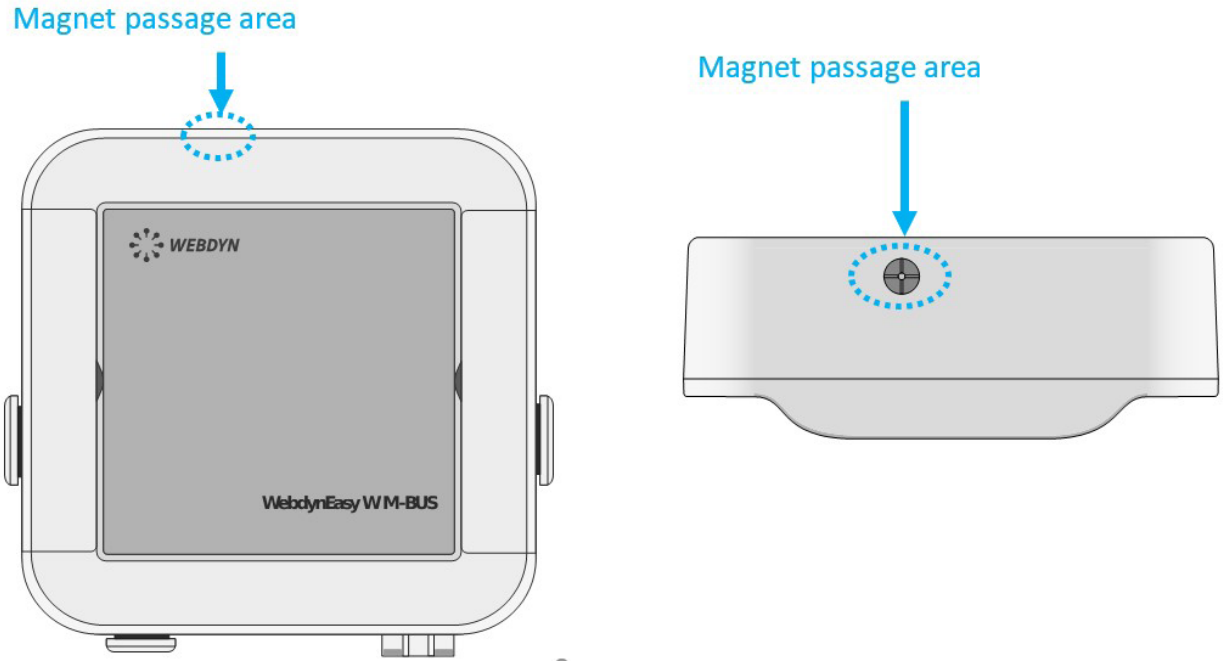
The hub is equipped with:

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



2.4.4.1 Hall Effect Sensor

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



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:
  - Place the magnet on the top of the box opposite the Hall effect sensor and keep it in place.
  - Briefly press the RESET button.
  - A long beep (1 second) is issued, informing that the factory settings have been re-applied.
  - 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:

- Switch the WebdynEasy W M-Bus to RUN mode.
- Connect the hub to a computer using a USB cord (A male - mini B male type).
- A new drive called “WebdynEasy W M-Bus” will appear on the computer (like a thumb drive).
- Open this drive.
- Copy the configuration, command or update file to the hard drive.
- Eject the removable drive and wait for confirmation from the system.
- Press the RESET button.

- The product restarts.
- The following steps are optional and are based on the commands issued.
- A new drive called “WebdynEasy W M-Bus” will appear on the computer (like a thumb drive).
- Open this drive.
- Open the ACK file and check the result of the command OR retrieve the files requested by the command.
- 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” mobile app to:

- Configure.
- Run a diagnostics command concerning its operating.

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

BLE is only available on the webdynEasy WM-Bus with the commercial reference WG0612-A12. (see chapter 1.5: “References of products and accessories” and chapter 2.2.2: “Identification”)

3. Product Operation

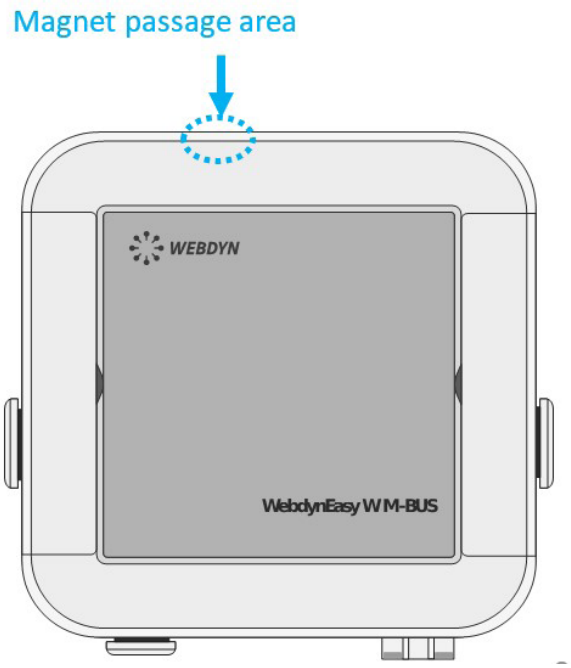
3.1 Starting the Product

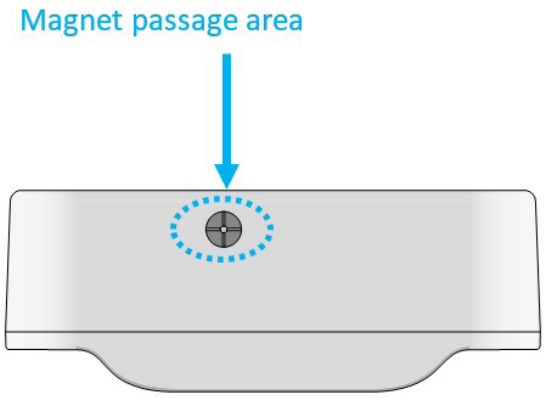
By default, the WebdynEasy W M-Bus hub is shipped in either:

- 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 for products to be stored for long periods of time limiting battery consumption.

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)





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 indicates the current mode.

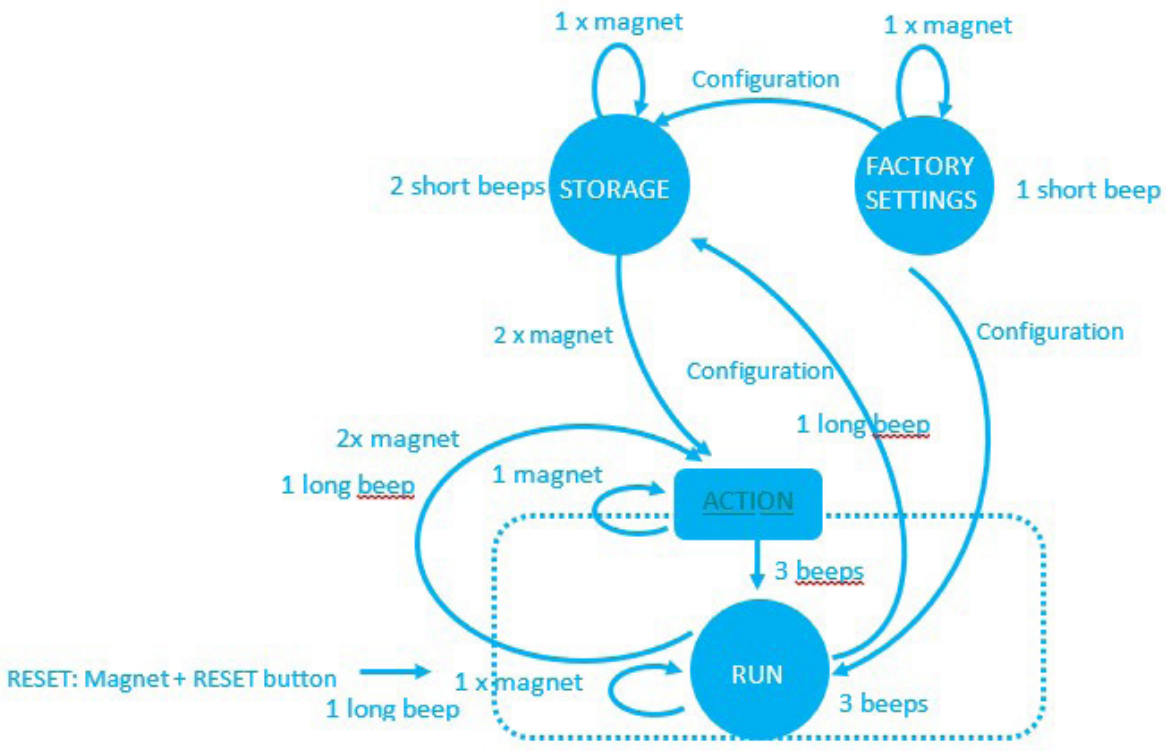
3.1.1 Operating Mode

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 RUNmode 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 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 paremeters “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 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.

In case of factory pre-configuration of custom settings, these are considered as the default settings and therefore applied when using a factory reset mechanism.

To find out if the WebdynEasy has a customer configuration, please check the commercial reference of the product. (see chapter 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 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 (only with commercial reference WG0612-A12).

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 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” 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)

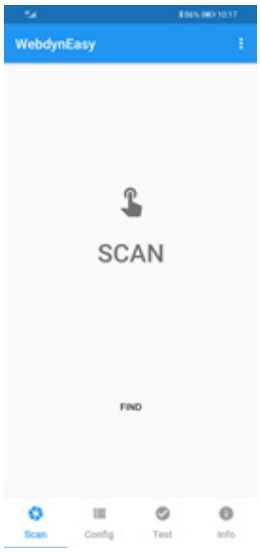
To use the local configuration with Bluetooth (BLE), an Android smartphone is required with the “WebdynEasy W M-Bus” app installed downloadable from Google Play(https://play.google.com/store/apps/details?id=com.webdyn.WebdynEasy W M-Bus).

BLE is only available on the webdynEasy WM-Bus with the commercial reference WG0612-A12. (see chapter 1.5: “References of products and accessories” and chapter 2.2.2: “Identification”)

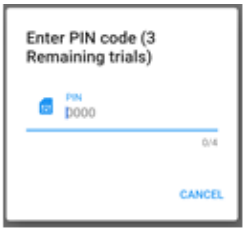
To synchronise the mobile app with the hub using Bluetooth, follow these steps:

- Pass the magnet over the product. (see section 2.4.4.1: “Hall Effect Sensor”).
- Activate Bluetooth on the smartphone.

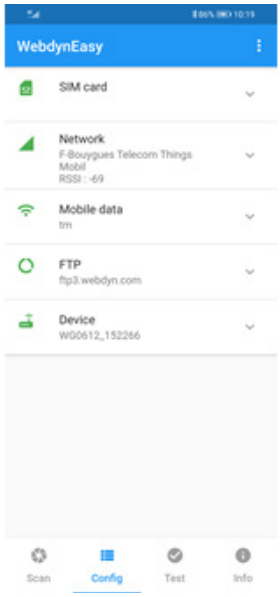
- Launch the “WebdynEasy W M-Bus” smartphone app.
- Click “SCAN”.



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



- 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.

If you launch a test from the mobile application and you have configured the multi listening mode, the test will be executed only on the first configured mode. (see chapter 4.2.2: “WM-Bus radio”).

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 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> -config.bson” file to “<uid> -cfg.bson”.

4.2.1 System

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

PARAMETERS	DESCRIPTION
mode	Hub operating mode: factorySettings: FACTORY SETTINGS mode storage: STORAGE mode run: RUN mode
name	Hub name

logLevel	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 STRORAGE 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
magnet	Configuration of the magnet action in RUN mode: 0: Bluetooth BLE+ Modem (default) 1: diagnosis command 2: request command
blePin	Bluetooth BLE identification code (default: “1234”)

whiteList	List of phone numbers authorised to issue commands (default: NULL, all numbers are authorised)
testCount	Number of diagnostic command sequences in a row between 1 and 30 (default 1)
timeout	Maximum action execution time in seconds between 60 and 3600. Functional only if the “testCount” parameter is equal to 1. (default disabled: 0)

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”:
      {
            “mode”: “run”,
            “name”: “WebdynEasy”,
            “logLevel”: 2
       },
      “crc”: 0
}
```

4.2.1.2 Security

All the hub security variables are in the “security” object in JSON.

PARAMETERS	DESCRIPTION
crcMode	BSON file CRC check: 0: No CRC check 1: Partial check (no check if CRC is 0) (default) 2: CRC check enabled

encryption	BSON file AES encryption: true: file encryption enabled false: file encryption disabled (default)
key	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 in hexadecimal (longer than 16 characters)	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 W M-Bus radio

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

PARAMETERS	DESCRIPTION
mode	WM-Bus mode used: 0: for S1 1: for T1 2 : for T1 + C1 (default)
duration	WM-Bus listening window duration in seconds (1 to 3600) The default is: 60
manufFilter	List of manufacturer codes (2 bytes: M-field) authorised in binary format (type 0x00 in JSON) (maximum 8 codes)
mediumFilter	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)

BFormatLFieldAdaptation	Modification of the L field of the WM-Bus frame for B format frames (see chapter 4.2.2.2: “L field for B format frames”): true: Adaptation of the L field from format B to format A (by default). false: Field L of format B is not modified.
longHeader	Use of meter information present in the long header in a long frame (refer to the “long header” of the “Open Metering system” specification): true: information used in the long header (default). false: information used in the short header
skipVersionField	In case of Whitelist, allows to ignore the version of the counter present in the header: true: ignore version false: takes the version into account (default)
skipMediumField	In the case of a Whitelist, allows you to ignore the type (“medium”) of the counter present in the header: true: ignore type false: takes the type into account (default)

If the “skipVersionField” and/or “skipMediumField” parameters of the WM-Bus radio part are modified, it is essential to reinject the whitelist file “<uid>-wlFilter.bson” into the concentrator.

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="},
                  " longHeader": true,
                  " skipVersionField": false,
                  " skipMediumField": false
            }
      },
      "crc": 0
}
```

If necessary, the concentrator has a multi-listening mode which makes it possible to recover the WM-Bus data from the sensors in one mode and then to recover the WM-Bus data from other sensors which are in another mode.

In the case of a multi listening mode, the “mode” and “duration” parameters are arrays. The first element of the arrays corresponds to the first mode triggered by the concentrator when retrieving WM-Bus data from the sensors, the second element of the arrays corresponds to the second mode used when retrieving data.

In the case of multiple listening modes, the list of manufacturer codes “manufFilter”, the list of medium codes “mediumFilter” and the white list “whiteList” apply to all the defined modes.

Example of a multi-mode listening:

By JSON file (to be converted into BSON format for the concentrator):

```
{
      "config":
      {
            "radio":
            {
                  "mode": [2,0],
                  "duration": [120,180],
                  "manufFilter": { "$type": "00","$binary":
"TDARpQ=="},
                  "mediumFilter": { "$type": "00","$binary": "BwE="},
            }
      },
      "crc": 0
}
```

Explanation :

Data recovery will start with T1+C1 mode for 120 seconds, then continue with S1 mode for 180 seconds.

4.2.2.1 Whitelist

It is possible to activate the management of acquisitions of sensors or counters according to the WM-Bus identifiers entered in the white list. During a listening window, it may close before the programmed duration if the data for all the equipment on the white list has been recorded.

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

The WebdynEasy hub whitelist file is in BSON format and its format is as follows: « <uid>-wlFilter.bson ».

If the “skipVersionField” and/or “skipMediumField” parameters of the WM-Bus radio part are modified, it is essential to reinject the whitelist file “<uid>-wlFilter.bson” into the concentrator.

The concentrator whitelist 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 allowing the identification of the sensor or the WM-BUS meter and is composed as follows:

M-FIELD	A-FIELD
2 octets	6 octets

To delete the equipment from the concentrator, you can either:

- Send a “whitelisterase” whitelist erasing command (see chapter 5.6.6: ““whiteListErase” command “whiteListErase” command”),
- Send a configuration file with a “null” value for the “wlFilter” object.

In the case of multiple listening modes, the white list applies to all the modes defined.

Example of a whitelist file converted to JSON:


```
{
      “wlFilter”:
       [
            { “id”: { “$binary”: “FIZEsRQABBE=MExBKpkgwAE=”, “$type”:
“00” } },
{ “id”: { “$binary”: “FIY1whQABBE=MEw9KZkgwAE=”, “$type”: “00” } },
{ “id”: { “$binary”: “FIaachQADw8=MEzHKZkgwAE=”, “$type”: “00” } },
{ “id”: { “$binary”: “MEzOXeAgmAE=MEw9KpkgwAE=”, “$type”: “00” } },
{ “id”: { “$binary”: “MEw5KpkgwAE=MEzBKZkgwAE=”, “$type”: “00” } },
{ “id”: { “$binary”: “MEw/2CMhDAQ=MEw2KpkgwAE=”, “$type”: “00” } },
{ “id”: { “$binary”: “MEzBHKogzAE=kiYQAwARHgM=”, “$type”: “00” } },
      ],
      “crc”: 0
}
```

Example of an erasure whitelist file converted to JSON:

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

4.2.2.2 L field for format B frames

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

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

In the WM-Bus frames recorded by the gateway, the CRCs are not present. Depending on the frame format, the processing of the frame is not the same and it is therefore necessary to know the format used by the counter.

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

However, it is possible to disable this behavior by applying the value "false" 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
mode	Connection type selection: 2G: forces the modem to 2G LTEM: Forces the modem to LTE-M NB-IoT: Forces the modem to NB-IoT auto: The modem manages the network automatically (default)
scanseq	Search sequence preference table. Functional only in “auto” mode. The 3 possible choices for the order of preference are: 2G: Authorize the modem in 2G LTEM: Authorize the modem in LTE-M NB-IoT: Authorize the modem in NB-IoT
band	List of authorized bands for the modem connection in NB-IoT (numerical value of the band). Available bands are: 1,2,3,4,5,8, 12,13,18,19,20,25,28,66,85
deregisterOnLocalAction	Deregistration of the modem on the operator network each time an action is triggered by the magnet: true: Deregistration enabled false: Deregistration disabled (default)
cpin	SIM card PIN code (max. 8 characters) null: disables the PIN code (default)
apn	Network Access Point (APN) identifier (32 characters max) null: no APN (default)
use	PPP connection identifier (32 characters max) null: no identifier (default)
pass	PPP connection password (32 characters max) null: no password (default)

timeout	Maximum time to connect to a network in seconds (between 60 and 1800) 300 (default)
randomDelay	Random wake-up time in minutes (between 0 and 60) 0: no time (default)

If no “band” parameter is entered, the modem scans all available NB-IoT bands. In order to optimize the search for the network in NB-IoT, it is strongly recommended to enter the bands authorized by the operator and the country. Please find below a table of the bands deployed worldwide for your information:

Bands	U.S.A.	China	Middle East	Japan	Korea	Europe	Australia
1		X		X			
2	X						
3		X	X		X	X	X
5		X			X		
8		X	X	X		X	
12	X						
13	X						
18				X			
19				X			
20						X	
26		X					
28			X				X

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",
                        "scanseq": [
                              "LTEM",
                              "2G",
                              "NB-IoT"
                        ],
                        "band": [
                              3,
                              8,
                              20
                        ],
                        "deregisterOnLocalAction": false,
                        "cpin": 1234,
                        "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
addr	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)

mode	FTP protocol choice: 0: FTP (default) 1: FTPS (FTP over TLS)
user	FTP account username (32 characters max)
pass	FTP account password (32 characters max)
dir	FTP server root directory (64 characters max) null: 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 uses the date and time in UTC+0 format.

PARAMETERS	DESCRIPTION
ntp	IP address or NTP server name (64 characters max) (Default address: “pool.ntp.org” 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.

By default, the concentrator uses the free NTP server “pool.ntp.org”, this server does not guarantee the accuracy of time synchronization and its robustness. It is strongly recommended to use a dedicated NTP server. Get closer to an NTP server provider.

Example:

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

```
{
      “config”:
      {
            “remote”:
            {
                  “ntp”: “pool.npt.org”
            }
      },
      “crc”: 0
}
```

4.2.4 Alarms

All the hub alarm variables are in the “alarms” object in JSON.

PARAMETERS	DESCRIPTION
oneAlarmPerDay	Only one alarm transmission per modem per day: true: sends at most one alarm transmission per day (default) false: transmission of all alarms during a day
temperature	Maximum temperature in degrees (°C) before an alarm is triggered (between 10 and 50) null or 0: disables the temperature alarms. (default)
timeGap	Monitoring of the hub clock drift compared to the time retrieved using NTP in seconds before an alarm is triggered (between 2 and 3600) null or 0: disables the clock drift alarms. (default)
battery	Monitoring of the remaining battery level in percent (%) before an alarm is triggered (between 10 and 99) null 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
cron	Timer trigger: mm hh dd MM DD
data	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.

In the case of a multi listening mode, the “data” parameter is an array. The first element of the array corresponds to the first mode triggered by the concentrator when retrieving WM-Bus data from the sensors, the second element of the arrays corresponds to the second mode used when retrieving data. (see chapter 4.2.2: “Radio WM-Bus” for the configuration of the multi mode)

Example of a multi-mode listening:

- By JSON file (to be converted into BSON format for the concentrator):

```
{
      "config":
      {
            "scheduleRadio": [
            { "cron" :"*/30 * * * * ", "data": [30,60]},
            { "cron" :"15 23 * * * ", "data": 120}
            ]
      },
      "crc": 0
}
```

Explanation :

In the example above, every 30 minutes we have a 30 second listening window on the first configured mode followed by a second 60 second listening window on the second configured mode. Every day at 11:15 p.m., we have a first listening window according to the duration defined in the config>radio>duration[1st mode] parameter, followed by a second listening window according to the duration defined in the config>radio parameter ->duration[2nd mode].

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
cron	Timer trigger: mm hh dd MM DD
data	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
monitoringPeriod	Hub monitoring period in minutes (between 15 and 1440) (60 by default) null or 0: 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/123456/").

You can configure your hub remotely from your FTP server. This is only possible if your WebdynEasy 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 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 1st, 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:

- At the start of the transfer the file has an additional ".tmp" extension.
- 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 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 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 A3 7B 0C 2A 00
```

5.2 The Configuration

The hub allows configurations using a .

Configuration File:

The WebdynEasy 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"

With:

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

Example: 123456-1591083697-config.bson

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

- Locally:

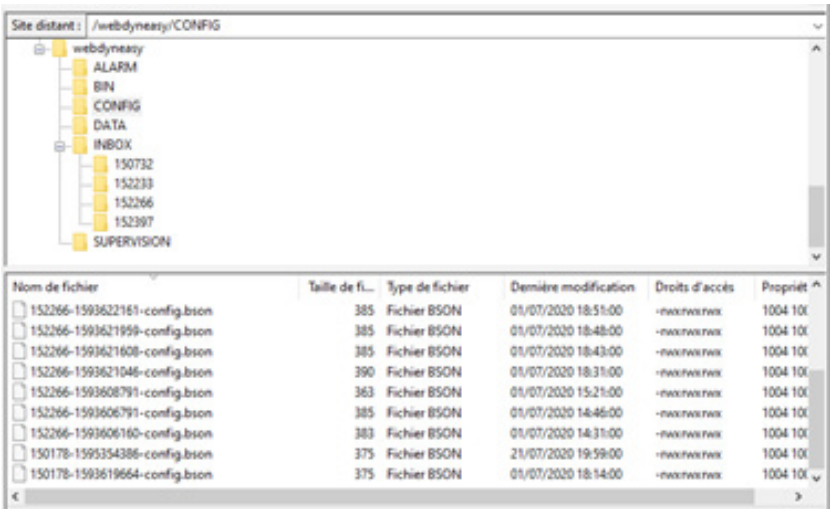
  Connect the USB to the hub. Then upload the configuration file (“<uid>-cfg.bson”).

  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:

  - Download the configuration file available on the server.
  - Delete the server configuration file.
  - Apply the new configuration.
  - Upload the acknowledgement file in the “ALARM” directory.



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

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

PARAMETERS	DESCRIPTION
T	Frame receipt timestamp (Number of seconds elapsed since January 1st , 1970 at midnight UTC precisely)
R	Frame RSSI level
F	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,
                  “F”:
                  {
                        “$type”: “80”,
                        “$binary”: “fAviGIskafDwA4E0UkO26w==”
                  }
            },
            {
                  “T”: 15600645887,
                  “R”: 178,
                  “F”:
                  {
                        “$type”: “80”,
                        “$binary”: “+plAP1iAwE0df2KBY1N7Iodr/
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 1st, 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
uid	Hub identifier
source	Source that triggered the (alarm) file upload
ts	File creation timestamp (Number of seconds elapsed since January 1st , 1970 at midnight UTC precisely)
alarm	Alarm Information

The "alarm" alarm information is as follows:

PARAMETERS	DESCRIPTION
type	Alarm type: temperature, battery or ntp
value	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 1st, 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
uid	Hub identifier
source	Source that triggered the file upload (ftp, usb, ...)
ts	File creation timestamp (Number of seconds elapsed since January 1st , 1970 at midnight UTC precisely)
ack	Command acknowledgement

The “ack” command acknowledgement consist of the following:

PARAMETERS	DESCRIPTION
type	Command type corresponding to the acknowledgement cmd: command config: configuration update
cid	Command identifier corresponding to the acknowledgement
result	Command result: ok: command or update successfully processed ko: error processing the command or the update
data	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”: “/fl0/152233-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; <uid>-<num>-cmd.bson

  With :

  - <uid>: Concentrator identifier
  - <num>: Number used to sequence a sequence of commands. (optional)

  Examples:

  123456-cmd.bson

  123456-1-cmd.bson

  123456-2-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 parameters in character string formats and which are:

- uid: unique hub identifier (optional).
- cid: command identifier (mandatory).

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
firmware	Hub software update
diag	Launch a diagnosis
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
supervisionGet	SupervisionGet Force the supervision deposit on the remote server
certificate	Add a certificate
certClean	Delete all certificates

If several command files are sent at the same time, the commands following “factory” and “firmware” can be lost. 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 "Firmware" Command

The "firmware" command is used to update the hub software (see section 6: "Update").

5.6.4 "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
```

5.6.5 “ConfigGet” Command

The “configGet” command forces the current configuration to copy the current configuration to the USB partition, this command has no effect if used in FTP. 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
}
```

5.6.6 “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
}
```

5.6.7 "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
}
```

5.6.8 "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
}
```

5.6.9 “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
}
```

5.6.10 “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
}
```

5.6.11 "supervisionGet" Command

The "supervisionGet" command forces the deposit of supervision on the remote server in the "/SUPERVISION" directory.

No subcommands or parameters are needed for this command.

Example :

- By JSON file (to be converted into BSON format for the concentrator):

```
{
      "cmd": {
            "type": "supervisionGetClean",
            "cid": "123456"
      },
      "crc": 0
}
```

5.6.12 "certificate" Command

The "certificate" command allows you to add a certificate to the concentrator.

In the order, you must indicate the name of the certificate to be taken into account in the "data" field.

You must also attach the certificate to add:

- Locally (USB): you must deposit the certificate and the order at the root of the mounted local disk "WebdynEasy". (see chapter 4.1.1: "USB")
- Remotely (FTP): the certificate must be deposited in the "/BIN" directory of the remote server, and a certificate addition command ("certificate") must be deposited in the "INBOX" FTP directory of your concentrator ("/INBOX/<uid> /", with <uid> the identifier of your concentrator).

Example :

- By JSON file (to be converted into BSON format for the concentrator):

```
{
      "cmd": {
            "type": "certificate",
            "cid": "123456",
            "data": "cert.pem"
      },
      "crc": 0
}
```

5.6.13 "certClean" Command

The "certClean" command erases all the certificates present in the concentrator.

No subcommands or parameters are needed for this command.

Example :

- By JSON file (to be converted into BSON format for the concentrator):

```
{
      "cmd": {
            "type": "certClean",
            "cid": "123456"
      },
      "crc": 0
}
```

5.7 Supervision

The supervision information is uploaded to the "/SUPERVISION" directory on the FTP server in BSON format. This is the supervision file name format: <uid>-<timestamp>-supervision.bson

Where:

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

The information from the “supervision” supervision is as follows:

SECTION	PARAMETERS	DESCRIPTION
identity	uid	Hub identifier
	name	Hub name
	SN	Serial number
version	hw	Hardware Version
	sw	Software version of the application
	swBle	Software version of the Bluetooth BLE module
	swModem	Software version of the Modem
uptime	run	Total operating time in RUN mode in seconds
	radio	Total radio listening time in seconds
	modem	Total modem time in seconds
	lowPower	Total standby time in seconds

measures	vAlim	Supply voltage in mV
	temperature	Hub temperature in °C
	histogram	Temperature bar chart
modem	rssi	RSSI for the last connection
	quality	Modem connection signal strength information
	nwinfo	Information on the technology, operator, band and channel used by the modem
	imei	Modem IMEI
	iccid	ICCID (identification number) of the SIM card
	num	SIM card phone number
	operator	Preferred operator name
	mode	Connection type (2G, LTEM, auto)
counters	wakeup	Total number of wake-ups
	magnetWakeup	Total number of wake-ups from the magnet
	bleWakeup	Total number of wake-ups from the Bluetooth BLE
	ftp	Total number of wake-ups by the FTP Schedule
	alarms	Total number of alarms sent
	radio	Total number of wake-ups by the radio Schedule
	usb	Total number of wake-ups by USB
	wd	Number of watchdog triggers
	fault	Number of fatal errors

dates	lastConfig	Timestamp of the last configuration update (optional)
	lastUpdate	Timestamp of the last software update (optional)
	lastPowerOn	Timestamp of the last power-up (optional)
	lastCnxOk	Timestamp of the last successful connection (optional)
	lastCnxKo	Timestamp of the last connection failure (optional)
battery	remainTime	Estimated remaining battery life in months (null: impossible to estimate the duration as a minimum of 1 month of use is required before the first estimate)
	capaPrcent	Percentage of remaining battery capacity
	capamAh	Capacity in mAh remaining in the battery
WmBus	CptRxPreamb	Number of radio preambles detected in radio reception
	CptRxSync	Number of radio synchronization codes detected during radio reception
	CPtRxFrames	Number of radio frames decoded in radio reception
	CptRxFramesCrcKo	Number of CRC errors in radio reception
	CptRxFramesDecodeKo	Number of decoding errors in radio reception
	CptRxFramesLenKo	Number of length errors in radio reception
	CptRxOverrun	Number of buffer overruns in radio reception
	CptRxTimeOut	Number of times exceeded in radio reception
	CptRecFrames	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
below -10°c	Number of times the hub temperature was below -10°C
-10°c to 0°c	Number of times the hub temperature was between -10°C and 0°C
0°c to 10°c	Number of times the hub temperature was between 0°C and 10°C
10°c to 20°c	Number of times the hub temperature was between 10°C and 20°C
20°c to 30°c	Number of times the hub temperature was between 20°C and 30°C
30°c to 40°c	Number of times the hub temperature was between 30°C and 40°C
above 40°c	Number of times the hub temperature was above -40°C

Information on the signal strength of the “quality” modem connection breaks down as follows:

PARAMETERS	DESCRIPTION
rssi	Received signal strength (available in 2G, LTE-M and NB-IoT)
rsrp	Received reference signal strength (available in LTE-M and NB-IoT)
rsrq	Received reference signal quality (available in LTE-M and NB-IoT)
sinr	Signal to noise ratio plus interference (available in LTE-M and NB-IoT)

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,
                  “quality”: {
                        “rssi”: -57,
                        “rsrp”: -79,
                        “rsrq”: -13,
                        “sinr”: 12
                  },
                  “nwinfo”: “\”eMTC\”,\”20801\”,\”LTE BAND 20\”,6400”,
```

```
                "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. This is the supervision file name format: <uid>-<timestamp>-log.bson

Where:

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

The list of "logs" is as follows:

PARAMETERS	DESCRIPTION
T	Event timestamp (Number of seconds elapsed since January 1st , 1970 at midnight UTC precisely)
C	Event Code (see "Appendix B - Log - Event List")
D	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,
filter disable"   "D": "[FRAME][Frame_FastFilter_Init():57]white list
            },
            {
                  "T": 1598348100,
                  "C": 33554432,
                  "D": "[RADIO][appRadioTask():98]Windowsduration: 30"
            },
            {
                  "T": 1598348130,
                  "C": 33554432,
count:5"          "D": "[FRAME][Frame_Process_Cmd():226]RX Frame
            },
            {
                  "T": 1598348134,
                  "C": 3,
                  "D": "[PERIODIC][periodicTask():49]periodic task"
            },
            {
                  "T": 1598348134,
                  "C": 33554432,
Tue Aug 25 09:40:00 2020\n data:30"  "D": "[RADIO][appRadioNextAlarm():144]Next Alarm:
            }
      ],
      "crc": 0
}
```


contact@webdyn.com | webdyn.com
V2.0 subject to changes | Webdyn © by Flexitron Group

6. Update

The WebdynEasy 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_Vx.x.bson” and “WebdynEasy 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 “firmware” 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.

BLE is only available on the webdynEasy WM-Bus with the commercial reference WG0612-A12. (see chapter 1.5: “References of products and accessories” and chapter 2.2.2: “Identification”).

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

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

```
{
      “cmd”: [{
            “type”: “firmware”,
            “cid”: “firmwareBLE”,
            “data”: “WebdynEasy_BLE_V1.0.bson”
       },
       {
            “type”: “firmware”,
            “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": "firmware",
            "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": "firmware",
            "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 "`firmware`" 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

WebdynEasy WM-Bus configuration software:

In order to facilitate the configuration of the WebdynEasy WM-Bus concentrator, product-specific configuration software has been developed by Webdyn. The software makes it possible to manage the configuration, to easily create “whiteLists” but also to send commands to the products.

This “WebdynEasyConfigurator” software is available free of charge for Windows on our site at this address:

https://www.webdyn.com/support

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. 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.

9. Appendix

9.1 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	INFO 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: Adaptation of the L field from format B to format A • false: Field L of format B is not modified	true	
/radio/duration	Duration of the WM-Bus listening window in seconds	Integer (min 1 max 3600)	60	
/radio/manufFilter	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/longHeader	Use of meter information present in the long header in a long frame (refer to the “long header” of the “Open Metering system” specification)	Boolean: • true: information used in the long header • false: information used in the short header	true	
/radio/skipVersionField	In case of Whitelist, allows to ignore the version of the counter present in the header	Boolean: • true: ignore the version • false: takes the version into account	false	
/radio/skipMediumField	In the case of a Whitelist, allows you to ignore the type (“medium”) of the counter present in the header Addition of	Boolean: • true: ignore the version • false: takes the version into account	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: file encryption disabled	false
/security/key	AES128 CBC Key	Hexadecimal 16 bytes	null
/local/magnet	Configuration of the magnet action in RUN mode	List: • 0: Bluetooth BLE + Modem • 1: diagnosis • 2: request	0
/local/blePin	Identification code for Bluetooth BLE	4-digit text	1234
/local/whiteList	List of phone numbers authorised to send commands (maximum 4)	Text up to 16 digits	null
/local/testCount	Number of diagnostic sequences in a row	Integer (min 1 max 30)	1
/local/timeout	Maximum execution time of the action in seconds. Functional only if the “testCount” parameter is 1.	Integer (min 60 max 3600) and disabled = 0	0
/remote/modem/mode	Connection type selection	List: • 2G: forces the modem to 2G • LTEM: Forces the modem to LTE-M • auto: The modem manages the network automatically	auto
/remote/modem/scanseq	Search Sequence Preference Table	List: • 2G: Forces the modem to 2G • LTEM: Forces the modem to LTE-M • NB-IoT: Forces the modem to NB-IoT	
/remote/modem/band	List of authorized bands for modem connection in NB-IoT	List: 1,2,3,4,5,8,12,13,18,19,20,25,28 ,66,85	
/remote/modem/ deregisterOnLocalAction	Deregistration of the modem on the operator network each time an action is triggered by the magnet	Boolean: • true: Unregistration enabled • false: Deregistration disabled (default)	false
/remote/modem/cpin	SIM card PIN code	Text (8 characters max)	null
/remote/modem/apn	Network access point (APN) identifier	Text (32 characters max)	null
/remote/modem/user	PPP connection login	Text (32 characters max)	null
/remote/modem/pass	PPP connection password	Text (32 characters max)	null
/remote/modem/timeout	Maximum time to connect to a network in seconds	Integer (min 60 max 1800)	300
/remote/modem/randomDelay	Random wake-up time in minutes	Integer (min 0 max 60) 0: no delays	0
/remote/ftp/mode	FTP protocol choice	List: • 0: FTP • 1: FTPS (FTP over TLS)	0

/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)	null
/remote/ftp/user	FTP account login	Text (32 characters max)	null
/remote/ftp/pass	FTP account password	Text (32 characters max)	null
/remote/ftp/dir	FTP server root directory	Text (64 characters max) null: FTP account root directory	null
/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)	null
/alarms/oneAlarmPerDay	Only one alarm transmission per modem per day	Boolean: • true: sends at most one alarm transmission per day (default) • false: transmission of all alarms during a day	
/alarms/temperature	Maximum temperature in degrees (°C) before an alarm is triggered	Integer (min 10 max 50) null or 0 : disables the temperature alarms.	null
/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.	null
/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.	null
/scheduleRadio	List of radio Schedule timer triggers		
/scheduleRadio/cron	Timer trigger	"cron" timer trigger: mm hh dd MM DD • 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 - = Tuesday - ... - 6 = Saturday	
/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	“cron” timer trigger: mm hh dd MM DD • 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 - = Tuesday - ... - 6 = Saturday	
/scheduleFTP/data	File type to send	Integer: • Bit 0: Data (1=enabled, 0=disabled) • Bit 1: supervision (1=enabled, 0=disabled) • Bit 2: log (1=enabled, 0=disabled)	
/monitoringPeriod	Hub monitoring period in minutes	Integer (min 15 max 1440) null or 0: disables monitoring	60

Offices & Support Contact

SPAIN

C/ Alejandro Sánchez 109
28019 Madrid

Phone: +34.915602737
Email: contact@webdyn.com

FRANCE

26 Rue des Gaudines
78100 Saint-Germain-en-Laye

Phone: +33.139042940
Email: contact@webdyn.com

INDIA

803-804 8th floor, Vishwadeep Building
District Centre, Janakpurt, 110058 Delhi

Phone: +91.1141519011
Email: contact@webdyn.com

PORTUGAL

Av. Coronel Eduardo Galhardo 7-1ºC
1170-105 Lisbon

Phone: +351.218162625
Email: comercial@lusomatrix.pt

TAIWAN

5F, No. 4, Sec. 3 Yanping N. Rd.
Datong Dist. Taipei City, 103027

Phone: +886.965333367
Email: contact@webdyn.com

SUPPORT

Madrid Offices

Phone: +34.915602737
Email: iotsupport@mtxm2m.com

Saint-Germain-en-Laye Offices

Phone: +33.139042940
Email: support@webdyn.com

Delhi Offices

Phone: +91.1141519011
Email: support-india@webdyn.com

Taipei City Offices

Phone: +886.905655535
Email: iotsupport@mtxm2m.com