FC0000-3-15-CM_LoRa-00-DO-1.6

# CM868LRxx & CMUS915LRxx Magnetic contact Programming manual



## 0.1 Table of Contents

# 1 Document history

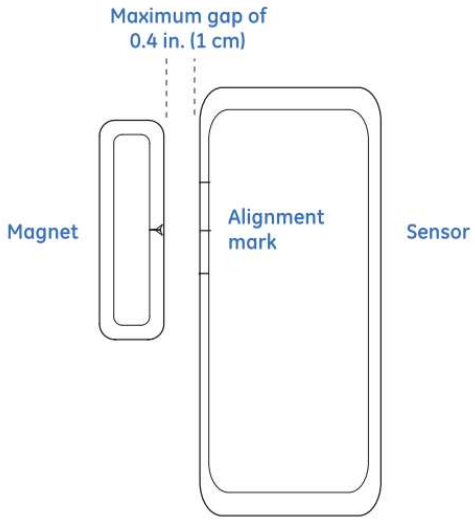
| Rev. | FW rev | Author | Note / remarks | Date |
|---|---|---|---|---|
| 1.0 | v0.7 | R&D laboratory GM | Document created. Preliminary version (DRAFT) | Nov. 10, 2015 |
| 1.1 | v0.7 | R&D laboratory GM | Added new feature on Hardware re. B | March 4, 2016 |
| 1.2 | v0.7 | R&D laboratory GM | Added OPCNT on Alive description on chapter 3;<br>Added CNFGRGST definition;<br>Added ACK/NACK messages on port 10 from sensor to server;<br>TMELPS [24 bit] increased to 16777215 sec;<br>Added Hysteresis values HYSTH and HYSTT;<br>Added threshold setting parameter NUMTHRL for temperature and humidity;<br>New structure of message from server to sensor: 1 byte header and 1 byte footer introduced;<br>added TMALIVE [16 bit] | April 23, 2016 |
| 1.3 | v0.7 | R&D laboratory GM | Added FLAGS features: LED always OFF, Stop Blinking LED | May 9, 2016 |
| 1.4 | v0.8 | R&D laboratory GM | Text correction on port 6 and 7 on chapter 4<br>Text correction on port 6 and 7 on chapter 4<br>Added more info to parameters in ch.3<br>Added LED blinking description message on port 20 on ch.4<br>Added payload information on port 12 specification – ch 5<br>Added graphical information on message from server to sensor on ch. 5 | May 25, 2016 |
| 1.5 | V0.8 | R&D laboratory GM | Changed minimum battery level at 25% | June 27, 2016 |
| 1.6 | V0.9 | R&D laboratory GM | FW v0.9 supports OTAA<br>Added OPCNT parameter into ALIVE<br>Amended NACK message length definition and added description of ACK and NACK messages.<br>Amended TMELPS definition<br>Amended MODE 3 message description of 10 Byte | Sept 5, 2016 |

# 2 Introduction

This is the Door/Window Sensor installation instructions for models CM868LR, CM868LRTH CMUS915LR and CMUS915LRTH.
You can install the sensor on doors, windows, and many other objects that can be opened and closed. The sensor transmits signals to the Lora™ network when a magnet mounted near to the sensor is moved away from or closer to the sensor. In the sensor, there are two reed switches for easy installation in many applications type. **Do not enable both reed simultaneously**.
The sensor is equipped with a cover tamper for added security.
CM868LR (868MHz) and CMUS915LR (915MHz) are contact magnetic sensors for European and US frequency bands.
CM868LRTH and CMUS915LRTH are special versions of the above with additional temperature and humidity sensors.
CM868LR, CM868LRTH CMUS915LR and CMUS915LRTH are equipped with an internal red LED that blinks for 100ms every 15s if MODE 2 is set and when the number of reed counter equals the threshold CNTELPS. This gives a visual indication when a certain number of detections that can be set is reached.

## 2.1 Installation guidelines

Use the following installation guidelines:
• Mount the sensor on the doorframe and the magnet on the door. If the sensor is used on double doors, mount the sensor on the least-used door and the magnet on the most-used door.
• Make sure the alignment arrow on the magnet points to the alignment mark on the sensor (Figure 2).
• Place sensors at least 4.7 in. (12 cm) above the floor to avoid damaging them.
• Avoid mounting sensors in areas where they will be exposed to moisture or where the sensor operating temperature range of -4 to 120°F (-20 to 55°C) will be exceeded.



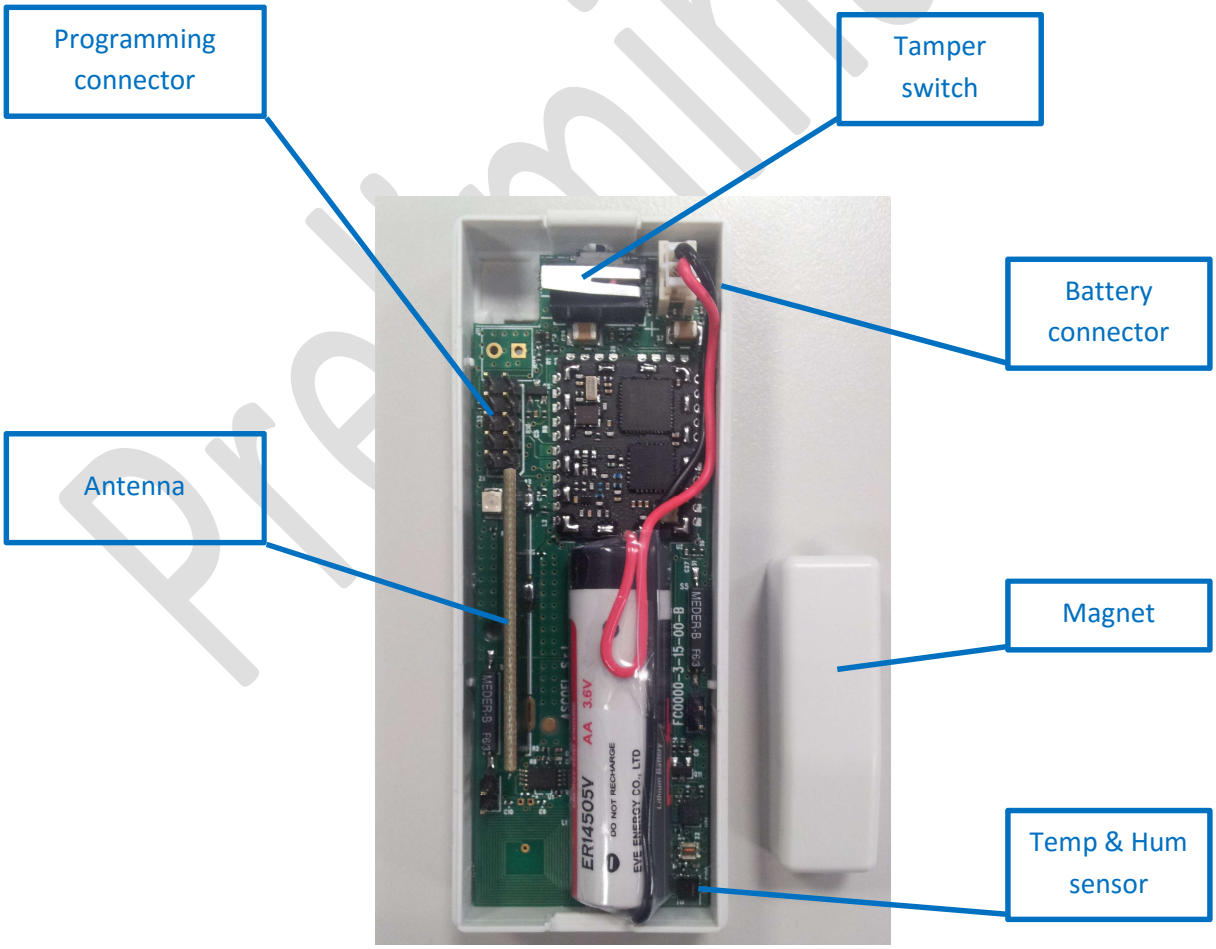
## 2.2 Battery replacement

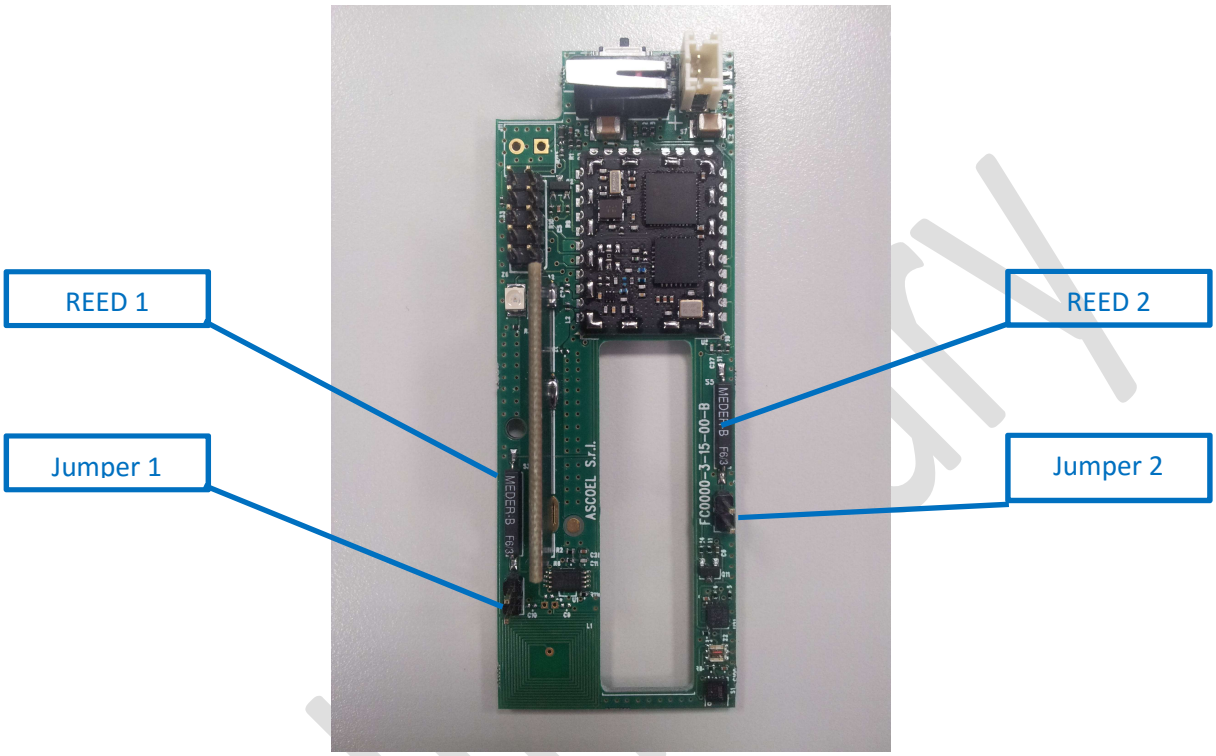
When the system indicates the sensor battery is low, replace it immediately.
Use the recommended replacement batteries (see Table 1 Technical Specification) or contact technical support for more information.
To replace the batteries, do the following:

1. To remove the sensor cover, press a small flathead screw-driver into the slot on both sides of the sensor. This will disengage the clips holding the cover and base.
2. Disconnect the battery cable from the board. Remove the old battery and replace it with another one as per battery specification reported in chapter 2.
3. Insert the replacement battery and plug the cable in to connector (see below picture)

**CAUTION**
**RISK OF EXPLOSION IF BATTERY IS REPLACED BY AN INCORRECT TYPE.**
**DISPOSE OF USED BATTERIES ACCORDING TO THE INSTRUCTIONS**

**Only authorized and qualified personnel may** do any of the assembly, disassembly, installation and commissioning work.






To select the desired reed, install the relative jumpers.
Do not enable both the reed at the same time

# 3 Technical Specification

| Specifications | | | | |
|---|---|---|---|---|
| **Parameters** | **Min** | **Typ** | **Max** | **unit** |
| Frequency band CM868LRxx | - | 867.1 < f < 868.5 | - | MHz. |
| CMUS915LRxx | - | 902 < f < 928 | - | MHz |
| RF power (EU868) | 2 | - | 14 | dBm EIRP |
| RF power (EN915) | 2 | | 18 | dBm EIRP |
| Modulation | | LoRa ™ | | |
| Protocol | | LoRaWan 1.0 Class A Client 3.4.1 | | |
| RX sensivity | | | -138 | dBm |
| Battery | | 1pz AA 3.6V lithium-thionyl 2700mA<br>By EVE P/N ER14505V | | |
| Temperature range | -20° | +25°C | +55°C | degree |
| Antenna | - | PCB printed | - | |
| Power supply | 2.1 | 3 | 3.6 | Vdc |
| Consumption standby | 13 | 15 | 17 | uA |
| Consumption TX | 60 | 70 | 80 | mA |
| Dimension | | 97 x 38 x 25 | | mm |
| Reference standards | | EN 60950-1:2006 + A11:2009 + A1:2010 + A12:2011 -<br>EN 62311:2008<br>EN 301 489-1 V1.9.2<br>EN 301 489-3 V1.6.1; Part 3<br>EN 300 220 V2.4.1 | | |

*Table 1 Technical Specification*

# 4 Parameter Definition

**ALIVE**

The payload is in raw format.

The ALIVE message is composed by:

*CM868LR and CMUS915LR* (4 Byte total):

| Name | Type | Function |
|---|---|---|
| BATTERY | Unsigned char (8 bits) | Report type and percentage of battery level |
| EVENT | Unsigned char (8 bits) | Event flag. See below |
| OPCNT | Unsigned Int (16bits) | Opening counter |

*CM868LRTH and CMUS915LRTH* (10 Byte total):

| Name | Type | Function |
|---|---|---|
| BATTERY | Unsigned char (8 bits) | Report type and percentage of battery level |
| EVENT | Unsigned char (8 bits) | Event flag. See below |
| TVALUE | Float (4 bytes) | Temperature value |
| RHVALUE | Float (4 bytes) | Relative Humidity value |

The ALIVE message is sent spontaneously to the server every 50min. This timing interval can be changed modifying TMALIVE from the server (see chapter 6)

**BATTERY**

1 Byte unsigned char. This parameter contains the information on type and percentage of the battery level:

| | *MSB 7* | *6* | *5* | *4* | *3* | *2* | *1* | *0 LSB* |
|---|---|---|---|---|---|---|---|---|
| *Bit value* | x | X | x | x | x | x | x | x |

[7] Battery Type
1 = 3.6V Lithium-thionyl
0 = 3.0V Alkaline Battery

[6:0] battery level expressed as a percentage of charge

The battery voltage can be derived as follows:

Batt_LOW_LVL + (((Batt_HIGH_LVL - Batt_LOW_LVL) * percentage)) / 100)

Where:

Batt_LOW_LVL = 2100mV
Batt_HIGH_LVL = 3000mV if battery type is Alkaline
Batt_HIGH_LVL = 3600mV if battery type is Lithium-thionyl
Percentage = the value of bit [6:0] *(cannot have the value 0)*

**EVENT**

1 Byte unsigned char.

| | MSB 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 LSB |
|---|---|---|---|---|---|---|---|---|
| *Bit value* | **0** | **0** | **0** | **x** | **x** | **x** | **x** | **x** |

[7:5] reserved

[4:3] **External input status** (still not implemented in current FW revision)
1:1 = line open
1:0 = line short circuit
0:1 = alarm
0:0 = OK

[2] **Battery status**
1 = low battery event
0 = battery OK

[1] **Tamper**
1 = Tamper alarm
0 = Tamper no alarm

[0] **REED sensor**
1 = Intrusion alarm
0 = Intrusion no alarm

If Low battery, tamper, reed event occurred the sensor will transmit a spontaneous message on port 30 (see ch 4)
More events are possible at the same time

**TVALUE**

Temperature value composed by 4 byte with 2digit, 2 digit format (e.g. 22,06°C)

**RHVALUE**

Relative humidity value composed by 4 byte with 2digit, 2 digit format (e.g. 56,82%)
TVALUE and RHVALUE are specific parameters used only by CM868LRTH and CMUS915LRTH.

To convert the four bytes in float value (little endian) a brief example in C language is reported:

```
union temperature
{
        unsigned char temp[4];
        float number;
}temp;

union humidity
{
        unsigned char rh[4];
        float number;
}hum;

sprintf_s(buf, "\n\rTemperature:%6.2f degree   Humidity RH:%6.2f%%", temp.number, hum.number);
```

The ALIVE message, who is sent based on TMALIVE, transmits the value of the BATTERY level, EVENT byte and TVALUE and RHVALUE.

**OPCNT**

16 bit unsigned counter.

Number of reed switch opening.

This counter can be reset at the end of the transmission using CNFGRGST.

**FLAGS**

16 bit parameter.

First byte is for settings, the second one is for requested command to be set.

| | MSB 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 LSB |
|---|---|---|---|---|---|---|---|---|
| Bit value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x |
| | MSB 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 LSB |
| Bit value | 0 | 0 | 0 | 0 | 0 | 0 | x | x |

[15:9] Unused - To be defined

[8] Setting LED always OFF
1: always OFF
0: if MODE 2 has been set the LED will blink when OPCNT=CNTELPS

[1] Setting Stop Blinking LED
1: stop blinking

[0] Reboot firmware
1 = reboot request

**CNTELPS**
16 bit unsigned counter.
Specify the number of times of the reed switch opening before to send an uplink event.

**TMELPS**
24 bit unsigned seconds counter ONLY for MODE 3.
It specifies the sampling period on MODE 3.
Once TMELPS is elapsed, the sensor sends a message on port 20 to the server and then it is reset to 0 ready for a new counter period.
Programmed values below 15 sec, will be forced to 15. Values above n*15 and below (n+1)*15 will be forced to (n+1)*15 [sec] so the minimum setting timing period is 15 seconds with multiple of 15 seconds.
Allowed TMELPS value is comprised from 15 to 16777215 sec (about 194 days)
Example:

- The server send TMELPS at 10 seconds then the sensor will force this at 15 seconds
- The server send TMELPS at 31 seconds then the sensor will force this at 45 seconds

**TMALIVE**
16 bit unsigned seconds counter.
It specifies the number of seconds between two ALIVE events.
TMALIVE counter is set to 0 at the end of every transmission of the ALIVE event.
Programmed values below 15 secs, will be forced to 15
Values above n*15 and below (n+1)*15 will be forced to (n+1)*15 [sec] so the minimum setting timing period is 15 seconds with multiple of 15 seconds.
Allowed TMALIVE value is comprised from 15 to 65536 seconds
Default TMALIVE of IR868LR and IRUS915LR is 50 minutes (3000 sec)
E.g.:

- The server sets TMALIVE at 9 seconds then the sensor will force this at 15 seconds
- The server sets TMALIVE at 40 seconds then the sensor will force this at 45 seconds

**TTHRHI**
Float. Temperature Threshold High
Temperature value composed by 4 byte with 2digit, 2 digit format (e.g. 25,00°C)

**TTHRLO**
Float. Temperature Threshold Low
Temperature value composed by 4 byte with 2digit, 2 digit format (e.g. 18,50°C). This parameter is used when two thresholds are required

**HTHRHI**
Float. Relative Humidity Threshold High
Relative humidity value composed by 4 byte with 2digit, 2 digit format (e.g. 75,00%)

**HTHRLO**
Float. Relative Humidity Threshold Low
Relative humidity value composed by 4 byte with 2digit, 2 digit format (e.g. 55,50% ). This parameter is used when two thresholds are required

**ACK/NACK**
3 bytes Char <ACK> and 4 bytes Char <NACK>)
Acknowledgement signal sent by the sensor to server. It can be Ack or Nack depending whether or not the message received from the server is fine. It also specifies on which protocol port the message has been received from the server. This message is sent every time the sensor receive a setting message from the server on ports 9, 12, 13, 30 and 31.

**HYSTT and HYSTH**
4 bytes float.
This parameter sets the hysteresis value for temperature and humidity.
Value HYSTT (temperature) is integer decimal while HYSTH (humidity) is percentage.

**NUMTHRL**
1 Byte integer.
This parameter sets the number of temperature and humidity thresholds (one or two) wanted

| | *MSB 7* | *6* | *5* | *4* | *3* | *2* | *1* | *0 LSB* |
|---|---|---|---|---|---|---|---|---|
| *Bit value* | **0** | **0** | **0** | **0** | **0** | **0** | **x** | **x** |

[7:2] Unused
To be defined

[1] Second threshold
1 = second threshold activated
0= second threshold deactivated

[0] First threshold
1 = first threshold activated – Default value
0 = first threshold deactivated. **Set it to 0 if second threshold is activated**

**CNFGRGST**

16 bit total.

This parameter is used in order to reset the OPCNT counter and to send the OPCNT value if the CNTELPS threshold is reached (only MODE 3)

| | MSB 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 LSB |
|---|---|---|---|---|---|---|---|---|
| Bit value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | MSB 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 LSB |
| Bit value | 0 | 0 | 0 | 0 | 0 | 0 | x | x |

[15:2] Unused
To be defined

[1] OPCNT value (only for MODE 3)
1 = send OPCNT value if OPCNT=CNTELPS
0 = OPCNT value is NOT sent (defaut value)

[0] OPCNT counter reset (MODE 2 and MODE 3)
1 = reset request
0 = OPCNT is not reset (default value)

**MODE**

Unsigned char (8 bits).

MODE is defined as following:

- **MODE 1** (<01>): In this mode every change of the reed switch status generates an uplink message. The opening counter OPCNT on the transmitted message is incremented by 1 and never reset. Only the door opening is calculated, so after two events (opening and closing door) the OPCNT is incremented only by 1.

- **MODE 2** (<02>): In this mode, the sensor generates an uplink message only when the specified counter CNTELPS is reached. Allowed CNTELPS number is comprised from 1 to 65535. Only the aperture of the magnetic contact is counted. At the end of the uplink message, the OPCNT counter can be reset using CNFGRGST

- **MODE 3** (<03>): In this mode, the sensor generate an uplink message when the specified time TMELPS is elapsed. Allowed TMELPS (24bit) number is comprised from 15 to 16777215 seconds. Programmed values below 15 secs, will be forced to 15. Minimum setting timing period is 15 seconds with multiple of 15 seconds. Only the aperture of the magnetic contact will be counted.
  The uplink message can be generated also based on CNTELPS, this means the sensor will send a message when OPCNT=CNTELPS, so independently from TMELPS.
  At the end of any uplink message, the OPCNT counter can be reset using CNFGRGST.

# 5 Message from contact magnetic sensor to server

Different ports on LoRaWAN protocol are used to transmit messages to the server.

Port assignments as follow:

| From sensor to server | Port # |
|---|---|
| Presentation | 5 |
| Serial Number | 6 |
| FW release, library release, HW release | 7 |
| Battery level | 8 |
| Alive | 9 |
| ACK | 10 |
| Specific Sensor Information message | 30 |

**Port 5 message specification:**

The payload contains the sensor model string in ASCII format (e.g. CM868LR)
The presentation message is sent every times the magnetic contact performs a reboot.
The reboot is caused by power-on or by a server command.
The presentation message is provided also if the server send an “Enq” on port 5. See chapter 5.

**Port 6 message specification:**

The payload contains the serial number string in ASCII format. 8 bytes length
(e.g. AA112233445566FF)
The Serial Number message is provided if the server send an “Enq” on port 6. See chapter 5.

**Port 7 message specification:**

The payload of 11 bytes ASCII format contains:

- the FW release (3 bytes; e.g.: 0.5);
- the LoRaWAN client library release (5 bytes; e.g.: 3.4.1);
- the HW release (1 byte; e.g.: B)
- Fields are separated by char comma “,”

Example:
the HEX payload for the above examples is
30 2e 35 2c 33 2e 34 2e 31 2c 42 (ASCII: **0.5,3.4.1,B)** (11bytes total)
The FW release, LoRaWAN client library version, HW release message message is provided if the server send an “Enq” on port 7. See chapter 5.

**Port 8 message specification:**

The payload of 1 byte contains battery type and current percentage charging level. Refer to BATTERY parameter definition at chapter 3.
The battery Level message is sent spontaneously on port 8 if the battery charging level is below 25%.
The Battery Level message is provided if the server send an “Enq” request on port 8. See chapter 5.

Port 9 message specification:

The payload contains the ALIVE message. Refer to ALIVE parameter definition at chapter 3
The ALIVE message is sent every 50 minutes as default timing value. Refer to chapter 5 to change this value from the server.

Port 10 message specification:

The payload is 4 bytes and contains the Ack/Nack message to be sent to the server every time a setting message is received by the sensor on ports 9, 12, 30 and 31.
If the message received from the server is fine then an Ack message is sent otherwise a Nack message is transmitted.
The total message contains also the port number on which the sensor received the message from the server.

- The Ack message has the following structure (4 Byte total):

| Name | Type | Function |
|---|---|---|
| Ack | Unsigned char (24 bits) | Ack message |
| PORTNUMB | Unsigned int (8 bits) | Specify the port number where the message has been received |

E.g.:
- 41636b0c → 4 Byte total message
- 41636b HEX → <Ack> ASCII 3Byte
- 0c → <12> Int Dec 1Byte

*In this case the sensor is acknowledging the server to have received a good message on port 12*

- The Nack message has the following structure (5 Byte total):

| Name | Type | Function |
|---|---|---|
| Nack | Unsigned char (32 bits) | Nack message |
| PORTNUMB | Unsigned int (8 bits) | Specify the port number where the message has been received |

E.g.:
- 4e61636b09 → 5 Byte total message
- 4e61636b HEX → <Nack> ASCII 3Byte
- 09 → <9> Int Dec 1Byte

*In this case the sensor is acknowledging the server to have received a corrupted/bad message on port 9.*

Port 30 **CM868LR** message specification:
The payload is in raw format (3 Bytes)

| Name | Type | Function |
|---|---|---|
| EVENT | Unsigned char (8 bits) | Event flag. See chapter 3 |
| OPCNT | Unsigned int (16 bits) | Opening counter |

The sensor sends **spontaneously** a message to the server if:

- If the tamper switch change its status (only from CLOSE to OPEN)
  The tamper status is reported into the EVENT parameter transmitted

- If the battery level reaches the 25% of the full charge. If this the case the sensor will transmit every 60min the battery value after the first advise.

- If **MODE 1** has been set by the server (see chapter 3) the sensor sends OPCNT value to the server at every change of the reed switch status. OPCNT is never reset.

- If **MODE 2** has been set by the server (see chapter 3) the sensor sends a message to the server if *OPCNT = CNTELPS (the measured opening counters equals the counter threshold).* OPCNT can be reset using CNFGRST parameter once the message has been sent to the server.

- If **MODE 3** has been set by the server (see chapter 3) the sensor sends a message to the server if

  1. *OPCNT = CNTELPS (the measured opening counters equals the counter threshold)*
     OPCNT can be reset using CNFGRST parameter once the message has been sent to the server. TMELPS is not reset.

  **OR**

  2. If *TMELPS is elapsed*, the sensor sends the message and then it is reset ready for a new counting period.
     OPCNT can be reset using CNFGRST parameter once the message has been sent to the server.

Port 30 **CM868LRTH** message specification:
The payload is in raw format (11 Bytes)

| Name | Type | Function |
|---|---|---|
| EVENT | Unsigned char (8 bits) | Event flag |
| OPCNT | Unsigned int (16 bits) | Opening counter |
| TVALUE | Float (4 bytes) | Temperature value |
| RHVALUE | Float (4 bytes) | Relative Humidity value |

The sensor sends **spontaneously** a message to the server if:

- If the tamper switch change its status (only from CLOSE to OPEN)
  The tamper status is reported into the EVENT parameter transmitted

- If the battery level reaches the 25% of the full charge. If this the case the sensor will transmit every 60min the battery value after the first advise.

- If TVALUE = (TTHRHI + HYSTT)
  (measured temperature equals high temperature threshold + HYSTT due to hysteresis value), the sensor sends TVALUE. In this case, if the measured temperature decreases the sensor sends a new TVALUE when TVALUE=TTHRHI.

  OR

  If NUMTHRL is set in order to have **two** temperature thresholds and
  TVALUE = (TTHRLO – HYSTL)
  (measured temperature equals low temperature threshold - HYSTL due to hysteresis value), the sensor sends a message. In this case, if the measured temperature increases the sensor sends a new TVALUE when TVALUE=TTHRLO.

- If RHVALUE = (HTHRHI + HYSTH)
  (measured humidity equals high relative humidity threshold + HYSTH due to hysteresis value), the sensor sends a message. In this case, if the measured temperature decreases the sensor sends a new RHVALUE when RHVALUE=HTHRHI.

  OR

  If NUMTHRL is set in order to have **two** temperature thresholds and
  RHVALUE = (HTHRLO – HYSTH)
  (measured humidity equals low relative humidity threshold - HYSTH due to hysteresis value), the sensor sends RHVALUE value. In this case, if the measured temperature increases the sensor sends a new RHVALUE when RHVALUE=HTHRLO.

- If **MODE 1** has been set by the server (see chapter 3) the sensor sends OPCNT value to the server at every change of the reed switch status. OPCNT is never reset.

- If **MODE 2** has been set by the server (see chapter 3) the sensor sends a message to the server if *OPCNT = CNTELPS (the measured opening counters equals the counter threshold).* OPCNT can be reset using CNFGRST parameter once the message has been sent to the server.

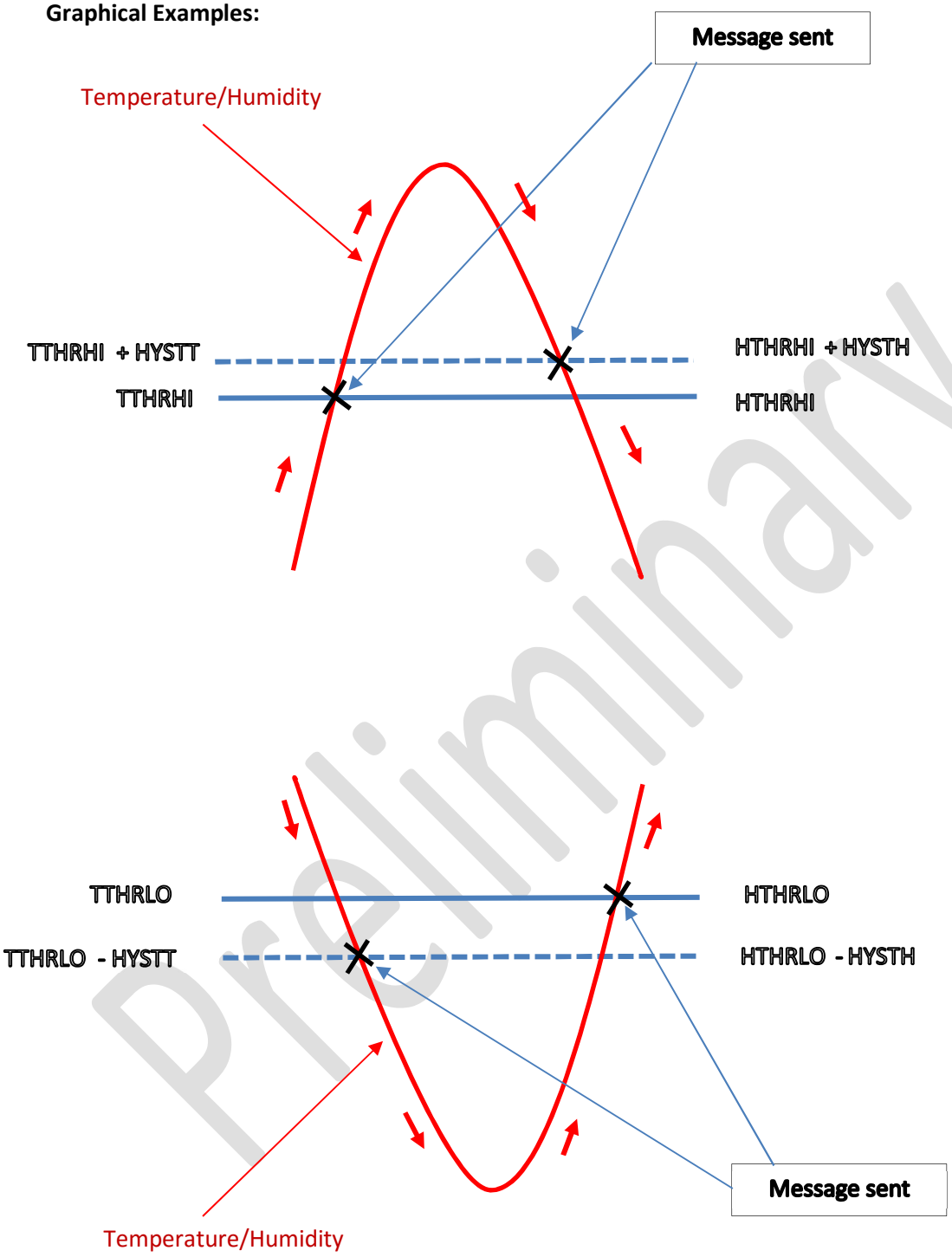
- If **MODE 3** has been set by the server (see chapter 3) the sensor sends a message to the server if

    3. O*PCNT = CNTELPS (the measured opening counters equals the counter threshold)* OPCNT can be reset using CNFGRST parameter once the message has been sent to the server. TMELPS is not reset.

    **OR**

    4. If *TMELPS is elapsed*, the sensor sends the message and then it is reset ready for a new counting period.
    OPCNT can be reset using CNFGRST parameter once the message has been sent to the server.

**Graphical Examples:**



# 6 Message from server to magnetic contact

The nature of LoraWAN class A permits to exchange messages only when the end-device transmits data to the server (uplink).
After sending the data, the end-device enable two RX windows to receive packets from the server.
In these windows, the server is able to send the data at the end node using specifics LoRaWAN protocol ports.
Downlink communications from the server at any other time different from the above mentioned RX windows, will have to wait until the next scheduled uplink occurs.

***Every message from server to sensor has 1 byte header that contains the total length of the message and 1 byte footer that contains the checksum. The checksum is calculated doing a logical XOR of all the bytes on the message except the last one, which is the checksum itself. Refer to Message Builder Toll to create and verify the right message to send to the sensor.***

| Header (1Byte) | Message on port 5,6,7,8,9,12,13,30,31 | Footer (1Byte) |
|---|---|---|

Port assignments as follow:

| Form server to sensor - Request for: | Port # |
|---|---|
| Presentation | 5 |
| Serial Number | 6 |
| FW release, library release, HW release | 7 |
| Battery level | 8 |
| Setting ALIVE interval TMALIVE | 9 |
| Setting MODE | 12 |
| FLAGS | 13 |
| Setting humidity upper and lower threshold | 30 |
| Setting temperature upper and lower threshold | 31 |

Ports 5, 6, 7 and 8 message specification:
This message is used by the server in order to have back from the sensor the required information. A text “Enq” message (HEX 456E71) must to be sent to one of these ports to ask the sensor for the required information. The sensor will reply with a message on the same port as reported in chapter 4.

Total message length 5 byte (3Byte for the <ENQ> text message and 1Byte header and and 1Byte footer)

**Example:**
To send and "Enq" on port 6 in order to receive back the Serial Number of the sensor, the server sends the HEX message:

05456E715F

Where

- 05: total length of the message 5Byte
- 456E71 → Enq (HEX to ASCII)
- 5F: Checksum

Port 9 specification:
Setting TMALIVE sampling ALIVE time period (2 Byte). See chapter 3.
Total message length 4 byte (including header and footer)

Port 12 specification:
Setting MODE "X" (X=1,2,3).
To set MODE, the server must send to the sensor a message on port 12.
**The total length of the message is 9 Byte on MODE 1 and MODE 2**: 1Byte header (total length of the message), 1Byte for MODE, 2Byte for CNFGRGST, 2Byte for CNTELPS, 2Byte for TMELPS and 1Byte for footer (checksum).
The payload as per the following table:

| Name | Type | Function |
|---|---|---|
| Header | Unsigned char (8bits) | Total length of the message in Byte |
| MODE | Unsigned char (8 bits) | Set the MODE:<br>Mode 1: see chapter 3 for more details<br>Mode 2: see chapter 3 for more details<br>Mode 3: see chapter 3 for more details |
| CNFGRGST | Unsigned int (16 bits) | Set the way to use OPCNT in MODE 2 and MODE 3 |
| CNTELPS | Unsigned int (16 bits) | Number of intrusions before to send the message |
| TMELPS | Unsigned int (16 bits) | Sampling period – ONLY MODE 3 |
| footer | Unsigned char (8bits) | Checksum |

Refer to chapter 3 for MODE, CNFGRGST, CNTELPS and TMELPS parameters definition.

**The total length of the message is 10 Byte on MODE 3**: 1Byte header (total length of the message), 1Byte for MODE, 2Byte for CNFGRGST, 2Byte for CNTELPS, 3Byte for TMELPS and 1Byte for footer (checksum).
The payload as per the following table:

| Name | Type | Function |
|---|---|---|
| Header | Unsigned char (8bits) | Total length of the message in Byte |
| MODE | Unsigned char (8 bits) | Set the MODE:<br>Mode 1: see chapter 3 for more details<br>Mode 2: see chapter 3 for more details<br>Mode 3: see chapter 3 for more details |
| CNFGRGST | Unsigned int (16 bits) | Set the way to use OPCNT in MODE 2 and MODE 3 |
| CNTELPS | Unsigned int (16 bits) | Number of intrusions before to send the message |
| TMELPS | Unsigned int (24 bits) | MODE 3 sampling period |
| footer | Unsigned char (8bits) | Checksum |

Refer to chapter 3 for MODE, CNFGRGST, CNTELPS and TMELPS parameters definition

**Example 1:**
to set MODE 2, without resetting OPCNT counter after the uplink and when OPCNT=CNTELPS=100 so after 100 reed detections, the HEX message the server must send on port 12 is

09020000006400006F → 9 Byte total

Where
- 09 HEX: total length of the massage (9 bytes) → 1 Byte
- 02 HEX: set MODE 2 → 1 Byte
- 0000 HEX: CNFGRGST all set to 0 → 2 Byte
- 0064 HEX: CNTELPS set to 100 → 2 Byte
- 0000 HEX: TMELPS set to 0 (to be used only in MODE 3. Any value is ignored by the sensor if not in MODE 3) → 2 Byte
- 6F HEX: checksum → 1 Byte

**Example 2:**
To set MODE 3, resetting OPCNT counter after the uplink, whit a timing threshold of 100 sec, the HEX message the server must send on port 12 is:

0A03000100000000646C → 10 Byte total

Where
- 0A HEX: total length of the massage (10 bytes) → 1 Byte
- 03 HEX: set MODE 3 → 1 Byte

- 0001 HEX: LSB of CNFGRGST register set to 1 to reset OPCNT after the transmission → 2 Byte
- 0000 HEX: CNTELPS set to 0 → 2 Byte
- 000064 HEX: TMELPS set to 100sec → 3 Byte
- 6C HEX: checksum → 1 Byte

**Example 3:**
To set MODE 3, resetting OPCNT counter after the uplink, with a timing threshold of 120 sec and a threshold of reed detections CNTELPS=80, the HEX message the server must send on port 12 is:

0A030001005000007820 → 10 Byte total

Where
- 0A HEX: total length of the massage (10 bytes) → 1 Byte
- 03 HEX: set MODE 3 → 1 Byte
- 0001 HEX: LSB of CNFGRGST register set to 1 to reset OPCNT after the transmission → 2 Byte
- 0050 HEX: CNTELPS set to 80 → 2 Byte
- 000078 HEX: TMELPS set to 120sec → 3 Byte
- 20 HEX: checksum → 1 Byte

Notice that if OPCNT=CNTELPS before than TMELPS expires, the sensor sends a message to the server but TMELPS is not reset

Port 13 specification:
Setting FLAGS parameters. Refer to FLAGS (16 bit) definition at chapter 3.
Total message length 4 byte (including header and footer)

**Example:**
To set a reboot of the sensor, the server must to sent the following HEX message:
04000105

Where:
- 04: total length of the message → 1 Byte
- 0001 → 0000 0001 (HEX to BIN) the LSB 1 indicates the reboot request → 2 Byte
- 05: checksum → 1 Byte

Port 30 specification:
With this message the server can set the thresholds high and low for humidity and the relative wanted hysteresis. Threshold can be 1 or 2.
Total message length is 15Byte
The message is composed by:

| Name | Type | Function |
|---|---|---|
| HTHRLO | Float (4 Bytes) | Relative Humidity Threshold Low |
| HTHRHI | Float (4 Bytes) | Relative Humidity Threshold High |
| HYSTH | Float (4 Bytes) | Hysteresis value |
| NUMTHRL | Unsigned int (1Byte) | Number of threshold wanted |

Refer to HTHRHI, HTHRLO, HYSTH and NUMTHRL parameter definitions at chapter 3
Total length of the message is 15B (including header and footer)

**Example:**
To set low RH threshold at 5%, high RH threshold at 45%, hysteresis 0.5% and two thresholds the HEX message to be sent to the sensor on port 30 is:

0F0000A04000003442000000 3F02A4 → 15 Byte

Where:

- 0F: 15Bytes total length → 1 Byte
- 0000A040: setting 5% as low RH threshold – float → 4 Byte
- 00003442: setting 45% as high RH threshold – float → 4 Byte
- 0000003F: setting 0.5% as hysteresis – float → 4 Byte
- 02: setting two thresholds → 1 Byte
- A4: checksum → 1 Byte

Port 31 specification:
With this message the server can set the thresholds high and low for temperature and the relative wanted hysteresis. Threshold can be 1 or 2.
Total message length is 15Byte
The message is composed by:

| Name | Type | Function |
|---|---|---|
| TTHRLO | Float (4 Bytes) | Temperature Threshold Low |
| TTHRHI | Float (4 Bytes) | Temperature Threshold High |
| HYSTT | Float (4 Bytes) | Hysteresis value |
| NUMTHRL | Unsigned int (1Byte) | Number of threshold wanted |

Float. Refer to TTHRHI, TTHRLO, HYSTT and NUMTHRL parameter definitions at chapter 3

**Example:**
To set low T threshold at 15°C, high T threshold at 55°C, hysteresis 0.3°C and two thresholds the HEX message to be sent to the sensor on port 31 is:

0F0000704100005C429A99993E0286 → 15 Byte

Where:

- 0F: 15Bytes total length → 1 Byte
- 00007041: setting 15°C as low T threshold – float → 4 Byte
- 00005C42: setting 55°C as high T threshold – float → 4 Byte
- 9A99993E: setting 0.3°C as hysteresis – float → 4 Byte
- 02: setting two thresholds → 1 Byte
- 86: checksum → 1 Byte

## Contact Information

**ASCOEL Srl**
**Via degli Artigiani 7 26025 Pandino -CR- ITALY**
**Phone: (+39)0373 970473**
**E-mail: info@ascoel.com**
**Internet: http://www.ascoel.it**