LoRaWAN™ Sensors

Application Layer Description

NOTICE

Nke Watteco reserves the right to make changes to specifications and product descriptions or to discontinue any product or service without notice. Except as provided in Nke Watteco's Standard Terms and Conditions of Sale for products, Nke Watteco makes no warranty, representation or guarantee regarding the suitability of its products for any particular application nor does Nke Watteco assume any liability arising out of the application or use of any product and specifically disclaims any and all liability, including consequential or incidental damages.

Certain applications using semiconductor products may involve potential risks of death, personal injury or severe property or environmental damage. Nke Watteco products are not designed, authorized or warranted to be suitable for use in life saving or life support devices or systems. Inclusion of Nke Watteco products in such applications is understood to be fully at the Customer's risk.

In order to minimize risks associated with the customer's application, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

Nke Watteco assumes no liability for applications assistance or customer product design. Nke Watteco does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of Nke Watteco covering or relating to any combination, machine or process in which such semiconductor products or services might be or are used. Nke Watteco's publication of information regarding any third party's products or services does not constitute Nke Watteco's approval, warranty and endorsement thereof.

Resale of Nke Watteco's products with statements of functionality different from or beyond the parameters stated by Nke Watteco for that product as defined by Nke Watteco's unique part number, voids all express and any implied warranties for that product, is considered by Nke Watteco to be an unfair and deceptive business practice and Nke Watteco is not responsible nor liable for any such use.

Embedded software is based on Nke Watteco proprietary drivers and applicative code and operates on the Contiki kernel from the SICS (Swedish Institute of Computer Science).

http://www.nke-watteco.com/

DOCUMENT HISTORY

Date	Revision	Modification Description
Mai 2015	1.0	Rewriting of nke Watteco's documentation : application layer description

CONTENTS

1 INTRODUCTION

This document presents the application layer protocol implemented in nke Watteco's sensors.
The application layer leverages on the ZigBee® Cluster Library (ZCL) binary format, release r02 (29th May 2008).

⚠ *This combination of ZCL on top of a low power LoRa network should not be confused with "ZigBee®" technology, which uses its own network layer, not based on LoRa and operating on 2.4GHz. Nke Watteco sensors range can be up to 20 times the range of typical ZigBee® sensors.*

Thus, in order to be as understandable as possible, the current document will first give a general description of the ZCL protocol used in nke Watteco's application layer. The standard and extended clusters used will be given as well as the supported commands, the attribute data types and the error codes.

Then, the document will go around the application protocol reference needed to have a complete comprehension of the application layer. In this part, the attributes and the applicable commands for each of these attributes will be given in details for each cluster.

Finally the "Use Guide" of the "br_uncompress" tool provided to uncompress the batch reports (cf. §3.3) will be given at the end of this document.

2 GENERAL PROTOCOL DESCRIPTION

This part describes the library used by the application layer: the “ZigBee Cluster Library (ZCL)”. The ZCL is very compact and optimized for radio transport, and offers the richest set of standard semantic objects of any automation protocol.More specifically, this part will go through an exhaustive list of the ZCL clusters (standard and extended) supported by nke Watteco’s sensors, the attribute data types and the error codes.

Some advices about the ISM band use will be given at the end of this part.

⚠ In order to be able to use the application layer of the nke Watteco’s sensors, it is necessary that the sensors are correctly connected to the network. In order to do that, it is highly recommended to read the documentation about the “LoRaWAN Public Network” as well as the sensor “User Guide”.

⚠ Take into consideration while sending commands that the target might be a sleeping node. Sleeping devices are able to receive commands only after periodic wake-up, so you will receive a response after a variable time. Commands are buffered in the network routers but these buffers may overflow: be prepared to resend un-acked commands.

⚠ Thus in order to format your commands, you will need to first have a look to the ZCL commands listed in §2.1.3 and then the frame format outlined in §3.

2.1.1 STANDARD CLUSTERS

Here is an exhaustive list of standard ZCL clusters currently supported by Nke Watteco IP Sensors:

- On/Off cluster (0x0006)
- Simple metering –like cluster (0x0052)
- Occupancy Sensing cluster (0x0406)
- Temperature Measurement cluster (0x0402)
- Relative Humidity Measurement cluster (0x0405)
- Analog Input (Basic) cluster (0x000C)
- Binary input (Basic) cluster (0x000F)
- Illuminance Measurement cluster (0x0400)
- Basic cluster (0x0000)
- MultiState output (0x0013)

2.1.2 EXTENDED CLUSTERS

Extended clusters were defined to support features not currently covered by the ZCL specification.

- Configuration cluster (0x0050)
- TIC information cluster / ERDF-ICE (0x0053)
- TIC information cluster / ERDF-CBE (0x0054)
- TIC information cluster / ERDF-CJE (0x0055)
- TIC information cluster / ERDF-STD (0x0056)

2.1.3 SUPPORTED COMMANDS

Only the following ZCL command frames are available:

- Read attributes
- Read attributes response
- Write attributes no response
- Configure reporting
- Configure reporting response
- Read reporting configuration
- Read reporting configuration response
- Report attributes

2.1.4 ATTRIBUTE DATA TYPES AND ENCODING

The following table describes the attribute data types for all attributes defined in the current implementation:

Label	*Type N°*	*Size (Bytes)*	*Comment*
BOOLEAN_TYPE	0x10	1	
GENERAL8_TYPE	0x08	1	For General types. Byte signification depends on the managed attribute.
GENERAL16_TYPE	0x09	2	"
GENERAL24_TYPE	0x0a	3	"
GENERAL32_TYPE	0x0b	4	"
BITMAP8_TYPE	0x18	1	For Bitmaps types. Bits signification depends on the managed attribute.
UINT8_TYPE	0x20	1	
UINT16_TYPE	0x21	2	
UINT32_TYPE	0x23	4	
INT8_TYPE	0x28	1	
INT16_TYPE	0x29	2	
INT32_TYPE	0x2b	4	
UINT8_ENUM	0x30	1	
CHAR_STRING	0x42	1+n	<size>[char1,char2, …]
BYTES_STRING	0x41	1+n	<size>[byte1,byte2, …]
LONG_BYTES_STRING	0x43	2+n	<sizeH><sizeL>[byte1,byte2, …]
STRUCTURE_ORDEREDSEQUENCE	0x4C	2+n	2 first bytes are the size of the following byte string. n depends of the managed attribute
SINGLE_TYPE	0x39	4	The single format representation is managed as follows : 8bits Exponent, 1 sign bit, 23 bits fractional “eeeeeeee s ffffffffffffffffffff”

Note: Any multi-byte numerical value is serialized in “Big Endian” order inside the ZCL-Like payload.

2.1.5 ERROR CODES

Command responses may return error codes within the status field of the ZCL response frame (Cf §3.4). The main error codes are:

0x80	Malformed command	Most of the time the command as got wrong number of fields.
0x81	Unsupported cluster command	The "cluster specific command" is not supported by the cluster
0x82	Unsupported general command	The "General command" is not supported by the cluster
0x86	Unsupported attribute	The attribute is not supported or the command does not apply to the attribute.
0x87	Invalid field	One of the parameter fields is invalid. *Frequent cause: Max < Min in configure reporting.*
0x88	Invalid value	The request is incorrectly formatted. *Frequent causes: Unexpected frame size, or Bad attribute type or on batch case if not specific batch is not available*
0x89	Insufficient space	The parameter or the expected response is too big. *Typical cause: attempting to read a very large attribute which is designed to be read through filters and scope specifiers, typically the TIC cluster. The client should limit the number of requested result fields.*
0x8c	Unreportable attribute	The "configure reporting" or "read reporting configuration" is not supported by the selected attribute. *Either the attribute does not exist in the cluster or it is not reportable.*
0xC2	Batch report: No free slot	The required batch configuration could not be added. *The number of batch reportable slots is currently limited to 8.*
0xC3	Batch report: Invalid Tag size.	At least one other batch report configuration uses a different Tag size. *Use the same tag size for all batches configured on a device.*
0xC4	Batch report: Duplicate tag label	A requested Tag label is already used by another batch report configuration. *Choose another tag label.*
0xC5	Batch report: Label out of range	Tag lbl is greater than possible according to Tag size. *Tag label must be a number between : 0 and ($[2^{(Tag\ size)}] - 1$)*

2.1.6 PAYLOAD SIZE WARNINGS

⚠ Nke Watteco's sensors uses ISM unlicensed radio bands. However the ISM band is subject to maximum power limitations and maximum average duty cycle limitations (Figure 1). While the maximum power limitation is enforced by design, the duty cycle may be affected by attribute reporting user configuration.

⚠ Users must make sure, while configuring reporting, that the resulting duty cycle (i.e. fraction of time the radio spends in active transmission) remains below 1% (0.1% on specific frequency bands) for each sensor individually. In addition, for the correct functioning of radio channel access, the overall duty cycle of all sensors combined should not exceed about 5%.

While this duty cycle limit will not be reached easily when using high bitrate modulations, it becomes an important checkpoint for low bitrate, long range transceiver configurations such as LoRa at 1kbps.

Edge Frequencies		Field / Power	Spectrum Access	Band Width
Fe-	Fe+			
865 MHz	868 MHz	+6.2 dBm / 100 kHz	1 % or LBT AFA	3 MHz
865 MHz	870 MHz	-0.8 dBm / 100 kHz	0.1% or LBT AFA	5 MHz
868 MHz	868.6 MHz	14 dBm	1 % or LBT AFA	600 kHz
868.7 MHz	869.2 MHz	14 dBm	0.1% or LBT AFA	500 kHz
869.4 MHz	869.65 MHz	27 dBm	10 % or LBT AFA	250 kHz
869.7 MHz	870 MHz	7 dBm	No requirement	300 kHz
869.7 MHz	870 MHz	14 dBm	1 % or LBT AFA	300 kHz

FIGURE 1: ETSI 868MHZ BAND POWER AND DUTY CYCLE RESTRICTIONS

See also remarks about ZCL TIC Payload sizes that can become very large when not appropriately filtered: "§3.4.16".

3 APPLICATION PROTOCOL REFERENCE

To manage nke Watteco's sensors, it is necessary to use the following Application Protocol based on the ZCL. Sub-chapters of §3.4 group all existing ZCL frame document according to each cluster. Chapters §3.1 and §3.3 show global information for End point management and Batched Compressed reporting capabilities.

3.1 ENDPOINT NUMBER MANAGEMENT

The application protocol implements the notion of "Endpoint number" to enable multiple instances of each cluster type on a given device. Different clusterIDs may share a common endpoint number but two instances of a given clusterID must have different endpoint numbers.

The "Endpoint number" bits are distributed in the first byte as described below. Hence an endpoint number can take 32 different values. For compatibility reason with previous version Flag 0x11 can still be used to identify default endpoint number 0.

Flag bitfield is encoded as follows:

Endpoint number (0-31)			Reserved (Must keep these value)		Endpoint number (0-31)		!Batch
bit2	bit1	bit0 (LSB)	1	0	bit4(MSB)	bit3	1

Note: All frames accepting "Endpoint number selection", are identified with a "0x11+en" flag. Frames that do not support "Endpoint number selection" are identified with a "0x11" flag value.

Due to "Compressed Batch reporting" capabilities, **the low significant bit of the Flag set to 0 is used to identify a "Batch report" frame**. The flag semantic is then completely changed. See for §3.3.2 for more details.

Batch specific (Cf §3.3.2)							!Batch
x	x	x	x	x	x	x	0

3.2 MINIMUM AND MAXIMUM INTERVAL FIELD

The minimum and maximum interval fields are on two bytes and define an interval in minutes or seconds depends to the first MSB Bit as defined below:

Minimum or Maximum interval															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
1: min or 0: sec	Interval														

The specific values 0xFFFF; 0x0000; 0x8000 does not define interval.

3.3 BATCH MANAGEMENT

The application layer implements the notion of "Batch" to aggregate AND compress several samples of measure in the same reporting frame. Not all attributes are batch reportable.

The list of attribute/fields that can be inserted in the batch report of a sensor, is be described, for each cluster, in section §3.5.

3.3.1 BATCH REPORTING CONFIGURATION

Formats of the "*configure reporting*" and "*Read reporting configuration response"* are different:

⇨ The usual "*Direction*" field which is considered as "*Batch*" field.

Configure reporting command:

Flag+en	CommandID	ClusterID	Batch (Size + Flag)	Attribut	…

Below batched parameters List repeated for each reportable field

…	Field Index	Minimum recording Interval	Maximum recording interval	Delta Recordable change (format depends of the selected Field index)	Resolution	Tag

Read reporting configuration request:

Flag+en	CommandID	ClusterID	Status	Batch	Attribut	…

Read reporting configuration response:

Flag+en	CommandID	ClusterID	Status	Batch (Size + Flag)	Attribut	…

Below batched parameters List repeated for each reportable field

…	Field Index	Minimum recording Interval	Maximum recording interval	Delta Recordable change (format depends of the selected Field index)	Resolution	Tag

"Direction" field becomes "Batch" field and is now encoded as follow:

Size of Batched Parameters List							Batch Flag
Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	0/1

When the "Batch Flag" field is `0x00` then the reporting is as before. Put the "*Batch Flag*" bit of field to `0b.......1` to:

⇨ Ask the current batch reporting configuration of the EndPoint, Cluster and Attribute with "*Read reporting configuration request*". In that case, the "Size of Batched Parameters List" field is unused and should be set to 0.

⇨ Set the configured EndPoint, Cluster and Attribute in the batch reporting frame with "*Configure reporting command*". In that case, the "Size of Batched Parameters List" field indicates the size of "Batched Parameters List".

Note: Only EndPoint, Cluster, Attribute and Value, representing numerical (Integer or float) are available for "batch reporting", please report to each Cluster/Attributes reportable values (§ 3.4) to know the possibility.

When Batch reporting is selected, fields of the "*configure reporting*" are considered as below:

- **Field index**: Each batch configuration is associated to a specific cluster/Endpoint/Attribut however some clusters have multifield reportable attributes. The "Field index" parameter allows selecting the field to be reported in the attribute. Most of the time this 1 byte "field index" is 0, like in clusters Temperature, Humidity, Binary, … But sometime it is necessary to define the requested field inside the attribute like in clusters Configuration, Simple metering or TIC.
  A "batch reportable" attribute will present its possible reportable field(s) in each "Configure reporting" command of chapter §3.5.

- **Minimum record interval**: minimum elapsed time since last "recording", to accept "recording" a new sample.

- **Maximum record interval**: maximum time without "recording" a new sample. When this time is elapsed, a new sample is automatically "recorded".
  The batch report is done on the minimum of the Maximum record interval of all batch's configuration.
  If "Batch" is set on more than one Cluster/Attribute/Endpoint then the effective "Maximum reporting interval" of the batch is configured to the minimum of them.

- **Recordable change**: minimum change on which a dated sample is recorded. The Data type of this field is the same than the reportable value itself (cf. § 3.4)

- **Resolution:** New field which has the same format of the Recordable change and specify the resolution of the returned measure. The Data type of this field is the same than the reportable value itself (cf. § 3.4)

- **Tag**: New field which will define the "Label" representing the compressed measure in the "*batch reporting*" frame. This field is one byte decomposed in two fields: `0xrllllnnn`. b2-b0 bits are the number of bits used for the Label size. The Label itself is contained in the b3-b6 bits. Even if a batch reporting could contain up to 2^4 (16) different tags (different reported values), it will be pertinent to have only a few attributes values for a batch reporting. The client should limit this according to the overall radio performances, current protocol, sampling parameter requirement and physical profile measurement.

  Tag requirements according to reports measures in a batch:
  - **All Tags have to be the same size**
  - **All defined Labels have to be different.**
  - It is important to dimension the size of the Tag related to the number of measures which will be batched. Longer the tag is, bigger the "batch reporting" will be.

  Example of Tag field: 0b00101011 gives a size of the Tag on 3 bits and the label will be 101. The maximum number of label in this case is 8.

Notes:

- A "Configure reporting command" may be refused, with various error codes in the response, according to its internal restrictions (See §2.1.5).

- At each event (new sample), according to "*Reportable change*", the length of the batch is calculated and if the length is higher than the maximum size of one frame, a report will be done.

- A specific command is added to the Configuration cluster (Cf §3.4.2) to ask for currently configured batch reports. This will induce one "*Read reporting configuration response"* for each Ep/Cluster/Attribute that composes the current "Batch reporting" configuration.

- Another specific command is added to the Configuration cluster (Cf §3.4.2) to ask a lost batch. Each batch is sent with a counter which is incremented from 0 to 7.

  Suggested client behavior: If no frame is received after the minimum of maximum recordable interval time, or if two consecutives batches are received with non consecutives counter, it will be interesting to ask the lost batches.

3.3.2 BATCH REPORTING FRAME FORMAT

Each "*batch reporting*" frame has a format composed of <**Header part**> and <**Samples part**>. Decided upon real time compression performance analysis, the **<Samples part>** may present two different formats (Cf. **§3.3.2.2** or **§3.3.2.3**). Last a **<Queue part>** is added to the batch compressed frame containing mainly the internal clock time when the frame is sent.

Discriminating usual zcd frames from batch report frames is done through the bit0 of the Flags:

- ➔ 1: Usual ZCD frame (report or not)
- ➔ 0: Batch report frame.

About batch reports: In order to make easier the uncompression of the batch reports, nke Watteco provides a software tool called br_uncompress. This tool allows to get back the samples send in the batch report with their time stamp. The description of this tool and the way of using it are explained in §4

3.3.2.1 <HEADER PART>

Flags	0bnnnnret0	0bnnnn....: number of type of measure included in the batch 0b....r...: set to 1 if the batch is a requested from "get a specific batch" cmd 0b.....e..: set to 1 if no sample part => no Coding Type in the Header size 0b......t.: set to 1 if Time Stamp is common for all the batch 1: <samples part 1> 0: <samples part 2>
Batch Counter	0bccc	Batch counter on three bits. Each time a batch is reported, the counter increments to 1.
Reserved	0b0	1 bit reserved for future evolution. Set to 0.
Tag Measure[0]	0b...	Label of the following measure
TimeStamp(0)[0]	0xXXYYZZQQ	TimeStamp of the first sample of the Tag.
Measure(0)[0]	0xYY..YY	Number of bytes depends on the real measure type (See cluster specific information).
Coding Type[0]	0bwxyz	0bwx.. => define the Type of Huffman table. 00 : ALDC Huffman Table 01 : Positive Huffman Table 10 : Negative Huffman Table 0b..yz => define the specific Table. 00: Table A 01: Table B 10: Table C
…		
Tag Measure[n]	0b...	Label of the following measure
TimeStamp(0)[n]	H(0)|index(0) or H(0)|TimeStamp(0)[n]	TimeStamp of the first sample of the Tag. It is the compressed delta with the previous header's timestampt based on the Huffman table B. *TimeStamp(0)[n]= TimeStamp(0)[n-1]+index(i)-1+2^b(i)*
Measure(0)[n]	0xYY..YY	Number of bytes depends on the real measure type (See cluster specific information).
Coding Type[n]	0bwxyz	0bwx.. => define the Type of Huffman table. 00 : ALDC Huffman Table 01 : Positive Huffman Table 10 : Negative Huffman Table 0b..yz => define the specific Table. 00: Table A 01: Table B 10: Table C

3.3.2.2 <SAMPLES PART 1> "COMMON TIMESTAMP SERIES FOR ALL GROUP OF MEASURE"

The second bit of <Header>/flags must be 1. (Ie: Flags mask: 0b.....1.)

NumberOfSample	0xnn	Number of following sample in the batch
TimeStamp Coding Type	0bxy	0b00 => Positive Huffman coding Table A 0b01 => Positive Huffman coding Table B 0b10 => Positive Huffman coding Table C
Delta TimeStamp(i)	H(i)|index(i) or H(i)|TimeStamp(i)	H(i): huffman coding corresponding to the "TimeStamp Coding Type" Table. From the Huffman Table, extract b(i) corresponding to the number of bit of index(i). *TimeStamp(i)= TimeStamp(i-1)+index(i)-1+2^b(i)*
..		
Delta TimeStamp(0xnn-1)		
Tag Measure[x]		
Delta Value(i)[x]	0bs|(H(i)|index(i))[x] or 0bs|H(i)|Measure(i)[x]	0b0: no Value corresponds to the TimeStamp(i). Value(i)[x] is empty. 0b1: The Delta Value corresponds to the TimeStamp(i) and is defined in step of Resolution: H(i): huffman coding corresponding to the "Coding Type[x]" Table. From the Huffman Table, extract b(i) corresponding to the number of bit of index(i). *If the Table is a Positive Huffman then use:* *Value(i)[x]= Value (i-1)[x]+index(i)-1+2^b(i)* *If the Table is a Negative Huffman then use:* *Value(i)[x]= Value (i-1)[x]-index(i)+1-2^b(i)* *If not then use:* *If index(i) >= 2^(b(i)-1) then* *Value(i)[x]= Value (i-1)[x]+index(i)* *If not* *Value(i)[x]= Value (i-1)[x]+index(i)+1-2^b(i)* *Measure(i)[x]=Resolution*Value(i)[x]*
…		
Delta Value(0xnn-1)[x]	0bs|(H(0xnn-1)|index(0xnn-1))[x] or 0bs|H(0xnn-1)|Measure(0xnn-1)[x]	
Tag Measure[x']		
Delta Value(i)[x']	0bs|(H(i)|index(i))[x'] or 0bs|H(i)|Measure(i)[x']	
…		
		**i = [0;0xnn-1]*

3.3.2.3 <SAMPLES PART 2> "WITH TIMESTAMP SERIES FOR EACH GROUP OF MEASURE"

The second bit of flags must be 0. (Ie: Flags mask: `0b.....0.`)

Tag Measure[x]	`0b…`	Tag from [0] to [n]
NumberOfSample[x]	`0xnn`	Number of following sample for this specific measure
TimeStamp Coding Type[x]	`0bxy`	`0b00` => Positive Huffman coding Table A `0b01` => Positive Huffman coding Table B `0b10` => Positive Huffman coding Table C
Delta TimeStamp(i)[x]	H(i)|index(i) or H(i)|TimeStamp(i)[x]	H(i): huffman coding corresponding to the "TimeStamp Coding Type" Table. From the Huffman Table, extract b(i) corresponding to the number of bit of index(i). *TimeStamp(i)[x]= TimeStamp(i-1)[x]+index(i)-1+2^b(i)*
Delta Value(i)[x]	(H(i)|index(i))[x] or H(i)|Measure(i)[x]	The Delta Value is defined in step of Resolution: H(i): huffman coding corresponding to the "Coding Type[x]" Table. From the Huffman Table, extract b(i) corresponding to the number of bit of index(i). *If the Table is a Positive Huffman then use:* *Value(i)[x]= Value (i-1)[x]+index(i)-1+2^b(i)* *If the Table is a Negative Huffman then use:* *Value(i)[x]= Value (i-1)[x]-index(i)+1-2^b(i)* *If not then use:* *If index(i) >= 2^(b(i)-1) then* *Value(i)[x]= Value (i-1)[x]+index(i)* *If not* *Value(i)[x]= Value (i-1)[x]+index(i)+1-2^b(i)* *Measure(i)[x]=Resolution*Value(i)[x]*
…		
Tag Measure[x']		
TimeStamp Coding Type[x']	`0bxy`	
NumberOfSample[x']	`0xnn`	Number of following sample for this specific measure
Delta TimeStamp(i')[x']	H(i')|index(i')	
Delta Value(i')[x']	(H(i')|index(i'))[x'] or H(i')|Measure(i')[x']	
…		

3.3.2.4 <QUEUE PART>

TimeStamp	H(i)|index(i) or H(i)|TimeStamp or 0xXXYYZZQQ	TimeStamp of the sent frame. The value is based on the delta from the last calculated TimeStamp of the frame and on the Huffman table B. *TimeStamp= Last_TimeStamp+index(i)-1+2^b(i)* If the number of type of measure in the headr's flags is set to 0 then the timestamp is on 4 Bytes without compression
Pad8	0b0…	Padding of 0 to stop the payload on a natural number of Bytes.

The last Time Stamp of the batch corresponds to the time which the sensor builds and reports the frame. All timestamps are set according to the internal relative clock of the sensor (seconds since last sensor bootstrap).

3.3.2.5 HUFFMAN TABLES

b(i)	H(i) Table A	H(i) Table B	H(i) Table C	ALDC D(i)	Positive D(i)	Négative D(i) [-1x]
0	00	1101111	1001	0	0	0
1	01	11010	101	-1,+1	1,2	1,2
2	11	1100	00	-3-2+2+3	3,4,5,6	3,4,5,6
3	101	011	01	-7,..,-4,+4,..,+7	7,..,14	7,..,14
4	1001	111	11	-15,..,-8,+8,..+15	15,..,30	15,..,30
5	10001	10	10001	-31,..,-16,+16,..+31	31,..,62	31,..,62
6	100001	00	100001	-63,..,-32,+32,..+63	63,..,126	63,..,126
7	1000001	010	1000001	-127,..,-64,+64,..+127	127,..,254	127,..,254
8	10000001	110110	10000001	-255,..,-128,+128,..+255	255,..,510	255,..,510
9	1000000000	110111011	1000000000	-511,..,-256,+256..+511	511,..,1022	511,..,1022
10	10000000010	110111001	10000000010	-1023,..,-512,+512,..+1023	1023,..,2046	1023,..,2046
11	10000000011	1101110101	10000000011	-2047,..,-1024,+1024,..+2047	2047,..,4094	2047,..,4094
12	10000000100	1101110100	10000000100	-4095,..,-2048,+2048,..+4095	4095,..,8190	4095,..,8190
13	10000000101	1101110000	10000000101	-8191,..,-4096,+4096,..+8191	8191,..,16382	8191,..,16382
14	10000000110	11011100011	10000000110	-16383,..,-8192,+8192,..+16383	16383,..,32766	16383,..,32766
15	10000000111	11011100110	10000000111	D(i) is not available: in this case the measure follows the index. Length depends on the real measure type.		

D(i) is the difference with the previous value.

3.4 MESSAGE SET BY CLUSTER

The following sections describe the available messages for each cluster.

See §2.1.4 for possible "Attribute type" values.
See §2.1.5 for possible "Status" values.

3.4.1 BASIC CLUSTER

Clusters	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Basic	Firmware Version	Read attribute request	0x11	0x00	0x00 0x00	0x00 0x01				
		Read attribute response	0x11	0x01	0x00 0x00	0x00 0x01	0x00	0x0b	0xMM 0xmm 0xrr 0xrc *0xMM: Major, 0xmm: Minor, 0xrr: Revision, 0xrc: RCBuild* *Ex:1/5/0/1 => 1.5.0 .1* *=> Source code defined*	Application version number
	Kernel Version	Read attribute request	0x11	0x00	0x00 0x00	0x00 0x03				
		Read attribute response	0x11	0x01	0x00 0x00	0x00 0x03	0x00	0x42	0xss "C … C" C … C: Up to 16 bytes *Default: "Contiki 2.5"*	Kernel name 0xss: Size of string
	Manufacturer	Read attribute request	0x11	0x00	0x00 0x00	0x00 0x04				
		Read attribute response	0x11	0x01	0x00 0x00	0x00 0x04	0x00	0x42	0xss "C … C" C … C: Up to 16 bytes *Default: "WATTECO"* *➔ Set during production or Source code defined to save ROM space*	Manufacturer name 0xss: Size of string
	Model identifier	Read attribute request	0x11	0x00	0x00 0x00	0x00 0x05				
		Read attribute response	0x11	0x01	0x00 0x00	0x00 0x05	0x00	0x42	0xss "C … C" C … C: Up to 16 bytes *Default: "IPS_TTTT-WXYZ.[F]xx"* *➔ Should be set during production set according to manufacturer nomenclature (Cf Watteco)* *Ex: IPS_SPG-2123.F10*	Identify uniquely the full system with all embedded PCBs
	Date Code	Read attribute request	0x11	0x00	0x00 0x00	0x00 0x06				*Date of manufacturer of the device. Production date.*
		Read attribute response	0x11	0x01	0x00 0x00	0x00 0x06	0x00	0x42	0x08"YYYYMMDD" *Default: "YYYYMMDD" (Can be the date of TAG of the released version)* *➔ Set during production according to … date of production …*	0x08: character string size YYYY : year, MM: month, DD: day
	Location description	Read attribute request	0x11	0x00	0x00 0x00	0x00 0x10				*Information aboutsensor position (sematique, GPS, etc)*
		Read attribute response	0x11	0x01	0x00 0x00	0x00 0x10	0x00	0x42	0xss"C … C" *Default: "Unknown"*	0xss: character string size "C … C": Up to 8 bytes
		Write attribute no response	0x11	0x05	0x00 0x00	0x00 0x10		0x42	0xss "C … C"	0xss: character string size "C … C": Up to 8 bytes
	Appli. Name	Read attribute request	0x11	0x00	0x00 0x00	0x80 0x01				Read Application
		Read attribute response	0x11	0x01	0x00 0x00	0x80 0x01	0x00	0x42	0xss "C … C" C … C: Up to 16 bytes Default: "APPNAME" *➔ Source code defined (Should be close from the prefix of firmware resulting file)*	Application name 0xss: Size of string

3.4.2 CONFIGURATION CLUSTER

Clusters	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Configuration (1/3)	WIPAddress	~~Read attribute request~~	~~0x11~~	~~0x00~~	~~0x00 0x50~~	~~0x00 0x00~~				~~Ask the current binded WIPAddress.~~
		~~Read attribute response~~	~~0x11~~	~~0x01~~	~~0x00 0x50~~	~~0x00 0x00~~	~~0x00~~	~~0x41~~	~~0x10 0xdd … 0xdd~~	~~0x10 : size of followed datas~~ ~~0xdd … 0xdd : 16 bytes MSB first of IPv6 address~~
		~~Write attribute no response~~	~~0x11~~	~~0x05~~	~~0x00 0x50~~	~~0x00 0x00~~		~~0x41~~	~~0x10 0xdd … 0xdd~~	~~0x10 and 0xdd … 0xdd : 16 bytes MSB first of IPv6 address~~ ~~OR~~ ~~0x00 to indicate that WIPAddress should be copied from source address of the IP frame. No data.~~ Deprecated but available for retro compatibility. Please use "ReportConnexion" herafter
	~~PanID~~	~~Read attribute request~~	~~0x11~~	~~0x00~~	~~0x00 0x50~~	~~0x00 0x01~~				~~Ask the current PanID~~
		~~Read attribute response~~	~~0x11~~	~~0x01~~	~~0x00 0x50~~	~~0x00 0x01~~	~~0x00~~	~~0x21~~	~~0xdd 0xdd~~	~~0xdd 0xdd : PanID of the network~~
		~~Write attribute no response~~	~~0x11~~	~~0x05~~	~~0x00 0x50~~	~~0x00 0x01~~		~~0x21~~	~~0xdd 0xdd~~	~~0xdd 0xdd : PanID of the network at the next restart.~~
	SimpleDesc	~~Read attribute request~~	~~0x11~~	~~0x00~~	~~0x00 0x50~~	~~0x00 0x02~~				~~Ask the clusterID called by the sensor~~ Deprecated but available for retro compatibility. Please use "Desc" herafter
		~~Read attribute response~~	~~0x11~~	~~0x01~~	~~0x00 0x50~~	~~0x00 0x02~~	~~0x00~~	~~0x4C~~	~~0x11 0x11~~ ~~0xmm [0xyy 0xyy …]~~ ~~0xnn [0xzz 0xzz …]~~	~~0xll 0xll : number of bytes~~ ~~0xmm : number of clusterID as input cluster~~ ~~0xyy 0xyy : input clusterID used~~ ~~0xnn : number of clusterID as output cluster~~ ~~0xzz 0xzz : output clusterID used~~ Deprecated but available for retro compatibility. Please use "Desc" herafter
	ReportConnexion	**Read attribute request**	0x11	0x00	0x00 0x50	0x00 0x03				Ask the current IPv6/port (UDP) connexion binded for reports.
		Read attribute response	0x11	0x01	0x00 0x50	0x00 0x03	0x00	0x41	0x14 0xdd … 0xdd 0xdp 0xdp 0xsp 0xsp	0x14 : size of followed datas 0xdd … 0xdd : 16 bytes MSB first of IPv6 address 0xdp 0xdp: Reporting destination port MSB first 0xsp 0xsp: Reporting source port MSB first
		Write attribute no response	0x11	0x05	0x00 0x50	0x00 0x03		0x41	0x14 0xdd … 0xdd 0xdp 0xdp 0xsp 0xsp Or 0x00	0x14 : size of followed datas 0xdd … 0xdd : 16 bytes MSB first of IPv6 address 0xdp 0xdp: Reporting destination port MSB first 0xsp 0xsp: Reporting source port MSB first Or 0x00 to indicate that report Address and port should be copied from source address and port of the UDP frame.
	Desc	**Read attribute request**	0x11	0x00	0x00 0x50	0x00 0x04				Ask the endpoint/clusterID list managed by the sensor
		Read attribute response	0x11	0x01	0x00 0x50	0x00 0x04	0x00	0x4C	0x11 0x11 0xen { 0xpp 0xmm [0xyy 0xyy]... 0xnn [0xzz 0xzz]... } ...	0xll 0xll : number of bytes 0xen : Number of endpoints 0xpp: Endpoint number 0xmm : number of input cluster 0xyy 0xyy : input clusterID used 0xnn : number of output cluster 0xyy 0xyy : output clusterID used

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Configuration (2/3)	Configuration mode	**Read attribute request**	0x11	0x00	0x00 0x50	0x00 0x05				Ask the current configuration mode status.
		Read attribute response	0x11	0x01	0x00 0x50	0x00 0x05	0x00	0x0a	0xmm 0xtt 0xtt	0xmm: Mode : 0: Not conf mode 1: Conf mode RX (Allways awake) 2: Conf mode NS (NS each 3 sec) 3-255: Reserved 0xtt 0xtt : Remaining duration in configuration mode in seconds. 0: Back to "periodic sleeping mode"
		Write attribute no response	0x11	0x05	0x00 0x50	0x00 0x05		0x0a	0xmm 0xtt 0xtt	*Idem Read attribute response (See also §3.5.1.1.3.5)* *If 0xmm != 1 and 0xmm != 2 => 0x00 and 0xtt 0xtt set to 0x0000*
		Configure reporting	0x11	0x06	0x00 0x50	0x00 0x00 0x05 0x10 0xmm 0xmm 0xMM 0xMM 0xcc 0x10 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : 00: No report on Configuration state change 01: Report when configigiuration mode step from '0' to '> 0' or from '>0' to '0'				Configure the reporting of "Configuration mode" 0x00 : direction (first byte) 0x00 0x05 : attributeID
		Configure reporting response	0x11	0x07	0x00 0x50	0x00 0x00 0x00 0x05				0x00 : status 0x00 : direction 0x00 0x05 : attributeID
		Read reporting configuration	0x11	0x08	0x00 0x50	0x00 0x00 0x05				0x00 : direction 0x00 0x05: attributeID
		Read reporting configuration response	0x11	0x09	0x00 0x50	0x00 0x00 0x00 0x05 0x10 0xmm 0xmm 0xMM 0xMM 0xcc				0x00 : status 0x00 : direction 0x00 0x05 attributeID *+ (See Configure reporting from 0x10)*
		Report attributes	0x11	0x0A	0x00 0x50	0x00 0x05 0x0a 0xmm 0xtt 0xtt				0x00 0x05: attributeID 0x21 : attribute type 0xmm: Mode 0xtt 0xtt : Remaining duration in configuration mode in seconds. 0: Back to "periodic sleeping mode" 0xFFFF: Stay infinitely in" Configuration mode"

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Configuration (3/3)	Node power descriptor *(Batch reportable)*	**Read attribute request**	0x11	0x00	0x00 0x50	0x00 0x06				
		Read attribute response	0x11	0x01	0x00 0x50	0x00 0x06	0x00	0x41	0xss 0xmm 0xrr[0xvv 0xvv] … 0xcc	0xss : size of Byte string 0xmm : Current "Power mode" 0xrr : Mask of "Power sources" 0xvv 0xvv : Voltage level(s) in V/1000 for each "Power source" defined in 0xrr 0xcc: Current "Power source"
		Configure reporting	0x11	0x06	0x00 0x50	0x00 0x00 0x06 0x41 0xmm 0xmm 0xMM 0xMM 0xss 0xmm 0xrr[0xvv 0xvv] … 0xcc				Configure the reporting of "Node power Descriptor" 0x00 : direction (first byte) 0x00 0x06 : attributeID 0x41 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xss : size of the criteria fields 0xmm : ask a report on Power Mode changing 0xrr : : Mask of changed bit that trig a report List of "Power sources criteria". Where For each selected Power sources criteria: [0xvv 0xvv: Delta on specific Power source level to trig] … 0xcc: ask a report on Power Source changing.
										Batch (See also §3.3.1): 0bssssss**1** 0x00 0x06 [0xii 0xmmmm 0xMMMM <Delta> <Resol> 0xtt]... *For each required field:* 0xmmmm 0xMMMM : Minimum and Maximum recording interval 0bssssss: is the size (in byte) of configuration string after attribute ID (0bssssss1 right shifted) 0xii: is the index of required field (*For each index the <Delta> and <Resol> [type] is also given.)* 0 : Power modes [U8], 1: Current power source [U8], 2..6: Power sources level, (See also § 3.5.1.1.3.6) [U16 mV] 2 -> Constant (mains) power 3 -> Rechargeable battery 4 -> Disposable battery 5 -> Solar harvesting 6 -> TIC harvesting <Delta>: the required delta value <Resol>: the required resolution 0xtt: the tag value (0bxxxxxzzz: xxxx = tag label, zzz=tag size);
		Configure reporting response	0x11	0x07	0x00 0x50	0x00 0x00 0x00 0x06				0x00 : status (first byte) 0x00: direction (next byte) 0x00 0x06 : attributeID (last two bytes)
										Batch (See also configure reporting): 0x00 0x01 0x00 0x06
		Read reporting configuration	0x11	0x08	0x00 0x50	0x00 0x00 0x06				0x00: direction (first byte) 0x00 0x06 : attributeID (last two bytes)
										Batch (See also configure reporting): 0x01 0x00 0x06
		Read reporting configuration response	0x11	0x09	0x00 0x50	0x00 0x00 0x00 0x06 0x41 0xmm 0xmm 0xMM 0xMM 0xss 0xmm 0xrr[0xvv 0xvv] … 0xcc				0x00 : status (first byte) 0x00 : direction (next two bytes) 0x00 0x06 : attributeID (next two bytes) *+ (See Configure reporting from 0x41)*
										Batch (See also configure reporting): 0x00 0bssssss**1** 0x00 0x06 [0xii 0xmmmm 0xMMMM <Delta> <Resol> 0xtt] ...
		Report attributes	0x11 +en	0x0A	0x00 0x50	0x00 0x06 0x41 0xss 0xmm 0xrr[0xvv 0xvv] …0xcc				0x00 0x06: attributeID 0x41: Byte string 0xss : size of Byte string 0xmm : Current "Power mode" 0xrr : available "Power sources" 0xvv 0xvv : Voltage level(s) in V/100 for each "Power source" defined in 0xrr 0xcc: Current "Power sources"
			0b…. …**0**		**Batch**: Please refer to "§3.3.2 Batch reporting frame format"					
	Cluster Commands	Command : **REBOOT**	0x11	0x50	0x00 0x50	0x00	Restart device as after a "Power On reset". All configuration parameters are kept.			
		Command : **REMOVE REPORTS** ***Since v1.6.0.1***	0x11	0x50	0x00 0x50	0x02 0xpp	0xpp is: 0b …. ..bs s=1 : remove standards reports, b=1 : remove batchs reports. *This command is used to clear currently running reports.* *No response if command is OK*			
		Command : **Get specific Batch** ***Since v1.6.0.1***	0x11	0x50	0x00 0x50	0x03 *0xcc*	0xcc is: 0b …. .nnn Get a specific batch from its counter. 0xcc: the batch counter to report. Only the first 3 bits are available. Return 0x88 if not specific batch is not available. *This command is usually applied to require previous buffer(s) when it(they) has(have) been missed.*			

3.4.3 ON/OFF CLUSTER

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
ON/OFF	OnOff	Read attribute request	0x11 +en	0x00	0x00 0x06	0x00 0x00				Ask the relay state of the remote SmartPlug
		Read attribute response	0x11 +en	0x01	0x00 0x06	0x00 0x00	0x00	0x10	0x0y	0x0y : relay state (0x01=ON / 0x00=OFF)
		Configure reporting	0x11 +en	0x06	0x00 0x06	0x00 0x00 0x00 0x10 0xmm 0xmm 0xMM 0xMM 0xcc				Configure the reporting of the relay state. 0x00 : direction (first byte) 0x00 0x00 : attributeID (next two bytes) 0x10 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued.
		Configure reporting response	0x11 +en	0x07	0x00 0x06	0x00 0x00 0x00 0x00				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration	0x11 +en	0x08	0x00 0x06	0x00 0x00 0x00				0x00 : direction (first byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration response	0x11 +en	0x09	0x00 0x06	0x00 0x00 0x00 0x00 0x10 0xmm 0xmm 0xMM 0xMM 0xcc				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (next two bytes) 0x10 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : reportable change => contains the minimum change to the attribute that will result in a report being issued.
		Report attributes	0x11 +en	0x0A	0x00 0x06	0x00 0x00 0x10 0x0y				0x00 0x00 : attributeID 0x10 : attribute type 0x0y : relay state (0x01=ON / 0x00=OFF)
	Cluster Commands	Command : OFF	0x11 +en	0x50	0x00 0x06	0x00				0x00 : switch OFF the relay of the remote smartplug
		Command : ON	0x11 +en	0x50	0x00 0x06	0x01				0x01 : switch ON the relay of the remote smartplug
		Command : TOGGLE	0x11 +en	0x50	0x00 0x06	0x02				0x02 : invert the relay state of the remote smartplug

3.4.4 VOLUME METER CLUSTER

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
VOLUME METER	Volume *(Batch reportable)*	Read attribute request	0x11 +en	0x00	0x80 0x02	0x00 0x00				Asks the volume index of the Senso sensor.
		Read attribute response	0x11 +en	0x01	0x80 0x02	0x00 0x00	0x 00	0x2B	0xvv 0xvv 0xvv 0xvv	0xv 0xv 0xv 0xv : volume index (int32=attribute type 0x2B).
		Write attribute no response	0x11 +en	0x05	0x80 0x02	0x00 0x00		0x2B	0xvv 0xvv 0xvv 0xvv	Set the Volume value (see before)
		Configure reporting	0x11 +en	0x06	0x80 0x02	0x00 0x00 0x00 0x2B 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc 0xcc 0xcc				Configure the reporting of the volume index. 0x00 : direction (first byte) 0x00 0x00 : attributeID (next two bytes) 0x2B : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc 0xcc 0xcc 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued.
						Batch (See also §3.3.1): 0x1D 0x0000 0x00 0xmmmm 0xMMMM 0xDDDDDDDD 0xRRRRRRRR 0xtt 0x1D: Batch (payload byte size from after the attribute ID+batch flag): 0bssss 0bsss**1**. 0x0E (14 bytes)*2+**1**(batch flag) =0x1D. 0x0000: The attribute ID. 0x00: The index of the required field (only one field for this attribute): 0 : Volume Index ; ZCL type=INT32 ; Sample_Type (corresponding batch sample type)= 11 0xmmmm 0xMMMM: The Minimum and Maximum recording intervals 0xDDDDDDDD: Delta (record a change greater or equal than Delta) 0xRRRRRRRR: Resolution (enables to degrade the accuracy for a better compression) 0xtt: The tag (0brxxx 0bxzzz: r=reserved, xxxx = tag label, zzz=tag size).				
		Configure reporting response	0x11 +en	0x07	0x80 0x02	0x00 0x00 0x00 0x00				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration	0x11 +en	0x08	0x80 0x02	0x00 0x00 0x00				0x00 : direction (first byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration response	0x11 +en	0x09	0x80 0x02	0x00 0x00 0x00 0x00 0x2B 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc 0xcc 0xcc				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (next two bytes) 0x2B : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc 0xcc 0xcc 0xcc : reportable change => contains the minimum change to the attribute that will result in a report being issued.
		Report attributes	0x11 +en	0x0A	0x80 0x02	0x00 0x00 0x2B 0xvv 0xvv 0xvv 0xvv				0x00 0x00 : attributeID 0x2B : attribute type 0xv 0xv 0xv 0xv : Volume index
	VolumeDisplayMode	Read attribute request	0x11 +en	0x00	0x80 0x02	0x00 0x01				Asks the volume unit of the Volume attribute of the Senso sensor.
		Read attribute response	0x11 +en	0x01	0x80 0x02	0x00 0x01	0x00	0x20	0xvv	0xvv : unit of the Volume attribute (uint8=attribute type 0x20). **VolumeDisplayMode Value** / **Description** 0x00 – deciLiter 0x01 – Liter reserved – …
	MinFlow *(Batch reportable)*	Read attribute request	0x11 +en	0x00	0x80 0x02	0x00 0x02				Asks the minimum flow of the Senso sensor.
		Read attribute response	0x11 +en	0x01	0x80 0x02	0x00 0x02	0x00	0x28	0xvv	0xv : minimum flow (int8=attribute type 0x28).
		Configure reporting	0x11 +en	0x06	0x80 0x02	0x00 0x00 0x02 0x28 0xmm 0xmm 0xMM 0xMM 0xcc				Configure the reporting of the Minflow index. 0x00 : direction (first byte) 0x00 0x02 : attributeID (next two bytes)

							0x28 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued.
						Batch (See also §3.3.1): 0x11 0x0002 0x00 0xmmmm 0xMMMM 0xDD 0xRR 0xtt 0x11 : Batch (payload byte size from after the attribute ID+batch flag): 0bssss 0bsss**1**. 0x08 (8 bytes)*2+**1**(batch flag) =0x11. 0x0002 : The attribute ID. 0x00 : The index of the required field (only one field for this attribute): 0 : Minflow ; ZCL type=INT8 ; Sample_Type (corresponding batch sample type)= 5 0xmmmm 0xMMMM : The Minimum and Maximum recording intervals 0xDD : Delta (record a change greater or equal than Delta) 0xRR : Resolution (enables to degrade the accuracy for a better compression) 0xtt : The tag (0brxxx 0bxzzz: r=reserved, xxxx = tag label, zzz=tag size).	
		Configure reporting response	0x11 +en	0x07	0x80 0x02	0x00 0x00 0x00 0x02	0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x02 : attributeID (last two bytes)
		Read reporting configuration	0x11 +en	0x08	0x80 0x02	0x00 0x00 0x02	0x00 : direction (first byte) 0x00 0x02 : attributeID (last two bytes)
		Read reporting configuration response	0x11 +en	0x09	0x80 0x02	0x00 0x00 0x00 0x02 0x80 0xmm 0xmm 0xMM 0xMM 0xcc	0x00 : status (first byte) 0x00 : direction (first byte) 0x00 0x02 : attributeID (next two bytes) 0x28 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued.
		Report attributes	0x11 +en	0x0A	0x80 0x02	0x00 0x02 0x28 0xvv	0x00 0x02 : attributeID 0x28 : attribute type 0xvv : Minflow index
	MaxFlow *(Batch reportable)*	Read attribute request	0x11 +en	0x00	0x80 0x02	0x00 0x03	Asks the maximum flow of the Senso sensor.
		Read attribute response	0x11 +en	0x01	0x80 0x02	0x00 0x03 0x00 0x28 0xvv	0xvv : maximum flow (int8=attribute type 0x28).
		Configure reporting	0x11 +en	0x06	0x80 0x02	0x00 0x00 0x03 0x28 0xmm 0xmm 0xMM 0xMM 0xcc	Configure the reporting of the Maxflow index. 0x00 : direction (first byte) 0x00 0x03 : attributeID (next two bytes) 0x28 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued.
		Configure reporting response	0x11 +en	0x07	0x80 0x02	0x00 0x00 0x00 0x03	0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x03 : attributeID (last two bytes)
		Read reporting configuration	0x11 +en	0x08	0x80 0x02	0x00 0x00 0x03	0x00 : direction (first byte) 0x00 0x03 : attributeID (last two bytes)
		Read reporting configuration response	0x11 +en	0x09	0x80 0x02	0x00 0x00 0x00 0x03 0x28 0xmm 0xmm 0xMM 0xMM 0xcc	0x00 : status (first byte) 0x00 : direction (first byte) 0x00 0x03 : attributeID (next two bytes) 0x28 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued.
						Batch (See also §3.3.1): 0x11 0x0003 0x00 0xmmmm 0xMMMM 0xDD 0xRR 0xtt 0x11 : Batch (payload byte size from after the attribute ID+batch flag): 0bssss 0bsss**1**. 0x08 (8 bytes)*2+**1**(batch flag) =0x11. 0x0003 : The attribute ID.	

						0x00 : The index of the required field (only one field for this attribute): 0 : Maxflow ; ZCL type=INT8 ; Sample_Type (corresponding batch sample type)= 5 0xmmmm 0xMMMM : The Minimum and Maximum recording intervals 0xDD : Delta (record a change greater or equal than Delta) 0xRR : Resolution (enables to degrade the accuracy for a better compression) 0xtt : The tag (0brxxx 0bxzzz: r=reserved, xxxx = tag label, zzz=tag size).			
		Report attributes	0x11 +en	0x0A	0x80 0x02	0x00 0x03 0x20 0xvv				0x00 0x03 : attributeID 0x28 : attribute type 0xvv : Maxflow index
	FlowDisplayMode	Read attribute request	0x11 +en	0x00	0x80 0x02	0x00 0x04				Asks the flow unit of the MinFlow and MaxFlow attributes of the Senso sensor.
		Read attribute response	0x11 +en	0x01	0x80 0x02	0x00 0x04	0x00	0x20	0xvv	0xvv : unit of the MinFlow and MaxFlow attributes (uint8=attribute type 0x20). **FlowDisplayMode Value** / **Description** 0x00 – Liter/min 0x01 – Liter/sec 0x02 – Liter/hour reserved – …
	Cluster Commands	Command : reset	0x11 +en	0x50	0x80 0x02	0x00 0xrr				0x00 : command ID. Resets the attribute(s) whose the corresponding bit of the flag is set to 1. 0xrr : Flag Reset : **Bit number** / **Description** 0 – Reset Volume 1 – Reset MinFlow 2 – Reset MaxFlow

3.4.5 SENSO CLUSTER

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
SENSO	Status *(Batch reportable)*	Read attribute request	0x11 +en	0x00	0x80 0x03	0x00 0x00				Asks the status of the Senso sensor.
		Read attribute response	0x11 +en	0x01	0x80 0x03	0x00 0x00	0x 00	0x18	0xvv	0xvv : status (Bitmap8=attribute type 0x18) : Bit number / Description 0 : Début/Fin fuite 1 : Début/Fin retour d'eau niveau 1 2 : Début/Fin retour d'eau niveau 2 3 : Début/Fin retour d'eau niveau 3 4 : Début/Fin d'arrachement 5 : Batterie NOK/OK 6 : Installation NOK/OK 7 : Reserved
		Configure reporting	0x11 +en	0x06	0x80 0x03	0x00 0x00 0x00 0x18 0xmm 0xmm 0xMM 0xMM 0xcc				Configure the reporting of the volume index. 0x00 : direction (first byte) 0x00 0x00 : attributeID (next two bytes) 0x18 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued. Here, the value is a status, not a digit (attribute type=0x18), so the reportable change sould be either 0 or 1.
						Batch (See also §3.3.1): 0x11 0x0000 0x00 0xmmmm 0xMMMM 0xDD 0xRR 0xtt 0x11 : Batch (payload byte size from after the attribute ID+batch flag): 0bssss 0bsss**1**. 0x08 (8 bytes)*2+**1**(batch flag) =0x11. 0x0000 : The attribute ID. 0x00 : The index of the required field (only one field for this attribute): 0 : Status ; ZCL type=BITMAP8 ; Sample_Type (corresponding batch sample type)= 4 0xmmmm 0xMMMM : The Minimum and Maximum recording intervals 0xDD : Delta (record a change greater or equal than Delta) 0xRR : Resolution (enables to degrade the accuracy for a better compression) 0xtt : The tag (0brxxx 0bxzzz: r=reserved, xxxx = tag label, zzz=tag size).				
		Configure reporting response	0x11 +en	0x07	0x80 0x03	0x00 0x00 0x00 0x00				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration	0x11 +en	0x08	0x80 0x03	0x00 0x00 0x00				0x00 : direction (first byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration response	0x11 +en	0x09	0x80 0x03	0x00 0x00 0x00 0x00 0x18 0xmm 0xmm 0xMM 0xMM 0xcc				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (next two bytes) 0x18 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : reportable change => contains the minimum change to the attribute that will result in a report being issued.
		Report attributes	0x11 +en	0x0A	0x80 0x03	0x00 0x00 0x18 0xvv				0x00 0x00 : attributeID 0x18 : attribute type 0xvv : Status
	CountDownThresholds	Read attribute request	0x11 +en	0x00	0x80 0x03	0x00 0x01				Asks the Countdown Thresholds attribute of the Senso sensor. Les trois compte à rebours servent à décompter les trois retours d'eau éventuels possibles (niveau 1, 2, 3).
		Read attribute response	0x11 +en	0x01	0x80 0x03	0x00 0x01	0x00	0x41	0x06 0xcc 0xcc 0xdd 0xdd 0xee 0xee	Countdown Thresholds (bytes_string=attribute type 0x41 ; each countdown is uint16) : 0x06 : data length 0xcc 0xcc : CountDown1 Threshold (default value=10) 0xdd 0xdd : CountDown2 Threshold (default value=100) 0xee 0xee : CountDown3 Threshold (default

										value=1000)
		Write attribute no response	0x11 +en	0x05	0x80 0x03	0x00 0x01		0x41	0x06 0xcc 0xcc 0xdd 0xdd 0xee 0xee	Set the Countdown Thresholds values (see before)
	InstallationRotation	Read attribute request	0x11 +en	0x00	0x80 0x03	0x00 0x02				Asks the InstallationRotation attribute of the Senso sensor. The InstallationRotation contains the attribute that indicates the number of rotation of the wheel before confirming the installation. Once confirmed installation the device shall set the associated bit of attribut Status to 1.
		Read attribute response	0x11 +en	0x01	0x80 0x03	0x00 0x02	0x00	0x20	0xvv	0xvv : InstallationRotation (uint8=attribute type 0x20).
	VolumeRotation	Read attribute request	0x11 +en	0x00	0x80 0x03	0x00 0x03				Asks the VolumeRotation of the Senso sensor. The VolumeRotation contains the attribute that indicates the volume to use for a rotation of the wheel. The unit of this volume is in deciliters.
		Read attribute response	0x11 +en	0x01	0x80 0x03	0x00 0x03	0x00	0x21	0xvv 0xvv	0xvv 0xvv : VolumeRotation (uint16=attribute type 0x21).
	Cluster Commands									

3.4.6 TRX CLUSTER

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
TRX	Link budget	Read attribute request	0x11 +en	0x00	0x00 0x0b	0x00 0x00				Ask the Link budget
		Read attribute response	0x11 +en	0x01	0x00 0x0b	0x00 0x00	0x00	0x41	0x04 0xrr 0xrr 0xss 0xtt	0xrr 0xrr : last RSSI pkt (int16_t) 0xss : last SNR pkt (int8_t) 0xtt : rssi noise (int8_t)
		Configure reporting	0x11 +en	0x06	0x00 0x0b	0x00 0x00 0x00 0x41 0xmm 0xmm 0xMM 0xMM 0x03 0xcc 0xdd 0xee 0xcc : reportable change RSSI PKT=> shall contain the minimum change to the attribute that will result in a report being issued 0xdd : reportable change SNR PKT 0xee : reportable change noise				Configure the reporting of link budget. 0x00 : direction (first byte) 0x00 0x00 : attributeID (next two bytes) 0x20 : attribute type 0xmm 0xmm :min reporting interval 0xMM 0xMM :max reporting interval [0xFF 0xFF => no report]
		Configure reporting response	0x11 +en	0x07	0x00 0x0b	0x00 0x00 0x00 0x00				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration	0x11 +en	0x08	0x00 0x0b	0x00 0x00 0x00				0x00 : direction (first byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration response	0x11 +en	0x09	0x00 0x0b	0x00 0x00 0x00 0x00 0x41 0xmm 0xmm 0xMM 0xMM 0x03 0xcc 0xdd 0xee 0xcc : reportable change RSSI PKT=> shall contain the minimum change to the attribute that will result in a report being issued 0xdd : reportable change SNR PKT 0xee : reportable change noise				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (next two bytes) 0x20 : attribute type 0xmm 0xmm :min reporting interval 0xMM 0xMM :max reporting interval [0xFF 0xFF => no report]
		Report attributes	0x11 +en	0x0A	0x00 0x0b	0x00 0x00 0x41 0x04 0xrr 0xrr 0xss 0xtt				0x00 0x00 : attributeID 0x41 : attribute type 0xrr 0xrr : last RSSI pkt (int16_t) 0xss : last SNR pkt (int8_t) 0xtt : rssi noise (int8_t)
	channel	Read attribute request	0x11 +en	0x00	0x00 0x0b	0x00 0x01				Ask the channel used
		Read attribute response	0x11 +en	0x01	0x00 0x0b	0x00 0x01	0x00	0x20	0xx	0xxx : read channel
		Write attribute no response	0x11 +en	0x05	0x00 0x0b	0x00 0x01		0x20	0xx	0xxx : *Update the channel*
	SF	Read attribute request	0x11 +en	0x00	0x00 0x0b	0x00 0x02				Ask the SF used
		Read attribute response	0x11 +en	0x01	0x00 0x0b	0x00 0x02	0x00	0x20	0xx	0xxx : read SF
		Write attribute no response	0x11 +en	0x05	0x00 0x0b	0x00 0x02		0x20	0xx	0xxx : *Update the SF*
	BW	Read attribute request	0x11 +en	0x00	0x00 0x0b	0x00 0x03				Ask the BW used
		Read attribute response	0x11 +en	0x01	0x00 0x0b	0x00 0x03	0x00	0x20	0xx	0xxx : read BW
		Write attribute no response	0x11 +en	0x05	0x00 0x0b	0x00 0x03		0x20	0xx	0xxx : *Update the BW*
	TXPOW	Read attribute request	0x11 +en	0x00	0x00 0x0b	0x00 0x04				Ask the TXPOW used
		Read attribute response	0x11 +en	0x01	0x00 0x0b	0x00 0x04	0x00	0x28	0xx	0xxx : read TXPOW (int8)
		Write attribute no response	0x11 +en	0x05	0x00 0x0b	0x00 0x04		0x28	0xx	0xxx : *Update the* TXPOW
	cmds	Command : read TRX config	0x11 +en	0x50	0x00 0x0b	0x01				
		Command : read TRX config response	0x11 +en	0x01	0x00 0x0b	0x00 0x05	0x00	0x41	0x04 0xll 0xmm 0xnn 0xoo	0xll : read channel (uint8) 0xmm : read SF (uint8) 0xnn : read BW (uint8) 0xoo : read txPow (int8)

3.4.7 SIMPLE-METERING LIKE CLUSTER

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Simple Metering-Like	Current Metering *(Batch reportable)*	Read attribute	0x11 +en	0x00	0x00 0x52	0x00 0x00				Ask the consumption of the remote SmartPlug.
		Read attribute response	0x11 +en	0x01	0x00 0x52	0x00 0x00	0x00	0x41	0x0c 0xpp 0xpp 0xpp 0xqq 0xqq 0xqq 0xnn 0xnn 0xww 0xww 0xrr 0xrr 0xww 0xww : active power in W (Signed 16 bit short) 0xrr 0xrr : reactive power in VAR (Signed 16 bit short)	0x0c : data length 0xpp 0xpp 0xpp : summation of the active energy in W.h (Signed 24 bit short) 0xqq 0xqq 0xqq : summation of the reactive energy in VAR.h (Signed 24 bit short) 0xnn 0xnn : number of sample*
		Configure reporting	0x11 +en	0x06	0x00 0x52	0x00 0x00 0x00 0x41 0xmm 0xmm 0xMM 0xMM 0x0c 0xpp 0xpp 0xpp 0xqq 0xqq 0xqq 0xnn 0xnn 0xww 0xww 0xrr 0xrr				Configure the reporting of the current summation.
						0x00 : direction (first byte) 0x00 0x00 : attributeID (next two bytes) 0x41 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] Reportable change : shall contain the minimum change to the attribute that will result in a report being issued. Data has to be filled as follow : 0x0c : data length 0xpp 0xpp 0xpp : variation of summation of the active energy in W.h (24 bits signed short (>0)) 0xqq 0xqq 0xqq : variation of summation of the reactive energy in VAR.h (24 bits signed short (>0)) 0xnn 0xnn : variation of number of sample* 0xww 0xww : variation of active power in W (Signed 16 bit signed short (>0)) 0xrr 0xrr : variation of reactive power in VAR (Signed 16 bit signed short (>0))				
						Batch (See also §3.3.1): 0bssssssss**1** 0x00 0x00 [0xii 0xmmmm 0xMMMM <Delta> <Resol> 0xtt]... *For each required field:* 0xmmmm 0xMMMM : Minimum and Maximum recording interval 0bssssssss: is the size (in byte) of configuration string after attribute ID (0bssssssss1 right shifted) 0xii: is the index of required field (*For each index the <Delta> and <Resol> [type] is also given.)* 0 : Active energy [I24, Wh], 1: Reactive energy [I24, VArh], 2: Accumulation duration [U16, s], 3 : Active power [I16, W], 4: Reactive power [I16, VAr], <Delta>: the required delta value <Resol>: the required resolution 0xtt: the tag value (0bxxxxxzzz: xxxx = tag label, zzz=tag size);				
		Configure reporting response	0x11 +en	0x07	0x00 0x52	0x00 0x00 0x00 0x00				0x00 : status (first byte) 0x00: direction (next byte) 0x00 0x00 : attributeID (last two bytes)
						Batch (See also configure reporting): 0x00 0x01 0x00 0x00				
		Read reporting configuration	0x11 +en	0x08	0x00 0x52	0x00 0x00 0x00				0x00: direction (first byte) 0x00 0x00 : attributeID (last two bytes)
						Batch (See also configure reporting): 0x01 0x00 0x00				
		Read reporting configuration response	0x11 +en	0x09	0x00 0x52	0x00 0x00 0x00 0x00 0x41 0xmm 0xmm 0xMM 0xMM 0x0c 0xpp 0xpp 0xpp 0xqq 0xqq 0xqq 0xnn 0xnn 0xww 0xww 0xrr 0xrr				0x00 : status (first byte) 0x00: direction (next byte) 0x00 0x00 : attributeID (next two bytes) *+ (See Configure reporting from 0x41)*
						Batch (See also configure reporting): 0x00 0bssssssss**1** 0x00 0x06 [0xii 0xmmmm 0xMMMM <Delta> <Resol> 0xtt] ...				
		Report attributes	0x11 +en	0x0A	0x00 0x52	0x00 0x00 0x41 0x0c 0xpp 0xpp 0xpp 0x00 0x00 0x00 0xnn 0xnn 0xww 0xww 0x00 0x00				0x00 0x00 : attributeID 0x41 : attribute type 0x0c : data length
						0xpp 0xpp 0xpp : summation of the active energy in W.h (Signed 24 bit short) 0xqq 0xqq 0xqq : summation of the reactive energy in VAR.h (Signed 24 bit short) 0xnn 0xnn : number of sample* 0xww 0xww : active power in W (Signed 16 bit short) 0xrr 0xrr : reactive power in VAR (Signed 16 bit short)				
			0b.... ...**0**		**Batch** : Please refer to "3.3.2 Batch reporting frame format"					
	Current Calibration	Read attribute	0x11 +en	0x00	0x00 0x52	0x80 0x00				Ask the current calibration coefficient.

		Read attribute response	0x11 +en	0x01	0x00 0x52	0x80 0x00	0x00	0x41	0x09 0xee 0xnn 0xnn 0xmm 0xmm 0xpp 0xpp 0xqq 0xqq	0x08 : data length 0xee: Current e2Pot value (unsigned char) (*programmable input resistor for range tunning*) *Following are signed 16 bit short:* 0xnn 0xnn : Active power multiplier 0xmm 0xmm : Active power divisor 0xpp 0xpp : Reactive power multiplier 0xqq 0xqq :ReaActive power divisor
		Write attribute no response	0x11 +en	0x05	0x00 0x52	0x80 0x00		0x41	0x09 0xee 0xnn 0xnn 0xmm 0xmm 0xpp 0xpp 0xqq 0xqq	Set the "Calibration parameters values (see before) NOTA: Setting a power divisor to 0 invalidate the corresponding (active or reactive) measurement.
	Cluster Commands	Reset command	0x11 +en	0x50	0x00 0x52	0x00				Reset the consumption of the remote SmartPlug
		Calibrate command	0x11 +en	0x50	0x00 0x52	0x80 0x06 0xcc 0xcc 0xnn 0xnn 0xrr 0xrr *Following are signed 16 bit short:* 0xcc 0xcc => Max current of connected tore or clamp. 0 if not a measurement with external Tore or Clamp. 0xnn 0xnn => Active power of connected load 0xrr 0xrr => Reactive power of connected load NOTA: setting Active or Reactive power to 0 avoid the calibration of the corresponding measure (Active or Reactive)				Request calibration using current connected load :

* Getting the average power value (in W or VAR) during 'nn' samples is obtained by 'pp / nn'

3.4.8 POWER QUALITY CLUSTER

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Power Quality	Current *(Batch reportable)*	Read attribute	0x11 +en	0x00	0x80 0x52	0x00 0x00				Ask current Power quality measurements.
		Read attribute response	0x11 +en	0x01	0x80 0x52	0x00 0x00	0x00	0x41	0x18[0xpppp]*12 [0xpppp] x 12 : 12 unsigned 16 bit short respectively: Freq, Freq Min, Freq Max : (x + 22232)(Hz /1000) Vrms, Vrms Min, Vrms Max : x V/10 Vpeak, Vpeak Min, Vpeak Max :x V/10 OverVoltageNumber, SagNumber, BrownoutNumber : Number *All values except BrownoutNumber are reset on each bootstrap.*	0x18 : data length
		Configure reporting	0x11 +en	0x06	0x80 0x52	0x00 0x00 0x00 0x41 0xmm 0xmm 0xMM 0xMM 0x18[0xpppp]*12 0x00 : direction (first byte) 0x00 0x00 : attributeID (next two bytes) 0x41 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] Reportable change : shall contain the minimum change to the attribute that will result in a report being issued. Data has to be filled as follow : 0x18 : data length [0xpppp] x 12 : 12 unsigned 16 bit short respectively: dFreq, dFreq Min, dFreq Max : (Hz /1000) dVrms, dVrms Min, dVrms Max : x V/10 dVpeak, dVpeak Min,dVpeak Max :x V/10 dOverVoltageNumber, dSagNumber, dBrownoutNumber : Number *All values except BrownoutNumber are reset on each bootstrap.*		Configure the reporting of the current Power Quality.
						Batch : (See also §3.3.1): 0bssssssss**1** 0x00 0x00 [0xii 0xmmmm 0xMMMM <Delta> <Resol> 0xtt]... *For each required field:* 0xmmmm 0xMMMM : Minimum and Maximum recording interval 0bssssssss: is the size (in byte) of configuration string after attribute ID (0bsssssss1 right shifted) 0xii: is the index of required field (*For each index the <Delta> and <Resol> [type] is also given.)* 0,1,2 : Freq, Freq Min, Freq Max : [U16 (Hz /1000)], 3,4,5 : Vrms, Vrms Min, Vrms Max: [U16 (V/10)], 6,7,8 : Vpeak, Vpeak Min, Vpeak Max: [U16 (V/10)], 9,10,11: OverVoltageNumber, SagNumber, BrownoutNumber [U16, Number] <Delta>: the required delta value <Resol>: the required resolution 0xtt: the tag value (0bxxxxxzzz: xxxx = tag label, zzz=tag size);		
		Configure reporting response	0x11 +en	0x07	0x80 0x52	0x00 0x00 0x00 0x00				0x00 : status (first byte) 0x00: direction (next byte) 0x00 0x00 : attributeID (last two bytes)
						Batch : (See also configure reporting): 0x00 0x01 0x00 0x00				
		Read reporting configuration	0x11 +en	0x08	0x80 0x52	0x00 0x00 0x00				0x00: direction (first byte) 0x00 0x00 : attributeID (last two bytes)
						Batch : (See also configure reporting): 0x01 0x00 0x00				
		Read reporting configuration response	0x11 +en	0x09	0x80 0x52	0x00 0x00 0x00 0x41 0xmm 0xmm 0xMM 0xMM 0x18[0xpppp]*12				0x00 : status (first byte) 0x00: direction (next byte) 0x00 0x00 : attributeID (next two bytes) *+ (See Configure reporting from 0x41)*
						Batch : (See also configure reporting): 0x00 0bssssssss**1** 0x00 0x06 [0xii 0xmmmm 0xMMMM <Delta> <Resol> 0xtt] ...				
		Report attributes	0x11 +en	0x0A	0x80 0x52	0x00 0x00 0x00 0x41 0x18[0xpppp]*12				0x00 0x00 : attributeID
						0x41 : attribute type 0x18 : data length [0xpppp] x 12 : 12 unsigned 16 bit short respectively: Freq, Freq Min, Freq Max : (x + 22232)(Hz /1000) Vrms, Vrms Min, Vrms Max : x V/10 Vpeak, Vpeak Min, Vpeak Max :x V/10 OverVoltageNumber, SagNumber, BrownoutNumber : Number *All values except BrownoutNumber are reset on each bootstrap.*				
			0b.... ...**0**		0x80 0x52	**Batch**: Please refer to "3.3.2 Batch reporting frame format"				
	Sag cycle Threshold	Read attribute	0x11 +en	0x00	0x80 0x52	0x00 0x01				Ask the current over Sag Cycle threshold *Number of voltage half cycle under Sag Voltage Treshold to increment SagNumber.* *Default value: 2 (1 cycle => 20 ms at 50 Hz)*

		Read attribute response	0x11 +en	0x01	0x80 0x52	0x00 0x01	0x00	0x20	0xpp	0xpppp : Sag half cycle threshold in number (0..255)
		Write attribute no response	0x11 +en	0x05	0x80 0x52	0x00 0x01		0x20	0xpp	0xpp : Sag half cycle threshold in (V/10)
	Sag voltage Treshold	Read attribute	0x11 +en	0x00	0x80 0x52	0x00 0x02				Ask the current Sag voltage threshold *Voltage Peak Treshold under which a Sag is consider to increment SagNumber.* *Default value: 260V peak*
		Read attribute response	0x11 +en	0x01	0x80 0x52	0x00 0x02	0x00	0x21	0xpppp	0xpppp : Sag voltage threshold in (V/10)
		Write attribute no response	0x11 +en	0x05	0x80 0x52	0x00 0x02		0x21	0xpppp	0xpppp : Sag over voltage threshold in (V/10)
	Over voltage Threshold	Read attribute	0x11 +en	0x00	0x80 0x52	0x00 0x03				Ask the current over voltage threshold *Voltage peak Treshold over which a OvrVoltageNumber is incremented.* *Default value: 390V peak*
		Read attribute response	0x11 +en	0x01	0x80 0x52	0x00 0x03	0x00	0x21	0xpppp	0xpppp : peak over voltage threshold in (V/10)
		Write attribute no response	0x11 +en	0x05	0x80 0x52	0x00 0x03		0x21	0xpppp	0xpppp : peak over voltage threshold in (V/10)
	Cluster Commands	Reset Power Quality fields	0x11 +en	0x50	0x80 0x52	0x00 0xrr				Request reset of accumulated fields :
						0xrr: Bitfield for which b0: Reset Freq Min&Max *(0xFFFF and 0x0000)* b1: Reset VRMS Min&Max *(0xFFFF and 0x0000)* b2: Reset VPeak Min&Max *(0xFFFF and 0x0000)* b3: Reset SAGNumber*(0x0000)* b4: Reset OverVoltageNumber *(0x0000)* b4: Reset OverBrownoutNumber *(0x0000)*			

3.4.9 TEMPERATURE MEASUREMENT CLUSTER

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Temperature Measurement	MeasuredValue*(Batch reportable)*	Read attribute request	0x11 +en	0x00	0x04 0x02	0x00 0x00				Ask the temperature of the remote sensor
		Read attribute response	0x11 +en	0x01	0x04 0x02	0x00 0x00	0x00	0x29	0xtt 0xtt	0xtt 0xtt : MeasuredValue Temperature (°C) = (MeasuredValue/100)
		Configure reporting	0x11 +en	0x06	0x04 0x02	0x00 0x00 0x00 0x29 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc				Configure the reporting of the MeasuredValue.
						0x00 : direction (first byte) 0x00 0x00 : attributeID (next two bytes) 0x29 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued.				
						Batch (See also §3.3.1): 0bssssssss**1** 0x00 0x00 0x00 0xmmmm 0xMMMM <Delta> <Resol> 0xtt 0xmmmm 0xMMMM : Minimum and Maximum recording interval 0bssssssss**1** : Batch (size+flag). 0bssssssss: is the size (in byte) of configuration string after attribute ID (0bssssssss1 right shifted). Here the size is 0x0A. So the Batch field is 0x0A*2+1=0x15. 0x00: is the index of required field (The MeasuredValue attribute has got only one field*):* 0 : Measured value [I16, °C/100] <Delta>: the required delta value <Resol>: the required resolution 0xtt: the tag value (0bxxxxxzzz: xxxx = tag label, zzz=tag size);				
		Configure reporting response	0x11 +en	0x07	0x04 0x02	0x00 0x00 0x00 0x00				0x00 : status (first byte) 0x00: direction (next byte) 0x00 0x00 : attributeID (last two bytes)
						Batch (See also configure reporting): 0x00 0x01 0x00 0x00				
		Read reporting configuration	0x11 +en	0x08	0x04 0x02	0x00 0x00 0x00				0x00: direction (next byte) 0x00 0x00 : attributeID (last two bytes)
						Batch (See also configure reporting): 0x01 0x00 0x00				
		Read reporting configuration response	0x11 +en	0x09	0x04 0x02	0x00 0x00 0x00 0x00 0x29 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc00				0x00 : status (first byte) 0x00 | 0x01: direction (next byte) 0x00 0x00 : attributeID (next two bytes) 0x29 : attribute type*+ (See Configure reporting from 0x29)*
						Batch (See also batch configure reporting): 0x00 0bssssssss**1** 0x00 0x00 0x00 0xmmmm 0xMMMM <Delta> <Resol> 0xtt				
		Report attributes	0x11 +en	0x0A	0x04 0x02	0x00 0x00 0x29 0xtt 0xtt				0x00 0x00 : attributeID 0x29 : attribute type
						0xtt 0xtt : MeasuredValue Temperature (°C) = (MeasuredValue/100)				
			0b.... ...**0**		**Batch** : Please refer to "3.3.2 Batch reporting frame format"					
	MinMeasuredValue	Read attribute request	0x11 +en	0x00	0x04 0x02	0x00 0x01				Ask the minimum value that the remote sensor is capable to measure
		Read attribute response	0x11 +en	0x01	0x04 0x02	0x00 0x01	0x00	0x29	0xtt 0xtt	0xtt 0xtt : MinMeasuredValue Temperature (°C) = (MinMeasuredValue/100)
	MaxMeasuredValue	Read attribute request	0x11 +en	0x00	0x04 0x02	0x00 0x02				Ask the maximum value that the remote sensor is capable to measure
		Read attribute response	0x11 +en	0x01	0x04 0x02	0x00 0x02	0x00	0x29	0xtt 0xtt	0xtt 0xtt : MaxMeasuredValue Temperature (°C) = (MaxMeasuredValue/100)

3.4.10 RELATIVE HUMIDITY MEASUREMENTS CLUSTER

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Relative Humidity Measurement	MeasuredValue *(Batch reportable)*	Read attribute request	0x11 +en	0x00	0x04 0x05	0x00 0x00				Ask the relative humidity of the remote sensor
		Read attribute response	0x11 +en	0x01	0x04 0x05	0x00 0x00	0x00	0x21	0xhh 0xhh	0xhh 0xhh : MeasuredValue Relative Humidity (%) = (MeasuredValue/100)
		Configure reporting	0x11 +en	0x06	0x04 0x05	0x00 0x00 0x00 0x21 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc				Configure the reporting of the MeasuredValue.
						0x00 : direction (first byte) 0x00 0x00 : attributeID (next two bytes) 0x21 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued.				
						Batch (See also §3.3.1): 0bssssss**1** 0x00 0x00 0x00 0xmmmm 0xMMMM <Delta> <Resol> 0xtt 0xmmmm 0xMMMM : Minimum and Maximum recording interval 0bssssss**1** : Batch (size+flag). 0bssssss: is the size (in byte) of configuration string after attribute ID (0bssssss1 right shifted). Here the size is 0x0A. So the Batch field is 0x0A*2+1=0x15. 0x00: is the index of required field (The MeasuredValue attribute has got only one field*):* 0 : Measured value [U16, %/100] <Delta>: the required delta value <Resol>: the required resolution 0xtt: the tag value (0bxxxxxzzz: xxxx = tag label, zzz=tag size);				
		Configure reporting response	0x11 +en	0x07	0x04 0x05	0x00 0x00 0x00 0x00				0x00 : status (first byte) 0x00: direction (next byte) 0x00 0x00 : attributeID (last two bytes)
						Batch (See also configure reporting): 0x00 0x01 0x00 0x00				
		Read reporting configuration	0x11 +en	0x08	0x04 0x05	0x00 0x00 0x00				0x00 : direction (first byte) 0x00 0x00 : attributeID (last two bytes)
						Batch (See also configure reporting): 0x01 0x00 0x00				
		Read reporting configuration response	0x11 +en	0x09	0x04 0x05	0x00 0x00 0x00 0x00 0x21 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (next two bytes) 0x21 : attribute type *+ (See Configure reporting from 0x21)*
						Batch (See also batch configure reporting): 0x00 0bssssss**1** 0x00 0x00 0x00 0xmmmm 0xMMMM <Delta> <Resol> 0xtt				
		Report attributes	0x11 +en	0x0A	0x04 0x05	0x00 0x00 0x21 0xhh 0xhh				0x00 0x00 : attributeID 0x21 : attribute type
						0xtt 0xtt : MeasuredValue RH (%) = (MeasuredValue/100)				
			0b.... ...**0**		**Batch** : Please refer to "3.3.2 Batch reporting frame format"					
	MinMeasuredValue	Read attribute request	0x11 +en	0x00	0x04 0x05	0x00 0x01				Ask the minimum value that the remote sensor is capable to measure
		Read attribute response	0x11 +en	0x01	0x04 0x05	0x00 0x01	0x00	0x21	0xhh 0xhh	0xhh 0xhh : MinMeasuredValue RH (%) = (MinMeasuredValue/100)
	MaxMeasuredValue	Read attribute request	0x11 +en	0x00	0x04 0x05	0x00 0x02				Ask the maximum value that the remote sensor is capable to measure
		Read attribute response	0x11 +en	0x01	0x04 0x05	0x00 0x02	0x00	0x21	0xhh 0xhh	0xhh 0xhh : MaxMeasuredValue RH (%) = (MaxMeasuredValue/100)

3.4.11 OCCUPANCY SENSING CLUSTER

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Occupancy sensing	Occupancy	Read attribute request	0x11 +en	0x00	0x04 0x06	0x00 0x00				Ask the status of the occupancy sensor.
		Read attribute response	0x11 +en	0x01	0x04 0x06	0x00 0x00	0x00	0x18	0x0y	0x0y : occupancy state (0x01=occupied / 0x00=unoccupied)
		Configure reporting	0x11 +en	0x06	0x04 0x06	0x00 0x00 0x00 0x18 0xmm 0xmm 0xMM 0xMM 0xcc				Configure the reporting of the Occupancy. 0x00 : direction (first byte)
						0x00 0x00 : attributeID (next two bytes) 0x18 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued.				
		Configure reporting response	0x11 +en	0x07	0x04 0x06	0x00 0x00 0x00 0x00				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration	0x11 +en	0x08	0x04 0x06	0x00 0x00 0x00				0x00 : direction (first byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration response	0x11 +en	0x09	0x04 0x06	0x00 0x00 0x00 0x00 0x18 0xmm 0xmm 0xMM 0xMM 0xcc				0x00 : status (first byte) 0x00 : direction (next byte)
						0x00 0x00 : attributeID (next two bytes) 0x18 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : reportable change => contains the minimum change to the attribute that will result in a report being issued.				
		Report attributes	0x11 +en	0x0A	0x04 0x06	0x00 0x00 0x18 0x0y				0x00 0x00 : attributeID 0x18 : attribute type 0x0y : occupancy state (0x01=occupied / 0x00=unoccupied)
	Occupancy Type	Read attribute request	0x11 +en	0x00	0x04 0x06	0x00 0x01				Ask the type of the remote occupancy sensor.
		Read attribute response	0x11 +en	0x01	0x04 0x06	0x00 0x01	0x00	0x30	0x00	0x00 : means PIR sensor
	PIROccupiedToUnoccupiedDela y	Read attribute request	0x11 +en	0x00	0x04 0x06	0x00 0x10				Ask the delay from occupied to unoccupied state of the remote occupancy sensor.
		Read attribute response	0x11 +en	0x01	0x04 0x06	0x00 0x10	0x00	0x20	0xyy	0xyy : delay from occupied to unoccupied state (in second).
		Write attribute no response	0x11 +en	0x05	0x04 0x06	0x00 0x10		0x20	0xyy	Update the delay from occupied to unoccupied state of the remote occupancy sensor.
	PIRUnoccupiedToOccupiedDelay	Read attribute request	0x11 +en	0x00	0x04 0x06	0x00 0x11				Ask the delay from unoccupied to occupied state of the remote occupancy sensor.
		Read attribute response	0x11 +en	0x01	0x04 0x06	0x00 0x11	0x00	0x20	0xyy	0xyy : delay from unoccupied to occupied state (in second).
		Write attribute no response	0x11 +en	0x05	0x04 0x06	0x00 0x11		0x20	0xyy	Update the delay from unoccupied to occupied state of the remote occupancy sensor.

3.4.12 ANALOG INPUT CLUSTER

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Analog Input (Basic)	PresentValue*(Batch reportable)*	Read attribute request	0x11+en	0x00	0x00 0x0C	0x00 0x55				Ask the Analog Input value of the sensor.
		Read attribute response	0x11+en	0x01	0x00 0x0C	0x00 0x55	0x00	0x39	0xss 0xss 0xss 0xss	0xss 0xss 0xss 0xss : result is a single precision value, in the ApplicationType attribute unit.
		Configure reporting	0x11+en	0x06	0x00 0x0C	0x00 0x00 0x55 0x39 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc 0xcc 0xcc				Configure the reporting of the Analog Input. 0x00 : direction (first byte)
						0x00 0x55 : attributeID (next two bytes) 0x39 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc 0xcc 0xcc 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued.				
						Batch (See also §3.3.1): 0bssssssss**1** 0x00 0x55 0x00 0xmmmm 0xMMMM <Delta> <Resol> 0xtt 0xmmmm 0xMMMM : Minimum and Maximum recording interval 0bssssssss**1** : Batch (size+flag). 0bssssssss: is the size (in byte) of configuration string after attribute ID (0bssssssss1 right shifted). Here the size is 0x0E. So the Batch field is 0x0E*2+1=0x1D. 0x00: is the index of required field (The MeasuredValue attribute has got only one field*):* 0 : Measured value [Single (type depends on usage. See ApplicationType attribute unit)] <Delta>: the required delta value <Resol>: the required resolution 0xtt: the tag value (0bxxxxxzzz: xxxx = tag label, zzz=tag size);				
		Configure reporting response	0x11+en	0x07	0x00 0x0C	0x00 0x00 0x00 0x55				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x55 : attributeID (last two bytes)
						Batch (See also configure reporting): 0x00 0x01 0x00 0x55				
		Read reporting configuration	0x11+en	0x08	0x00 0x0C	0x00 0x00 0x55				0x00 : direction (first byte) 0x00 0x55 : attributeID (last two bytes)
						Batch (See also configure reporting): 0x01 0x00 0x55				
		Read reporting configuration response	0x11+en	0x09	0x00 0x0C	0x00 0x00 0x00 0x55 0x39 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc 0xcc 0xcc				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x55 : attributeID (next two bytes)
						0x39 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc 0xcc 0xcc 0xcc : reportable change => contains the minimum change to the attribute that will result in a report being issued.				
						Batch (See also batch configure reporting): 0x00 0bssssssss**1** 0x00 0x55 0x00 0xmmmm 0xMMMM <Delta> <Resol> 0xtt				
		Report attributes	0x11+en	0x0A	0x00 0x0C	0x00 0x55 0x39 0xss 0xss 0xss 0xss				0x00 0x55 : attributeID 0x39 : attribute type 0xss 0xss 0xss 0xss : result is a single precision value, in the ApplicationType attribute unit.
			0b.... ...**0**		**Batch** : Please refer to "3.3.2 Batch reporting frame format"					
	ApplicationType	Read attribute request	0x11+en	0x00	0x00 0x0C	0x01 0x00				Ask the ApplicationType of the remote sensor.
		Read attribute response	0x11+en	0x01	0x00 0x0C	0x01 0x00	0x00	0x23	0x00 0x05 0x00 0x00	0x00 0x05 0x00 0x00 : present value means Return Carbon Dioxide AI application. Unit is in PPM (Parts Per Million).
									0x00 0xFF 0x00 0x00	0x00 0xFF 0x00 0x00 = 0x00 : Group Analog Input (AI) ; 0xFF: Type Others ; 0x0000 : Index milliampers mA (Watteco defined)
									0x00 0xFF 0x00 0x01	0x00 0xFF 0x00 0x01 = 0x00 : Group Analog Input (AI) ; 0xFF: Type Others ; 0x0001 : Index millivolts mV (Watteco defined)

3.4.13 BINARY INPUT CLUSTER

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Binary Input (Basic) (1/2)	PresentValue(*Batch reportable*)	Read attribute request	0x11 +en	0x00	0x00 0x0F	0x00 0x55				Ask the Binary Input value of the sensor.
		Read attribute response	0x11 +en	0x01	0x00 0x0F	0x00 0x55	0x00	0x10	0x0b	0x0b : binary value (0x00 or 0x01)
		Configure reporting	0x11 +en	0x06	0x00 0x0F	0x00 0x00 0x55 0x10 0xmm 0xmm 0xMM 0xMM 0xcc 0x00 : direction (first byte) 0x00 0x55 : attributeID (next two bytes) 0x10 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued. **Batch** (See also §3.3.1): 0bssssssss**1** 0x00 0x55 0x00 0xmmmm 0xMMMM <Delta> <Resol> 0xtt 0xmmmm 0xMMMM : Minimum and Maximum recording interval 0bssssssss**1** : Batch (size+flag). 0bssssssss: is the size (in byte) of configuration string after attribute ID (0bssssssss1 right shifted). Here the size is 0x08. So the Batch field is 0x08*2+1=0x11. 0x00: is the index of required field (The PresentValue attribute has got only one field*):* 0 : Present value [0/1] <Delta>: the required delta value <Resol>: the required resolution 0xtt: the tag value (0bxxxxxzzz: xxxx = tag label, zzz=tag size);				Configure the reporting of the Present Value Input.
		Configure reporting response	0x11 +en	0x07	0x00 0x0F	0x00 0x00 0x00 0x55 **Batch** (See also configure reporting): 0x00 0x01 0x00 0x55				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x55 : attributeID (last two bytes)
		Read reporting configuration	0x11 +en	0x08	0x00 0x0F	0x00 0x00 0x55 **Batch** (See also configure reporting): 0x01 0x00 0x55				0x00 : direction (first byte) 0x00 0x55 : attributeID (last two bytes)
		Read reporting configuration response	0x11 +en	0x09	0x00 0x0F	0x00 0x00 0x00 0x55 0x10 0xmm 0xmm 0xMM 0xMM 0xcc **Batch** (See also batch configure reporting): 0x00 0bssssssss**1** 0x00 0x55 0x00 0xmmmm 0xMMMM <Delta> <Resol> 0xtt				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x55 : attributeID (next two bytes) 0x10 : attribute type *See configure reporting for next bytes*
		Report attributes	0x11 +en	0x0A	0x00 0x0F	0x00 0x55 0x10 0xcc				0x00 0x55 : attributeID 0x10 : attribute type 0xcc : binary value (0x00 or 0x01).
			0b.... ...**0**		**Batch** : Please refer to "3.3.2 Batch reporting frame format"					
	Polarity	Read attribute request	0x11 +en	0x00	0x00 0x0F	0x00 0x54				Ask the current polarity of the remote sensor.
		Read attribute response	0x11 +en	0x01	0x00 0x0F	0x00 0x54	0x00	0x10	0xnn	0xnn: 0x00: Normal 0x01: reversed
		Write attribute no response	0x11 +en	0x05	0x00 0x0F	0x00 0x54		0x10	0xnn	0xnn: 0x00: Normal 0x01: reversed
	ApplicationType	Read attribute request	0x11 +en	0x00	0x00 0x0F	0x01 0x00				Ask the ApplicationType of the remote sensor.
		Read attribute response	0x11 +en	0x01	0x00 0x0F	0x01 0x00	0x00	0x23	0x03 0x01 0x00 0x02	0x03 0xff 0xff 0xff : Default Other example: 0x03 0x01 0x00 0x02 : Motion Closure
		Write attribute no response	0x11 +en	0x05	0x00 0x0F	0x01 0x00		0x23	0x03 0xmm 0xcc 0xcc	0x03: Binary input 0xmm: Type 0xcc 0xcc: Application usage
	Edge selection	Read attribute request	0x11 +en	0x00	0x00 0x0F	0x04 0x00				Ask the current Edge selection
		Read attribute response	0x11 +en	0x01	0x00 0x0F	0x04 0x00	0x00	0x18	0xnn 0x01 : Falling edge 0x02: Rising Edge 0x03: Both Rising and falling 0x04: Polling	0xnn:
		Write attribute no response	0x11 +en	0x05	0x00 0x0F	0x04 0x00		0x18	0xnn 0x01 : Falling edge 0x02: Rising Edge 0x03: Both Rising and falling 0x04: Polling	0xnn:

Cluster	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Binary Input (Basic) (2/2)	Debounce period	Read attribute request	0x11 +en	0x00	0x00 0x0F	0x04 0x01				Ask the current Debounce period
		Read attribute response	0x11 +en	0x01	0x00 0x0F	0x04 0x01	0x00	0x21	0xnn 0xnn	0xnn 0xnn: Debounce period (milliseconds)
		Write attribute no response	0x11 +en	0x05	0x00 0x0F	0x04 0x01		0x21	0xnn 0xnn	0xnn 0xnn: Debounce period (milliseconds)
	Count *(Batch reportable)*	Read attribute request	0x11 +en	0x00	0x00 0x0F	0x04 0x02				
		Read attribute response	0x11 +en	0x01	0x00 0x0F	0x04 0x02	0x00	0x23	0xcc 0xcc 0xcc 0xcc	Current value of the counter (Big Endian)
		Configure reporting	0x11 +en	0x06	0x00 0x0F	0x00 0x04 0x02 0x23 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc 0xcc 0xcc				Configure the reporting of the Counter
						0x00 : direction (first byte) 0x04 0x02 : attributeID (next two bytes) 0x23 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc 0xcc 0xcc 0xcc : reportable change => the minimum change to the attribute that will result in a report being issued.				
						Batch (See also §3.3.1): 0bssssssss**1** 0x04 0x02 0x00 0xmmmm 0xMMMM <Delta> <Resol> 0xtt 0xmmmm 0xMMMM : Minimum and Maximum recording interval 0bssssssss**1** : Batch (size+flag). 0bssssssss: is the size (in byte) of configuration string after attribute ID (0bssssssss1 right shifted). Here the size is 0x0E. So the Batch field is 0x0E*2+1=0x1D. 0x00: is the index of required field (The Counter attribute has got only one field*):* 0 : Counter value [U32] <Delta>: the required delta value <Resol>: the required resolution 0xtt: the tag value (0bxxxxxzzz: xxxx = tag label, zzz=tag size);				
		Configure reporting response	0x11 +en	0x07	0x00 0x0F	0x00 0x00 0x04 0x02				0x00 : status (first byte) 0x00 : direction (next byte) 0x040x02 : attributeID (last two bytes)
						Batch (See also configure reporting): 0x00 0x01 0x04 0x02				
		Read reporting configuration	0x11 +en	0x08	0x00 0x0F	0x00 0x04 0x02				
						Batch (See also configure reporting): 0x01 0x04 0x02				
		Read reporting configuration response	0x11 +en	0x09	0x00 0x0F	0x00 0x00 0x04 0x02 0x23 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc 0xcc 0xcc				0x00 : status (first byte) 0x00 : direction (next byte) 0x04 0x02 : attributeID (next two bytes) 0x23 : attribute type *See configure reporting for next bytes*
						Batch (See also batch configure reporting): 0x00 0bssssssss**1** 0x04 0x02 0x00 0xmmmm 0xMMMM <Delta> <Resol> 0xtt				
		Report attributes	0x11 +en	0x0A	0x00 0x0F	0x04 0x02 0x23 0xcc 0xcc 0xcc 0xcc				0x23 : attribute type 0xcc 0xcc 0xcc 0xcc Current value of the counter (Big Endian)
			0b.... ...**0**		**Batch** : Please refer to "3.3.2 Batch reporting frame format"					
	Cluster command	Command : Reset	0x11 +en	0x50	0x00 0x0F	0x00				Reset current counter value to 0

3.4.14 ILLUMINANCE MEASURMENTS CLUSTER

Clusters	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
Illuminance Measurement	MeasuredValue *(Batch reportable)*	Read attribute request	0x11 +en	0x00	0x04 0x00	0x00 0x00				Ask the Illuminance of the remote sensor.
		Read attribute response	0x11 +en	0x01	0x04 0x00	0x00 0x00	0x00	0x21	0xii 0xii	0xii 0xii : MeasuredValue Illuminance=10^((MeasuredValue-1)/10000) Illuminance unit is lux.
		Configure reporting	0x11 +en	0x06	0x04 0x00	0x00 0x00 0x00 0x21 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc				Configure the reporting of the MeasuredValue.
						0x00 : direction (first byte) 0x00 0x00 : attributeID (next two bytes) 0x21 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc 0xcc : reportable change => shall contain the minimum change to the attribute that will result in a report being issued.				
		Configure reporting response	0x11 +en	0x07	0x04 0x00	0x00 0x00 0x00 0x00				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration	0x11 +en	0x08	0x04 0x00	0x00 0x00 0x00				0x00 : direction (first byte) 0x00 0x00 : attributeID (last two bytes)
		Read reporting configuration response	0x11 +en	0x09	0x04 0x00	0x00 0x00 0x00 0x00 0x21 0xmm 0xmm 0xMM 0xMM 0xcc 0xcc				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x00 : attributeID (next two bytes)
						0x21 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc 0xcc : reportable change => contains the minimum change to the attribute that will result in a report being issued.				
		Report attributes	0x11 +en	0x0A	0x04 0x00	0x00 0x00 0x21 0xii 0xii				0x00 0x00 : attributeID 0x21 : attribute type 0xii 0xii : MeasuredValue Illuminance=10^((MeasuredValue-1)/10000) Illuminance unit is lux.
	MinMeasuredValue	Read attribute request	0x11 +en	0x00	0x04 0x00	0x00 0x01				Ask the minimum value that the remote sensor is capable to measure
		Read attribute response	0x11 +en	0x01	0x04 0x00	0x00 0x01	0x00	0x21	0xii 0xii	0xii 0xii : MinMeasuredValue IlluminanceMin=10^((MinMeasuredValue-1)/10000) IlluminanceMin unit is lux.
	MaxMeasuredValue	Read attribute request	0x11 +en	0x00	0x04 0x00	0x00 0x02				Ask the maximum value that the remote sensor is capable to measure
		Read attribute response	0x11 +en	0x01	0x04 0x00	0x00 0x02	0x00	0x21	0xii 0xii	0xii 0xii : MaxMeasuredValue IlluminanceMax=10^((MaxMeasuredValue-1)/10000) IlluminanceMax unit is lux.

3.4.15 MULTISTATE OUTPUT CLUSTER

Clusters	Attribute name	Features	Payload							Comment
			Flag[+en]	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
MultiState Output Cluster (Basic)		~~Read attribute request~~	~~0x11 +en~~	~~0x00~~	~~0x00 0x13~~	~~0x00 0x0E~~				~~Ask the Binary Input value of the sensor.~~
	~~StateText*~~	~~Read attribute response~~	~~0x11 +en~~	~~0x01~~	~~0x00 0x13~~	~~0x00 0x0E~~	~~0x00~~	~~0x42~~	~~0xnn 0xss <str1>…<strn>~~	~~0xnn : number of stateText (0xss : size of string, str1: string), … If (ApplicationType = 0x0E00200) { 0: "Confort" ; 1: "Economique" ; 2: "Hors gel" ; 3: "Arret" ; 4 : "Confort -1°" ; 5: "Confort -2°" }~~
	Number of states	Read attribute request	0x11 +en	0x00	0x00 0x13	0x00 0x4A				Request Number of state

		Read attribute response	0x11 +en	0x01	0x00 0x13	0x00 0x4A	0x00	0x20	0xnn	0xnn: Number of states
										If (ApplicationType = 0x0E00200) {
										4: 4 orders pilot wire
										6: 6 orders pilot wire
										}

		Write attribute no response	0x11 +en	0x05	0x00 0x13	0x00 0x4A		0x20	0xnn	0xnn: Number of states
										If (ApplicationType = 0x0E00200) {
										4: 4 orders pilot wire
										6: 6 orders pilot wire
										(Other values not allowed and will report an error)
										}

	ApplicationType	Read attribute request	0x11 +en	0x00	0x00 0x13	0x01 0x00				Request Application Type
		Read attribute response	0x11 +en	0x01	0x00 0x13	0x01 0x00	0x00	0x23	0x0E 0xff 0xff 0xff	0x0E 0xff 0xff 0xff : Default **IPSensor-PilotWire implements: 0x0E 0x00 0x20 0x00 : Pilot wire**
		~~Write attribute no response~~	~~0x11 +en~~	~~0x05~~	~~0x00 0x13~~	~~0x01 0x00~~		~~0x23~~	~~0x03 0xmm 0xcc 0xcc~~	~~0x0E: Multistate output 0x00: Domain 0xff 0xff: Application usage~~
	PresentValue	Read attribute request	0x11 +en	0x00	0x00 0x13	0x00 0x55				Request Present value

		Read attribute response	0x11 +en	0x01	0x00 0x13	0x00 0x55	0x00	0x20	0xnn	Present value of multistate output
										If (ApplicationType = 0x0E00200) {
										0: Confort
										1: Economique
										2: HorsGel
										3: Arret
										4: ConfortM1
										2: ConfortM2
										}
										Accepted order must stay below 'Number of states' attribute.

		Write attribute no response	0x11 +en	0x05	0x00 0x13	0x00 0x55		0x20	0xnn	Set present value of multistate output
		Configure reporting	0x11 +en	0x06	0x00 0x13	0x00 0x00 0x55 0x20 0xmm 0xmm 0xMM 0xMM 0xcc 0x00 : direction (first byte) 0x00 0x55 : attributeID (next two bytes) 0x10 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xcc : 0: Do nor report on change / 1: Report on any change				Configure the reporting of the presentValue
		Configure reporting response	0x11 +en	0x07	0x00 0x13	0x00 0x00 0x00 0x55				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x55 : attributeID (last two bytes)
		Read reporting configuration	0x11 +en	0x08	0x00 0x13	0x00 0x00 0x55				
		Read reporting configuration response	0x11 +en	0x09	0x00 0x13	0x00 0x00 0x00 0x55 0x20 0xmm 0xmm 0xMM 0xMM 0xcc				0x00 : status (first byte) 0x00 : direction (next byte) 0x00 0x55 : attributeID (next two bytes) 0x10 : attribute type *See configure reporting for next bytes*
		Report attributes	0x11 +en	0x0A	0x00 0x13	0x00 0x55 0x20 0xcc				0x20 : attribute type 0xcc : Current presentValue

*Statetext and ApplicationType/write attribute are not implemented yet

3.4.16 TELE INFORMATION CLIENT (TIC) CLUSTER

Important Notes about payload size:

These remarks are mainly (only for current list of sensor) applicable for TIC cluster management, as it can work with big data payloads.

Next two advices must/can be achieved using correct "TICFieldSelector" according to *"§ 3.5.4.2.6 Available TIC Subfields and conversion"* and avoid the "read attribute request of 0x0000". Prefer the command "Read filtered TICData".

- The cluster commands, response or reports may have big ZCL payload. User must take care of these two principles:
  - Do not exceed 151 bytes for ZCL payload. (either you'll got an error or no report)
  - Avoid ZCL payloads upper than 85 bytes to avoid fragmentation of 6LowPAN frame. However up to 151 bytes can be set or expected in a ZCL Payload.
- Similarly the TIC Criteria list used in configure reporting transaction (after TIC_FIELD_LIST and TIC_FIELD_SELECTOR) MUST be smaller than 24 bytes.

Cluster	Attribute name	Features	Payload							Comment
			Flag	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
TIC Information (1/3)	TIC Meter Type	Read attribute request	0x11 +en	0x00	0xcc 0xcc	0x00 0x10				Ask the currently selected type of remote meter. The selected remote meter specifies the type of frames that can be received, hence the possible read and reported TIC attributes.
		Read attribute response	0x11 +en	0x01	0xcc 0xcc	0x00 0x10	0x00	0x20	0xtt	0xtc: Meter type => This implies the accepted type of frames. *See also* "§3.5.4.2.6 Available TIC Subfields and conversion" *0x00 : Unknown* *0x01 : "Concentrateur teleport"* *0x02 : "Compt. Bleu Electr. Monophasé»* *0x03 : "Compt. Bleu Electr. Monophasé ICC»* *0x04 : "Compt. Bleu Electr. Triphasé»* *0x05 : "Compt. Jaune Electronique»* *0x06 : "Compt. Interface Clientelle Emeraude»* *0x07 : "TIC Standard (Linky) »* *0x08-0xFF : Unknown*

Clusters	Attribute name	Features	Payload							Comment
			Flag	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
TIC Information (2/3)	TIC meter Data (ICE: 0x0053 / CBE: 0x0054 / CJE : 0x0055) (1/2) *(Batch reportable)*	Read attribute request	0x11 +en	0x00	0xcc 0xcc	0xtt 0xtt				Ask the last sample of data from the remote meter.
						0xcc 0xcc : Cluster ID can be "0x0053 => ICE Meter TIC data" or "0x0054 => CBE Meter TIC data" or "0x0055 => CJE Meter TIC data" or "0x0056 TIC standard(Linky)" 0xtt 0xtt : 0x00 0x00 for cluster IDs 0x0053, 0x0054, 0x0055 and 0x0056. And only if cluster ID is 0x0053 attribute ID can take following values: 0x0001 for "ICE p", 0x0002 for "ICE p1" *See also "§3.5.4.2.6 Available TIC Subfields and conversion"*				
		Read attribute response	0x11 +en	0x01	0xcc 0xcc	0xtt 0xtt	0x00	0x41 or 0x43	0xss <TICFieldList> Or 0xss0xss <TICFieldList>	
						0xcc 0xcc: clusterID (See "Read attribute request" comment) 0xtt 0xtt: attributeID (See "Read attribute request" comment) '0xss' or '0xss 0xss': Size of <TICFieldList > field <TICFieldList > contains a variable length list of TIC fileds. The variable list of TIC fileds is defined by the <TICFieldSelector>, 64 bits Big endian, for which each bit set to 1 indicate that the corresponding field is present. The available values and the corresponding descriptor depend on the selected meter and its current working mode. *See also "§3.5.4.2.6 Available TIC Subfields and conversion"*				
		Configure reporting *	0x11 +en	0x06	0xcc 0xcc	0x00 0xtt 0xtt 0x41 0xmm 0xmm 0xMM 0xMM 0xss <TICFieldReportSelector> <TICFieldList>				Configure report : 0x00 : direction (first byte)
			0xcc 0xcc: clusterID (See "Read attribute request" comment) 0xtt 0xtt: attributeID (See "Read attribute request" comment) 0x41 : attribute type 0xmm 0xmm : minimum reporting interval 0xMM 0xMM : maximum reporting interval [0xFF 0xFF => no report] 0xss: Size of <TICFieldReportSelector> + <TICFieldList> <TICFieldReportSelector> is a 64 bits Big endian, for which each bit set to 1 indicate that the corresponding field must be reported. <TICFieldList> = <TICFieldSelector><Variable List of criteria> <TICFieldSelector> is a 64 bits Big endian, for which each bit set to 1 indicate that the corresponding field must used as criteria and will be filled in the remaining parameter string(<Variable length list of criteria> ***NOTE:*** *The MSB bit of this descriptor is set to 1 if the last TIC frame received is considered as obsolete .Obsolescence is contextual to real information recurrence on the Meter. It is different according to Cluster and Attributes IDs, but mainly varies from 5 Seconds for a CBE Meter to up to 2 minutes for ICEp1(0x0053/0x0002) attribute*. For each bit present in the < TICFieldReportSelector > the corresponding field will be reported. For each bit present in the < TICFieldSelector > a corresponding trigger criteria MUST be set in the <Variable List of criteria>. Any criterion has got the same type and size that its corresponding TIC filed value. If a field is an "**unsigned numerical type value**" the criteria is the minimal reportable change in absolute value. The sense of evolution, increasing or decreasing, can't be a criterion. For any positive numerical field all bits set to 1 or to 0 defines "No report on this field change". Notice that value of 1 means report at ANY change. If a field is a "**signed numerical type value**" the criteria is the minimal reportable change in absolute value. The sense of evolution, increasing or decreasing, can't be a criterion. For these signed numerical field all bits set to 0 or all bits set to 1 except most significant bit defines "No report on this field change". If a field is a "**Character**" value, the reporting is triggered if the Character has changed and becomes or was like the criterion Character. If the criterion is '?' then any change trigger a reporting. If the criterion is '^' then no trigger will occur. If a field is a "**bitfield**" (Ex U32xbe for TIC_STD). The delta field is view as a mask and the reporting is triggered if one of the selected bit in the mask as changed. If a field is a "**date**" " **YMDhms** " or " **hmDM** " or "**DMh**" type value, the reporting is triggered upon the following assessment: One of the specified byte in the criterion (without 0x00 or 0xFF), at least, has changed since last reporting AND each corresponding value byte is multiple of the criterion. This implementation allows periodic triggering with user defined period. Examples: For a"**YMDhms**" field: Setting 0x00,0x00,0x00,0x00,0x0A,0x00 will trigger a reporting if Minute has changed and Minute is a multiple of 10 or 0. This example can be used to trig a "Point 10 minutes" reporting. For a "**DMh**" field: Setting 0x00,0x01,0x02 will trigger reporting if Month or Hour have changed and Month is multiple of 1 (any new month), and Hour is an ven number or 0. If a field is a "date" prefixed with season character "S**YMDhms** ", then "**Character**" criterion and "**date**" criterion are simply "ored". If the above case is combined with a numerical suffixe like "S**YMDhmsU8** ","S**YMDhmsU8** " or "S**YMDhmsU24**" the numerical criterion is also "ored". If a field is a "**String type**" value, the reporting is triggered upon the following assessment: The last reported string (S1) and the read value (S2) are different, and one of the two strings (S1 and S2) matches the given criterion (C). Strings Sx and C are matching if all chars of C are the same than all chars of Sx. For string matching "?" (ascii 0x3F) and "*" have a special meanings: "?" matches ANY characters once. "*" matches ANY characters zero or many. Examples: Criterion "PM.." for field "PTEC" will trigger a reporting for any change of the "PTEC" field that changes the value FROM or TO the value "PM..". Criterion "????" or "*" for field "PTEC" will trigger a reporting for any change of the "PTEC" field. Criterion "P*" for field "PTEC" will trigger a reporting for any change of the "PTEC" field than change value. FROM or TO the value beginning with 'P'. Nota: When this "Configure reporting" string is sent, all previous reporting configuration is fully replaced by this new configuration, even if some fields are not part of the new reporting configuration. *See also "§3.5.4.2.6 Available TIC Subfields and conversion"*				

Clusters	Attribute name	Features	Payload							Comment
			Flag	CommandID	ClusterID	Payload				
						Attribute ID	status	Attribute type	data	
		Configure reporting * *(batch)*	0x11 +en	0x06	0xcc 0xcc	0x00 0xtt 0xtt 0x41 0xmm 0xmm 0xMM 0xMM 0xss <TICFieldReportSelector> <TICFieldList>				Configure report : 0x00 : direction (first byte)
TIC Information (3/3)	TIC meter Data (ICE: 0x0053 / CBE: 0x0054 / CJE : 0x0055)(2/2) *(Batch reportable)*					**Batch** (See also §3.3.1): 0bssssssss**1** 0xtt 0xtt [0xii 0xmmmm 0xMMMM <Delta> <Resol> 0xtt]... *For each required field:* 0xmmmm 0xMMMM : Minimum and Maximum recording interval 0b0ssssssss: is the size (in byte) of configuration string after attribute ID (0bssssssss1 right shifted) 0xii: is the index of required field ***For the TIC fields Index and data type, please refer to chapter §3.5.4.2.6 which describe all possible retrieved values inside according to each type of EDF meters. The TIC field index can be found in columns 'BIT' or 'BITNUM', and the data type can be found in the column 'Type'.*** ***NOTICE: Only numerical scalar fields can be retrieved through Batch reports. In case a none scalar field is requested an error will be returned as response (see §2.1.5).*** <Delta>: the required delta value <Resol>: the required resolution 0xtt: the tag value (0bxxxxxzzz: xxxx = tag label, zzz=tag size);				
		Configure reporting response	0x11 +en	0x07	0xcc 0xcc	0x00 0x00 0xtt 0xtt 0xtt 0xtt: attributeID (See "Read attribute request" comment)				0x00 : status (first byte) 0x00 : direction (next byte)
						Batch (See also configure reporting): 0x00 0x01 0xtt 0xtt				
		Read reporting configuration*	0x11 +en	0x08	0xcc 0xcc	0x00 0xtt 0xtt (See "Read attribute request" comment)				0x00 : direction (first byte) 0xtt 0xtt: attributeID
						Batch (See also configure reporting): 0x01 0xtt 0xtt				
		Read reporting configuration response *	0x11 +en	0x09	0xcc 0xcc	0x00 0x00 0xtt 0xtt 0x41 0xmm 0xmm 0xMM 0xMM 0xss <TICFieldList> or 0x00 0x00 0xtt 0xtt **0x43** 0xmm 0xmm 0xMM 0xMM **0xss0xss** <TICFieldList>				0x00 : status (first byte) 0x00 : direction (next byte) 0xtt 0xtt: attributeID (See "Read attribute request" comment) *See "Configure reporting" feature for next fileds definition.*
						Batch (See also configure reporting): 0x00 0bssssssss**1** 0xtt 0xtt [0xii 0xmmmm 0xMMMM <Delta> <Resol> 0xtt] ...				
		Report attributes *	0x11 +en	0x0A	0xcc 0xcc	0xtt 0xtt 0x41 0xss <TICFieldList> Or 0xtt 0xtt 0x43 0xss 0xss <TICFieldList> *See "Read attribute response" feature for next 0xss and <TICFieldList> definition.*				0xtt 0xtt: attributeID (See "Read attribute request" comment) 0x41 or 0x43: attribute type
			0b.... ...**0**		**Batch :** Please refer to "§3.3.2 Batch reporting frame format"					
		Command : Read filtered TICdata *	0x11 +en	0x50	0xcc 0xcc	0x00 0xtt 0xtt <TICFieldSelector>				0xcc 0xcc: attributeID (See "Read attribute request" comment) 0x00 : Request read filtered list of field 0xtt 0xtt: attributeID (See "Read attribute request" comment) <TICFieldSelector> is a long word (64 bits Big endian), for which each bit set to 1 indicate that the corresponding field must be reported. Upon this command the server respond using the "Read attribute response" frame with all required field selected by TICFieldSelector.

3.5 CLUSTER SPECIFIC DETAILS

3.5.1 CONFIGURATION CLUSTER

3.5.1.1 OVERVIEW

A specific cluster has been implemented to configure some attributes or to know the different services of a device:

- The "WIP address" of the client where the auto-reporting will be sent,
- The "SimpleDesc" or "Desc" is the descriptor of services provided by a device.
- The configuration mode allowing temporary more awaked state from a sleeping sensor.
- The Node power descriptor giving various data about current power supply of the sensor.

The "Node Power Descriptor" attribute is Batch reportable (Cf. §3.3). See "Batch reportable fields" in the "Configuration cluster" messages set description §3.4.2.

3.5.1.1.1 DEPENDENCIES

None

3.5.1.1.2 ATTRIBUTES

For convenience, attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such as the most significant nibble specifies the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in the following table:

Attribute Set Identifier	Description
0x000	Configuration
0x001 – 0xfff	reserved

3.5.1.1.3 CONFIGURATION SET

The Configuration attribute set contains the attributes summarized in the following table:

Identifier	Name	Type	Range	Access	Default	Mandatory/optional
0x0000	WIPAddress	Octet string	0-16 octets	Read/Write	Null string	M
~~0x0001~~	~~PanID~~	~~Unsigned 16-bit integer~~	~~0-0xffff~~	~~Read/Write~~	~~0xabcd~~	~~M~~
0x0002	SimpleDesc	Structure	-	Read	-	M
0x0003	Connexion	Octet string	0 | 18 octets	Read/Write	Null string	M
0x0004	Desc	Structure	-	Read	-	M
0x0005	Configuration mode	Unsigned 16-bit integer	0-0xffff	Read/Write	0	O *(Only available on sleeping devices)*
0x0006	Node Power Descriptor	Octet string	-	Read	0	M

3.5.1.1.3.1 *WIPADDRESS* ATTRIBUTE

WIPAddress represents the address where the sensor sends auto-reporting. If *WIPAddress* is Null string, then no auto-reporting will be done.

WIPAddress is updated by writing the full client address in this attribute or by setting the length of the *Octet string* to 0. In this last case, sender address will be used. *WIPAddress* is also updated automatically when a configure report frame is send to the sensor. In this case too, the sender address will be used.

Notice: This attribute is kept for backward compatibility purpose but it's deprecated. "Connexion" should be used instead.

3.5.1.1.3.2 *SimpleDesc* Attribute

SimpleDesc represents the descriptor of services provided by a device.

Notice: This attribute is kept for backward compatibility purpose but it's deprecated. "Desc" should be used instead.

3.5.1.1.3.3 *Connexion* Attribute

The full UDP connexion (IPv6 @ + Port) where the sensor sends auto-reporting.

Connexion is updated by writing the full IPv6 address + UDP Port in this attribute or by setting the length of the *Octet string* to 0. In this last case, sender address and source port will be used. *WIPAddress* can be updated by another way as explained in §3.5.1.1.3.1.

3.5.1.1.3.4 *Desc* Attribute

Desc represents the descriptor of endpoints/services provided by a device.

3.5.1.1.3.5 *Configuration mode* Attribute

Configuration mode represents the remaining time in seconds of mode configuration (Device ON). Notice that a value of 0 means back to regular sleep mode.

This attribute is available only if the device is a sleeping device.

When writing this attribute:

The duration defines the time in seconds that the device will stay on. Notice that 0 means immediately back to sleep mode. By the way the duration can be configured from 1 second to 65535 seconds (about 18h12 minutes).

IMPORTANT NOTES:

- Setting configuration mode more than a few seconds may empty the energy reserves of a battery powered or harvesting device. SO THINK ABOUT USING SHORT (less than 3 minutes) CONFIGURATION PERIOD.
- On most of Battery powered or harvesting devices, the USER Button can be used to set configuration mode locally on the device. When using the Button, the configuration Mode is programmed for duration of 10 minutes.

3.5.1.1.3.6 *Node Power Descriptor* Attribute

Node Power Descriptor represents the power mode and supply characteristics of the device. The Byte string data is composed by 3 different parts. First byte represents the working power mode of the sensor, the second byte represent the power sources of the sensor, where the last Words are the current level of each selected power source.

Node power descriptor		
8-bits	8-bits	N x 16 bits
Power mode	*Power sources*	*Power sources level*

Power mode *Bits field*	Description
0x00	ON when idle
0x01	Periodically ON
0x02	ON on user event (Button or specific configuration cmd* or …)
Others	*Reserved*
** See also §3.5.1 Configuration cluster where specific attribute and command are defined to manage explicitly the "configuration(ON) mode"*	

Note : therefore a sleep sensor could return 0x03 (0x01+x0x2)

Power sources* *Bits field*	Description
0x01	Constant (mains) OR external power
0x02	Rechargeable battery
0x04	Disposable battery
0x08	Solar harvesting
0x10	TIC harvesting
Others	*Reserved*

**Note: 0x00 => Undefined should not occur.*

Power sources level	Description
16-…	For each "Bit number" selected in "Power sources" field, a "Power source level" in V/1000 coded in a word big-endian is set.

3.5.1.1.4 Commands Received

Command IDs for the Simple TIC-Information cluster are listed in the following table:

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Reboot device	M

3.5.1.1.4.1 Reboot device

This command has not payload.

3.5.1.1.4.1.1 Effect on Receipt

On receipt of the "Reboot device" command the server initiate a "restart" on the device similar to a "Power On Reset".

Notice that after this restart all "Stored configuration values are NOT lost". This concerns information like PANID, Reporting configuration,

3.5.1.1.5 Commands Generated

No cluster specific commands are generated by the Server cluster.

3.5.1.2 Client

3.5.1.2.1 Dependencies

None

3.5.1.2.2 Attributes

The Client cluster has no attributes.

3.5.1.2.3 Commands Received

No cluster specific commands are received by the Client cluster.

3.5.1.2.4 Commands Generated

No cluster specific commands are generated by the Client cluster.

3.5.2 BINARY CLUSTER

3.5.2.1 OVERVIEW

This cluster manage a binary entry, it deliver the current state and the number of state change.

This cluster can manage to major kind of physical data sources. Either a source that toggle between two states (Door state, Presence …), or sources that only trig an impulsion (S0 counter, Pulse counter, …). Depending on the sensor type the given attributes may have different senses. To help specific configuration the system can be configured to:

- Select polarity of the Active/Inactive Binary state information
- Select 0 to 1 transitions, 1 to 0 transitions or both (Edge counting selection), counter information.

3.5.2.2 SERVER

3.5.2.2.1 DEPENDENCIES

None

3.5.2.2.2 ATTRIBUTES

For convenience, attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such as the most significant nibble specifies the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in the following table:

Attribute Set Identifier	Description
0x000	Binary information set
0x001 – 0xfff	reserved

3.5.2.2.3 BINARY INFORMATION SET

The Binary information set contains the attributes summarized in the following table:

Identifier	Name	Type	Range	Access	Default	Mandatory/optional
0x0054	Polarity	Boolean	0/1	R/W	0	M
0x0055	Present value	Boolean	0(Inactive)/1(Active)	Read	0	M
0x0100	Application type	Unsigned 32 bit	0- 4294967295 (0xFFFFFFFF)	R/W	(Cf 3.5.2.2.6)	M
0x0400	Edge counting selection	Bitmap 8 bits	0x00: Deactivate edge counting 0x01: Falling 0x02 : Rising 0x03 : Both 0x04: Poll changes	R/W	1	M
0x0401	Debounce period (ms)	Unsigned 16 bits integer	0000 – 65535 (about 1 minute max)	R/W	250	M
0x0402	Counter	Unsigned 32 bits integer	0- 4294967295 (0xFFFFFFFF)	Read	0	M

3.5.2.2.4 POLARITY INFORMATION SET

The Polarity attribute set contains only the attribute representing if the physical input is active when 1 (Polarity = 1) or active when 0 (Polarity = 0).

3.5.2.2.5 BINARY INFORMATION SET

The Present value information attribute set contains only the attribute representing the current state of the device:

Usually:

Present Value = 0 represents False, Off, Normal, Inactive, Empty
Present Value = 1 represents True, On, Alarm, Active, Occupied

3.5.2.2.6 APPLICATION TYPE SET

The application type attribute informs about the current kind of application of the cluster. This should be set according to the sensor type or sensor usage, and preferably, according to the usual ZigBee specifications.

Here are examples of typical configurations for Nke Watteco's sensors:

Sensor	Application type	Polarity	Present value
Motion/Closure	0x03 0x01 0x00 0x02	1	0x03: Binary input / 0x01: Security domain / 0x00 0x02: Motion detection 1: Closed 0: Opened
Undefined	0x03 0xFF 0xFF 0xFF		This should be the default value for an application defined usage of Binary Input. Then the specific commissioning procedure may Set the attribute.

3.5.2.2.7 EDGE COUNTING SELECTION

The Edge counting selection attribute set contains only the attribute that define the type of counting either Transition from 0 to 1 rising or from 1 to 0 falling or both:

Usage notes:

- If the Edge counting selection is set to BOTH, the debounce period is not used.

3.5.2.2.8 DEBOUNCE PERIOD

The Debounce period attribute set contains only the attribute that define the delay that is used to accept a new "Pulse/Edge" after the last detected.

Usage notes:

- This information is processed as close as possible of the required parameter in milliseconds. However the programmable step depends a lot from the device/sensor hardware capabilities. As an example, most of the sleeping sensors can have a minimal timing resolution of 125 ms, then the required debouce will be processed modulo 125 ms: 0-124 ➔ 0, 125-254 ➔125, ….
- If the debounce period is set to 0 then the device will try to count all the programmed Edge counting selection.
- If the Edge counting selection is set to BOTH, the debounce period is not used.

3.5.2.2.9 COUNTER INFORMATION SET

The counter information attribute set contains the number of state change of the sensor:
When the number of samples has reached its maximum, then there is no more accumulation. The client system must reset the counting information to restart the summation.

3.5.2.2.10 COMMANDS RECEIVED

Command IDs for the Binary cluster are listed in the following table:

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Reset	M

3.5.2.2.10.1 RESET COMMAND

This command does not have any payload.

3.5.2.2.10.1.1 EFFECT ON RECEIPT

When this command is received, the device shall reset all its counting information.

3.5.2.2.11 COMMANDS GENERATED

No cluster specific commands are generated by the Server cluster.

3.5.2.3 CLIENT

3.5.2.3.1 DEPENDENCIES

None

3.5.2.3.2 ATTRIBUTES

The Client cluster has no attributes.

3.5.2.3.3 COMMANDS RECEIVED

No cluster specific commands are received by the Client cluster.

3.5.2.3.4 COMMANDS GENERATED

No cluster specific commands are generated by the Client cluster.

3.5.3 SIMPLE METERING –LIKE CLUSTER

3.5.3.1 OVERVIEW

A specific cluster has been implemented to manage a device having an electrical metering sensor.

3.5.3.2 SERVER

3.5.3.2.1 DEPENDENCIES

None

3.5.3.2.2 ATTRIBUTES

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such as the most significant nibble specifies the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in the following table:

Attribute Set Identifier	Description
0x000	Simple metering information
0x001 – 0xfff	reserved

3.5.3.2.3 SIMPLE METERING INFORMATION SET

The Simple metering information attribute set contains the attributes summarized in the following table:

Identifier	Name	Type	Range	Access	Default	Mandatory/optional
0x0000	CurrentMetering	Octet string	-	Read	0	M
0x8000	CurrentCalibration	Octet string	-	Read/write	0	O*

3.5.3.2.3.1 *CURRENTMETERING* ATTRIBUTE

CurrentMetering represents the active and reactive energy accumulation and the active and reactive power.

The Byte string data is composed by 6 different elements.

First byte represents the length of the following data, next 24-bit represent the active energy in W.h, and the second 24-bit represents the reactive energy in VAR.h. The following 16-bit represent the number of sample of the summation (one per minutes). Next 16-bit represent the Active power in Watt and the last 16-bit represents the reactive power in VA.

CurrentMetering				
24-bit (signed int)	24-bit (signed int)	16-bit (unsigned int)	16-bit (signed int)	16-bit (signed int)
Active energy (W.h)	*Reactive energy (VAR.h)*	*Number of sample (one per minutes)*	*Active power (W)*	*Reactive power (VAR)*

When the number of samples has reached its maximum, then there is no more accumulation. Reset the metering information to restart the summation.

Important note: Real values and units are Meter dependant for the simple metering that mirror the TIC values. Please refer to nke Watteco's "TIC sensor User Guide" for real Currentmetering affectations.

3.5.3.2.4 CALIBRATION INFORMATION SET

This attribute may not be present, if the calibration is not implemented or if the simple-metering cluster is only a "front-end" for another energy meter like "Electrical meters, or …"

Important note: This information is only useful for real metering devices like Plugs or Tore inputs. It can't be used for simplemetering that mirror TIC Meter input.

3.5.3.2.4.1 *CurrentCalibration* Attribute

CurrentCalibration represents the necessary calibration parameters for calculation of Active and Reactive energy powers and accumulations.

The Byte string data is composed by 5 different elements.

Each of the elements are 16 bits signed values.

CurrentCalibration				
8bits	16-bit (signed)	16-bit (signed)	16-bit (signed)	16-bit (signed)
Programmable resistor value (0-255) only for external Tore	*Active power multiplier*	*Active power divisor*	*Reactive power multiplier*	*Reactive power divisor*

3.5.3.2.5 Commands Received

Command IDs for the Simple metering-like cluster are listed in the following table:

Command Identifier Field Value	**Description**	**Mandatory / Optional**
0x00	Reset	M
0x80	Calibrate	O

3.5.3.2.5.1 Reset command

This command does not have any payload.

3.5.3.2.5.1.1 Effect on Receipt

When this command is received, the device shall reset all its metering information. Active energy, Reactive energy and the number of samples are set to null.

3.5.3.2.5.2 Calibrate command

The payoad of Calibrate command is a byte string described as follow:

Type	**Size**	**Max Current**	**Current Active Power**	**Current Reactive Power**	**Mandatory/optional**
0x41	0xss (0x06)	0xnnnn (0 to avoid programmable resistor calibration)	0xnnnn (0 to avoid Active power calibration)	0xnnnn (0 to avoid ReActive power calibration)	M

Important note: This command is only useful for real metering devices like Plugs or Tore inputs. It can't be used for simplemetering that mirror TIC Meter input.

3.5.3.2.5.3 Effect on Receipt

When this command is received, the device uses the next measurements Active and/orReactive (see also below) to update the calibration attribute value(s). Hence using a referent Resistive and/or Capacitive load the measurement of the plug may be calibrated, setting automatically new multiplier(s) and divisor(s).

NOTES about the calibrate command parameters:

- Parameters may be Negative, ie for a system injecting energy through the plug, the active power may be negative. Range for each parameter is from -32765 to +32764.
- If one of the calibration parameters is null then corresponding Energy coefficients (E2Pot or multiplier/divisor) are not calibrated.

3.5.3.2.6 Commands Generated

No cluster specific commands are generated by the Server cluster.

3.5.3.3 Client

3.5.3.3.1 Dependencies

None

3.5.3.3.2 Attributes

The Client cluster has no attributes.

3.5.3.3.3 Commands Received

No cluster specific commands are received by the Client cluster.

3.5.3.3.4 Commands Generated

No cluster specific commands are generated by the Client cluster.

3.5.4 TIC-INFORMATION CLUSTER

3.5.4.1 OVERVIEW

A specific cluster has been implemented to manage a French electrical meter named "Compteur Bleu Electronique" that deliver TIC meter information. Read ERDF document "ERDF-NOI-CPT_02E" for detailed information about the available TIC meter data.

The current version focuses on data available from CT, CBEMM, CBEMM-ICC, CBETM, CJE and ICE types of electrical meters.

3.5.4.2 SERVER

3.5.4.2.1 DEPENDENCIES

None

3.5.4.2.2 ATTRIBUTES

For convenience, the attributes defined in this specification are arranged into sets of related attributes; each set can contain up to 16 attributes. Attribute identifiers are encoded such as the most significant nibble specifies the attribute set and the least significant nibble specifies the attribute within the set. The currently defined attribute sets are listed in the following table:

Attribute Set Identifier	Description
0x000	TIC Meter Informations
0x001 – 0xfff	reserved

3.5.4.2.3 TIC INFORMATION SET

The TIC information attribute set contains the attributes summarized in the following table:

Identifier	Name	Type	Range	Access	Default	Mandatory/optional
0x0000	TICData-General	Octet string	-	Read	0	M
0x0001	TICData-ICEp	Octet string	-	Read	0	O
0x0002	TICData-ICEp1	Octet string	-	Read	0	O
0x0010	TICMeterType	Unsigned 8-bit integer	0x00-0xff	R/W	0x00	M

3.5.4.2.3.1 *TICDATA* ATTRIBUTE

TICData represents all the data fields that can be report by a TIC meter.

The Octet string data, in that case also named **<TICFieldList>** is composed of

- a "Size" byte indicating the length of the "Octet string",
- a specific "Subfield Selector" bit field (64 bits) selecting ALL or PART of the TIC subfields available for the meter as defined in chapter "§3.5.4.2.6 Available TIC Subfields and conversion",
  NOTE: The MSB bit of this descriptor is set to 1 if the last TIC frame received is older than 5 seconds.
- Followed by up to 28 subfields elements each of them are defined in chapter "§3.5.4.2.6 Available TIC Subfields and conversion".

NOTA: When using the Usual "Read Attribute" commandID, ALL the subfields of the corresponding meter are returned. Meanwhile specific "Read Filtered TICData" command described in chapter "§3.5.4.2.4.1 Read filtered TICData" can be used to restrict reading to PART of existing subfields.

3.5.4.2.3.2 *TICMeterType* Attribute

TICMeterType represents give the currently selected meter type. Frame fields received from the TIC meter are filtered according to the selected type. This means that the descriptor matches the meter type frames definition. See chapter "§3.5.4.2.6 Available TIC Subfields and conversion".

The TICMeterType attribute can be read, write. The value is stored hence still available after device powered off/on. Notice that the MT_NULL meter type can be written to force discovering of newly connected meter.

TICMeterType can take following values. According to the TICMeterType, the last column gives the chapter that describes the corresponding descriptor:

Type name	value	Counter description	Used descriptor
MT_UNKNOWN	0	*Using this value tells the TIC sensor to not decode any TIC information.*	-
MT_CT	1	Concentrateur Teleport	ERDF-CBE : § 3.5.4.2.6.1
MT_CBEMM	2	Compteur Bleu Electronique Monophasé Multitarif	ERDF-CBE : § 3.5.4.2.6.1
MT_CBEMM_ICC	3	Compteur Bleu Electronique Monophasé Multitarif extension ICC	ERDF-CBE : § 3.5.4.2.6.1
MT_CBETM	4	Compteur Bleu Electronique Triphasé Multitarif	ERDF-CBE : § 3.5.4.2.6.1
MT_CJE	5	Compteur Jaune	ERDF-CJE : § 0
MT_ICE	6	Compteur Clientelle Emeraude	ERDF-ICE : § 3.5.4.2.6.3
MT_STD	7	Compteur Linky (TIC Standard)	ERDF-STD : § 3.5.4.2.6.4
MT_NULL	8 . . .	*Set this value tells the TIC sensor to find again the TIC meter type.*	-

3.5.4.2.4 Commands Received

Command IDs for the Simple TIC-Information cluster are listed in the following table:

Command Identifier Field Value	Description	Mandatory / Optional
0x00	Read filtered TIC data	M

3.5.4.2.4.1 Read filtered TICData

The payoad of Read filtered TICData is described as follow:

Attribute ID	Sub field selector	Mandatory/optional
0x nn nn	0x ss ss ss ss ss ss ss ss	M

The "Attribute ID" contains the aimed attribute. Most of the time, the only concerned attribute is "0x0000", which is the TICData-General attribute. When using ICE Meter with "Tarif BASE Ax" the following complementary attributes are available and populated. "0x0001:TICData-ICEp" and "0x0002:TICData-ICEp1".

The "Sub field descriptor" is a bits field for which each bit represents a requested field from the full "TICData" Attribute. The correspondence between each bit and the TIC field is described in chapter "§3.5.4.2.6 Available TIC Subfields and conversion".

3.5.4.2.4.1.1 Effect on Receipt

On receipt of the "Read filtered TICData" command the server replies all the requested subfields that are currently available using the "Read attribute response frame".

3.5.4.2.5 Commands Generated

No cluster specific commands are generated by the Server cluster.

3.5.4.2.6 AVAILABLE TIC SUBFIELDS AND CONVERSION

The following table describes all sub-fields available in the <TICFieldList> of TICData attribute. Each of these fields will be concatenated using the ZCL type for the corresponding request. The standard Read Attribute command will return all subfields corresponding to a specific Meter Type. Meanwhile the complementary command "Read filtered field" and the "Reporting capabilities" allow request and report of only part of the subfields using the specific <FiledSelect> bit field in the client request.

3.5.4.2.6.1 « CONCENTRATEUR TELEPORT » AND « COMPTEURS BLEUS ELECTRONIQUES »

Compteur	Trame	Label	
CTEL	Longue	CT	Concentrateur de téléreport
CBEMM	Longue	MM	Trame de Compteur « Bleu » électronique monophasé multitarif
CBEMM-ICC	Longue	ICC	Trame de Compteur « Bleu » électronique monophasé multitarif évolution ICC
CBETM	Longue	TM	Compteur « Bleu » électronique triphasé multitarif (CBETM)
CBETM	Courte	TMc	Trame d'alerte courte en cas de dpassement sur une phase

	Counter/Frame type										Ipsensor types	
BIT*	CT	MM	ICC	TM	TMc		TIC Field	Label	Type	Size	Unit	ZCL type**	ZCL size
0					1		ADIR1	Avert. de Dépass. I Phase 1	String	3	A	U16	2
1					1		ADIR2	Avert. de Dépass. I Phase 2	String	3	A	U16	2
2					1		ADIR3	Avert. de Dépass. I Phase 3	String	3	A	U16	2
3	1	1	1	1	1		ADCO	Adresse compteur	String	12		CString	13
4	1	1	1	1			OPTARIF	Option tarif	String	4		CString	5
5		1	1	1			ISOUSC	Intensité souscrite	String	2	A	U8	1
6	1	1	1	1			BASE	Index option base	String	9	Wh	U32	4
7	1	1	1	1			HCHC	Index Heures creuses	String	9	Wh	U32	4
8	1	1	1	1			HCHP	Index Heures pleines	String	9	Wh	U32	4
9	1	1	1	1			EJPHN	EJP Heures normales	String	9	Wh	U32	4
10	1	1	1	1			EJPHPM	EJP Heures pointe mobile	String	9	Wh	U32	4
11		1	1	1			BBRHCJB	Tempo: Heures creuses J bleus	String	9	Wh	U32	4
12		1	1	1			BBRHPJB	Tempo: Heures pleines J bleus	String	9	Wh	U32	4
13		1	1	1			BBRHCJW	Tempo: Heures creuses J blancs	String	9	Wh	U32	4
14		1	1	1			BBRHPJW	Tempo: Heures pleines J blancs	String	9	Wh	U32	4
15		1	1	1			BBRHCJR	Tempo: Heures creuses J rouges	String	9	Wh	U32	4
16		1	1	1			BBRHPJR	Tempo: Heures pleines J rouges	String	9	Wh	U32	4
17		1	1	1			PEJP	Préavis EJP	String	2	Min	U8	1
18	1						GAZ	Index Gaz	String	7	dal ?	U32	4
19	1						AUTRE	Index Troisième compteur	String	7	dal ?	U32	5
20	1	1	1	1			PTEC	Période Tarifaire en cours	String	4		CString	5
21		1	1	1			DEMAIN	Couleur du lendemain	String	4		CString	5
22		1	1				IINST	Intensité instantanée	String	3	A	U16	2
23				1	1		IINST1	Intensité instantanée Phase1	String	3	A	U16	2
24				1	1		IINST2	Intensité instantanée Phase2	String	3	A	U16	2
25				1	1		IINST3	Intensité instantanée Phase3	String	3	A	U16	2
26		1	1				ADPS	Avertissement dépass puiss sous	String	3	A	U16	2
27		1	1				IMAX	Intensité maximale appelée	String	3	A	U16	2
28				1			IMAX1	Intensité maximale appelée P1	String	3	A	U16	2
29				1			IMAX2	Intensité maximale appelée P2	String	3	A	U16	2
30				1			IMAX3	Intensité maximale appelée P3	String	3	A	U16	2
31				1			PMAX	Puiss. Maximale tri-phasée	String	5	W	U32	4
32			1	1			PAPP	Puiss.Apparente tri-phasée	String	5	VA	U32	4
33		1	1	1			HHPHC	Horaire Heure Pleine/creuse	String	1		Char	1
34	1	1	1	1			MOTDETAT	Mot d'état du compteur	String	6		CString	7
35				1			PPOT	Présence des potentiels	String	2		CString	3
								Max size of buffer for this counter type ==>					83

***** The BIT number represents the corresponding bit of a 64 bits fields descriptor, this descriptor is always written Big endian in all frames.*

*** ZCL Type defines the kind of data of a TIC field converted to our current format. All numbers are converted to a binary format (Integer data types) sufficient to encode the original source String. All other fields are not compressed to binary format. They keep the same string length (zero padded) that the original TIC field.*

NOTA: See "§3 Application Protocol *" for reporting field selection and triggering criteria.*

3.5.4.2.6.2 « COMPTEUR JAUNE ELECTRONIQUE (CJE) »

TIC original format									"ZCL-like" Conversion		
Group name	Group Label	Field index	Group format	Read format	Nb Max bytes	TIC field size max	Description	Unit	BitNum	Type	Size max
En-tête	JAUNE	0	hh:mn:jj:mm	02d	2		Heure/Minute/Jour/Mois	hmDM	0	hmDM	4
En-tête	JAUNE	4	pt	s	2		Poste tarifaire		4	Cstring	3
En-tête	JAUNE	5	dp	s	2		Préavis de dépass.		5	Cstring	3
En-tête	JAUNE	6	abcde	05d	5		Puiss. App courante	dVA	6	U24	3
En-tête	JAUNE	7	kp	02d	2		Coef decl. Préavis	%	7	U8	1
Energies	ENERG	0	111111	06d	6		HPH ou PM	kWh	8	U24	3
Energies	ENERG	1	222222	06d	6		HCH ou HH	kWh	9	U24	3
Energies	ENERG	2	333333	06d	6		HPE	kWh	10	U24	3
Energies	ENERG	3	444444	06d	6		HCE	kWh	11	U24	3
Energies	ENERG	4	Reserved ?	06d	6		Index	kWh	12	U24	3
Energies	ENERG	5	Reserved ?	06d	6		Index	kWh	13	U24	3
Glissement G-1	PERCC	0	jj:mm:hh	02d	2		Jour/Mois/heure	DMh	14	DMh	3
Glissement G-1	PERCC	3	cg	02d	2		Code glissement		17	U8	1
Puissances maximales de la période P	PMAXC	0	11111	05d	5		Pmax P	dVA	18	U24	3
Puissances maximales de la période P	PMAXC	1	22222	05d	5		Pmax P	dVA	19	U24	3
Puissances maximales de la période P	PMAXC	2	Reserved ?	05d	5		Pmax P	dVA	20	U24	3
Puissances maximales de la période P	PMAXC	3	Reserved ?	05d	5		Pmax P	dVA	21	U24	3
Temps de dépassement de la période P	TDEPA	0	11111	05d	5		Minutes	m	22	U24	3
Temps de dépassement de la période P	TDEPA	1	22222	05d	5		Minutes	m	23	U24	3
Temps de dépassement de la période P	TDEPA	2	Reserved ?	05d	5		Minutes	m	24	U24	3
Temps de dépassement de la période P	TDEPA	3	Reserved ?	05d	5		Minutes	m	25	U24	3
Glissement G-2	PERCP	0	jj:mm:hh	02d	2		Jour/Mois/heure	DMh	26	DMh	1
Glissement G-2	PERCP	3	cg	02d	2		Code glissement		29	U8	1
Puissances maximales de la période P-1	PMAXP	0	11111	05d	5		Pmax P-1	dVA	30	U24	3
Puissances maximales de la période P-1	PMAXP	1	22222	05d	5		Pmax P-1	dVA	31	U24	3
Puissances maximales de la période P-1	PMAXP	2	Reserved ?	05d	5		Pmax P-1	dVA	32	U24	3
Puissances maximales de la période P-1	PMAXP	3	Reserved ?	05d	5		Pmax P-1	dVA	33	U24	3
Puissances souscrites de la période P	PSOUSC	0	11111	05d	5			dVA	34	U24	3
Puissances souscrites de la période P	PSOUSC	1	22222	05d	5			dVA	35	U24	3
Puissances souscrites de la période P	PSOUSC	2	Reserved ?	05d	5			dVA	36	U24	3
Puissances souscrites de la période P	PSOUSC	3	Reserved ?	05d	5			dVA	37	U24	3
Puissances souscrites de la période P+1	PSOUSP	0	11111	05d	5			dVA	38	U24	3
Puissances souscrites de la période P+1	PSOUSP	1	22222	05d	5			dVA	39	U24	3
Puissances souscrites de la période P+1	PSOUSP	2	Reserved ?	05d	5			dVA	40	U24	3
Puissances souscrites de la période P+1	PSOUSP	3	Reserved ?	05d	5			dVA	41	U24	3
Fenêtre d'écoute client	FCOU	0	hh:mn	02d	2		Heures/Minutes	hm	42	hm	2
Fenêtre d'écoute client	FCOU	2	dd	02d	2		Durée écoute	m	44	U8	1
					161			Max size of buffer for this counter type ==>			101

Format autorisant plusieurs champs (variables) par "Groupe" de données, Séparateur des données au sein d'un groupe: ':'

3.5.4.2.6.3 « COMPTEUR « INTERFACE CLIENTELE EMERAUDE » (ICE) »

IMPORTANT NOTE : Currently, for ICE Meters, the device supports only the ICE 2Q ("Deux cadrans"). The ICE 4Q will come soon.

3.5.4.2.6.3.1 ICE GENERAL (ATTRIBUT 0X0000)

												"ZCL-like" Conversion		
Etiquette	**Donnée**	**Unité**	**2.4**	**Metro. Quali.**		**BASE**		**EJP**		**MODULABLE**		**BitNum**	**Type**	**Size max**
CONTRAT	Type de tarif et option tarifaire			1	8		0		0		0	0	Cstring	8
DATECOUR	Date courante			1	6		0		0		0	1	DMYhms	6
DATE	*Date courante*		*1*		0		0		0		0	2	DMYhms	6
EA	Energie active depuis le dernier top Td minutes	Wh	1	1	3		0		0		0	3	U24	3
ERP	Energie réactive positive depuis le dernier top Td minutes	varh	1	1	3		0		0		0	4	U24	3
PTCOUR	Période tarifaire courante		1	1	4		0		0		0	5	CString	4
PREAVIS	Chaîne "DEP", "EJP", "HM", "DSM" ou "SCM"		1	1	4		0		0		0	6	CString	4
MODE	Chaîne "CONTROLE"			1	0		0		0		0	7	*Vide*	0
DATEPA1	Date du point de mesure n°1			1	6		0		0		0	8	DMYhms	6
PA1	Valeur puissance moyenne active n°1	kW		1	2		0		0		0	9	U16	2
DATEPA2	Date du point de mesure n°2			1	6		0		0		0	10	DMYhms	6
PA2	Valeur puissance moyenne active n°2	kW		1	2		0		0		0	11	U16	2
DATEPA3	Date du point de mesure n°3			1	6		0		0		0	12	DMYhms	6
PA3	Valeur puissance moyenne active n°3	kW		1	2		0		0		0	13	U16	2
DATEPA4	Date du point de mesure n°4			1	6		0		0		0	14	DMYhms	6
PA4	Valeur puissance moyenne active n°4	kW		1	2		0		0		0	15	U16	2
DATEPA5	Date du point de mesure n°5			1	6		0		0		0	16	DMYhms	6
PA5	Valeur puissance moyenne active n°5	kW		1	2		0		0		0	17	U16	2
DATEPA6	Date du point de mesure n°6			1	6		0		0		0	18	DMYhms	6
PA6	Valeur puissance moyenne active n°6	kW		1	2		0		0		0	19	U16	2
p	*Champ générique de basculement vers les energ/période*						*0*		*0*		*0*	*20*		*0*
KDC	Coefficient de préavis de dépassement	%	1	1	1		0		0		0	21	U8	1
KDCD	Coefficient de dégagement de préavis de dépassement	%	1	1	1		0		0		0	22	U8	1
TGPHI	*Tangente phi moyenne 10 minutes (2.4)*		*1*		0		0		0		0	23	U32	4
PSP	Puissance souscrite de la période tarifaire P	kW	1		0	1	2		0		0	24	U16	2
PSPM	Puissance souscrite de la période tarifaire PM	kW	1		0		0	1	2	1	2	25	U16	2
PSHPH	Puissance souscrite de la période tarifaire HPH	kW	1		0	1	2		0		0	26	U16	2
PSHPD	Puissance souscrite de la période tarifaire HPD	kW	1		0	1	2		0		0	27	U16	2
PSHCH	Puissance souscrite de la période tarifaire HCH	kW	1		0	1	2		0		0	28	U16	2
PSHCD	Puissance souscrite de la période tarifaire HCD	kW	1		0	1	2		0		0	29	U16	2
PSHPE	Puissance souscrite de la période tarifaire HPE	kW	1		0	1	2	1	2		0	30	U16	2
PSHCE	Puissance souscrite de la période tarifaire HCE	kW	1		0	1	2	1	2		0	31	U16	2
PSJA	Puissance souscrite de la période tarifaire JA	kW	1		0	1	2	1	2		0	32	U16	2
PSHH	Puissance souscrite de la période tarifaire HH	kW	1		0		0	1	2		0	33	U16	2
PSHD	Puissance souscrite de la période tarifaire HD	kW	1		0		0	1	2		0	34	U16	2
PSHM	Puissance souscrite de la période tarifaire HM	kW	1		0		0		0	1	2	35	U16	2
PSDSM	Puissance souscrite de la période tarifaire DSM	kW	1		0		0		0	1	2	36	U16	2
PSSCM	Puissance souscrite de la période tarifaire SCM	kW	1		0		0		0	1	2	37	U16	2
MODE	*Chaîne "CONTROLE"*		*1*		0		0		0		0	38	*Vide*	0
PA1MN	Puissance moyenne active 1 minute	kW		1	2		0		0		0	39	U16	2
PA10MN	Puissance moyenne active 10 minutes.	kW		1	2		0		0		0	40	U16	2
PREA1MN	Puissance moyenne réactive 1 minute signée	kvar		1	2		0		0		0	41	I16	2
PREA10MN	Puissance moyenne réactive 10 minutes signée	kvar		1	2		0		0		0	42	I16	2
TGPHI	Tangente phi moyenne 10 minutes			1	4		0		0		0	43	U32	4
U10MN	Valeur moyenne des 3 tensions composées sur 10 minutes	V		1	2		0		0		0	44	U16	2
			23	27	**92**	8	**16**	6	**12**	4	**8**			130
Attention 2.4:												Max size of buffer for this counter type ==>		**108**
DATECOUR => DATE														
MODE situé en fin de trame après PSSCM														
TGPHI entre KDCD et PSP														

3.5.4.2.6.3.2 ICE P (ATTRIBUT 0X0001)

												"ZCL-like" Conversion			
Etiquette	**Donnée**	**Unité**	**2.4**	**Metro. Quali.**		**BASE**		**EJP**		**MODULABLE**		**Bit Num**	**Type**	**Size max**	**Unpacked Absolute Poistion**
DEBUTp	Date de début période p			1	6		0		0		0	0	DMYhms	6	0
FINp	Date de fin période p			1	6		0		0		0	1	DMYhms	6	6
CAFp	Code Action Facturation période p			1	2		0		0		0	2	U16	2	12
DATE_EAp	*Date réception champs Actif Période p*			*1*	*6*		*0*		*0*		*0*	3	*DMYhms*	*6*	14
EApP	Index énergie active P période p kWh				0	1	3		0		0	4	U24	3	20
EApPM	Index énergie active PM période p	kWh			0		0	1	3	1	3	5	U24	3	23
EApHCE	Index énergie active HCE période p	kWh			0	1	3	1	3		0	6	U24	3	26
EApHCH	Index énergie active HCH période p	kWh			0	1	3		0		0	7	U24	3	29
EApHH	Index énergie active HH période p	kWh			0		0	1	3		0	8	U24	3	32
EApHCD	Index énergie active HCD période p	kWh			0	1	3		0		0	9	U24	3	35
EApHD	Index énergie active HD période p	kWh			0		0	1	3		0	10	U24	3	38
EApJA	Index énergie active JA période p	kWh			0	1	3	1	3		0	11	U24	3	41
EApHPE	Index énergie active HPE période p	kWh			0	1	3	1	3		0	12	U24	3	44
EApHPH	Index énergie active HPH période p	kWh			0	1	3		0		0	13	U24	3	47
EApHPD	Index énergie active HPD période p	kWh			0	1	3		0		0	14	U24	3	50
EApSCM	Index énergie active SCM période p	kWh			0		0		0	1	3	15	U24	3	53
EApHM	Index énergie active HM période p	kWh			0		0		0	1	3	16	U24	3	56
EApDSM	Index énergie active DSM période p	kWh			0		0		0	1	3	17	U24	3	59
DATE_ERPp	*Date réception champs Réactif Pos. Période p*			*1*	*6*		*0*		*0*		*0*	18	*DMYhms*	*6*	62
ERPpP	Index énergie réac. pos. P période p	kvarh			0	1	3		0		0	19	U24	3	68
ERPpPM	Index énergie réac. pos. PM période p	kvarh			0		0	1	3	1	3	20	U24	3	71
ERPpHCE	Index énergie réac. pos. HCE période p	kvarh			0	1	3	1	3		0	21	U24	3	74
ERPpHCH	Index énergie réac. pos. HCH période p	kvarh			0	1	3		0		0	22	U24	3	77
ERPpHH	Index énergie réac. pos. HH période p	kvarh			0		0	1	3		0	23	U24	3	80
ERPpHCD	Index énergie réac. pos. HCD période p	kvarh			0	1	3		0		0	24	U24	3	83
ERPpHD	Index énergie réac. pos. HD période p	kvarh			0		0	1	3		0	25	U24	3	86
ERPpJA	Index énergie réac. pos. JA période p	kvarh			0	1	3	1	3		0	26	U24	3	89
ERPpHPE	Index énergie réac. pos. HPE période p	kvarh			0	1	3	1	3		0	27	U24	3	92
ERPpHPH	Index énergie réac. pos. HPH période p	kvarh			0	1	3		0		0	28	U24	3	95
ERPpHPD	Index énergie réac. pos. HPD période p	kvarh			0	1	3		0		0	29	U24	3	98
ERPpSCM	Index énergie réac. pos. SCM période p	kvarh			0		0		0	1	3	30	U24	3	101
ERPpHM	Index énergie réac. pos. HM période p	kvarh			0		0		0	1	3	31	U24	3	104
ERPpDSM	Index énergie réac. pos. DSM période p	kvarh			0		0		0	1	3	32	U24	3	107
DATE_ERNp	*Date réception champs Réactif Neg. Période p*			*1*	*6*		*0*		*0*		*0*	33	*DMYhms*	*6*	110
ERNpP	Index énergie réac. neg. P période p kWh	kvarh			0	1	3		0		0	34	U24	3	116
ERNpPM	Index énergie réac. neg. PM période p	kvarh			0		0	1	3	1	3	35	U24	3	119
ERNpHCE	Index énergie réac. neg. HCE période p	kvarh			0	1	3	1	3		0	36	U24	3	122
ERNpHCH	Index énergie réac. neg. HCH période p	kvarh			0	1	3		0		0	37	U24	3	125
ERNpHH	Index énergie réac. neg. HH période p	kvarh			0		0	1	3		0	38	U24	3	128
ERNpHCD	Index énergie réac. neg. HCD période p	kvarh			0	1	3		0		0	39	U24	3	131
ERNpHD	Index énergie réac. neg. HD période p	kvarh			0		0	1	3		0	40	U24	3	134
ERNpJA	Index énergie réac. neg. JA période p	kvarh			0	1	3	1	3		0	41	U24	3	137
ERNpHPE	Index énergie réac. neg. HPE période p	kvarh			0	1	3	1	3		0	42	U24	3	140
ERNpHPH	Index énergie réac. neg. HPH période p	kvarh			0	1	3		0		0	43	U24	3	143
ERNpHPD	Index énergie réac. neg. HPD période p	kvarh			0	1	3		0		0	44	U24	3	146
ERNpSCM	Index énergie réac. neg. SCM période p	kvarh			0		0		0	1	3	45	U24	3	149
ERNpHM	Index énergie réac. neg. HM période p	kvarh			0		0		0	1	3	46	U24	3	152
ERNpDSM	Index énergie réac. neg. DSM période p	kvarh			0		0		0	1	3	47	U24	3	155
			0	6	**32**	24	**72**	18	**54**	12	**36**	Unpacked buf size		158	
Attention 2.4: Aucune de ces info n'existe pour un 2.4. Seulement à partir de 2.7								Max size of "packed buffer" for this counter type ==>						104	
Cet attribut ne concerne que la version 2.7.Connecté à un ICE 2.4 cela retourne "unsupported attribute"															

3.5.4.2.6.3.3 ICE P-1 (ATTRIBUT 0X0002)

												"ZCL-like" Conversion			
Etiquette	**Donnée**	**Unité**	**2.4**	**Metro. Quali.**		**BASE**		**EJP**		**MODULABLE**		**Bit Num**	**Type**	**Size max**	**Unpacked Absolute Poistion**
DEBUTp1	Date de début période p1			1	6		0		0		0	0	DMYhms	6	0
FINp1	Date de fin période p1			1	6		0		0		0	1	DMYhms	6	6
CAFp1	Code Action Facturation période p1			1	2		0		0		0	2	U16	2	12
DATE_EAp1	*Date réception champs Actif Période p1*			*1*	*6*		*0*		*0*		*0*	3	*DMYhms*	*6*	14
EAp1P	Index énergie active P période p1 kWh				0	1	3		0		0	4	U24	3	20
EAp1PM	Index énergie active PM période p1	kWh			0		0	1	3	1	3	5	U24	3	23
EAp1HCE	Index énergie active HCE période p1	kWh			0	1	3	1	3		0	6	U24	3	26
EAp1HCH	Index énergie active HCH période p1	kWh			0	1	3		0		0	7	U24	3	29
EAp1HH	Index énergie active HH période p1	kWh			0		0	1	3		0	8	U24	3	32
EAp1HCD	Index énergie active HCD période p1	kWh			0	1	3		0		0	9	U24	3	35
EAp1HD	Index énergie active HD période p1	kWh			0		0	1	3		0	10	U24	3	38
EAp1JA	Index énergie active JA période p1	kWh			0	1	3	1	3		0	11	U24	3	41
EAp1HPE	Index énergie active HPE période p1	kWh			0	1	3	1	3		0	12	U24	3	44
EAp1HPH	Index énergie active HPH période p1	kWh			0	1	3		0		0	13	U24	3	47
EAp1HPD	Index énergie active HPD période p1	kWh			0	1	3		0		0	14	U24	3	50
EAp1SCM	Index énergie active SCM période p1	kWh			0		0		0	1	3	15	U24	3	53
EAp1HM	Index énergie active HM période p1	kWh			0		0		0	1	3	16	U24	3	56
EAp1DSM	Index énergie active DSM période p1	kWh			0		0		0	1	3	17	U24	3	59
DATE_ERPp1	*Date réception champs Réactif Pos. Période p1*			*1*	*6*		*0*		*0*		*0*	18	*DMYhms*	*6*	62
ERPp1P	Index énergie réac. pos. P période p1	kvarh			0	1	3		0		0	19	U24	3	68
ERPp1PM	Index énergie réac. pos. PM période p1	kvarh			0		0	1	3	1	3	20	U24	3	71
ERPp1HCE	Index énergie réac. pos. HCE période p1	kvarh			0	1	3	1	3		0	21	U24	3	74
ERPp1HCH	Index énergie réac. pos. HCH période p1	kvarh			0	1	3		0		0	22	U24	3	77
ERPp1HH	Index énergie réac. pos. HH période p1	kvarh			0		0	1	3		0	23	U24	3	80
ERPp1HCD	Index énergie réac. pos. HCD période p1	kvarh			0	1	3		0		0	24	U24	3	83
ERPp1HD	Index énergie réac. pos. HD période p1	kvarh			0		0	1	3		0	25	U24	3	86
ERPp1JA	Index énergie réac. pos. JA période p1	kvarh			0	1	3	1	3		0	26	U24	3	89
ERPp1HPE	Index énergie réac. pos. HPE période p1	kvarh			0	1	3	1	3		0	27	U24	3	92
ERPp1HPH	Index énergie réac. pos. HPH période p1	kvarh			0	1	3		0		0	28	U24	3	95
ERPp1HPD	Index énergie réac. pos. HPD période p1	kvarh			0	1	3		0		0	29	U24	3	98
ERPp1SCM	Index énergie réac. pos. SCM période p1	kvarh			0		0		0	1	3	30	U24	3	101
ERPp1HM	Index énergie réac. pos. HM période p1	kvarh			0		0		0	1	3	31	U24	3	104
ERPp1DSM	Index énergie réac. pos. DSM période p1	kvarh			0		0		0	1	3	32	U24	3	107
DATE_ERNp1	*Date réception champs Réactif Neg. Période p1*			*1*	*6*		*0*		*0*		*0*	33	*DMYhms*	*6*	110
ERNp1P	Index énergie réac. neg. P période p1 kWh	kvarh			0	1	3		0		0	34	U24	3	116
ERNp1PM	Index énergie réac. neg. PM période p1	kvarh			0		0	1	3	1	3	35	U24	3	119
ERNp1HCE	Index énergie réac. neg. HCE période p1	kvarh			0	1	3	1	3		0	36	U24	3	122
ERNp1HCH	Index énergie réac. neg. HCH période p1	kvarh			0	1	3		0		0	37	U24	3	125
ERNp1HH	Index énergie réac. neg. HH période p1	kvarh			0		0	1	3		0	38	U24	3	128
ERNp1HCD	Index énergie réac. neg. HCD période p1	kvarh			0	1	3		0		0	39	U24	3	131
ERNp1HD	Index énergie réac. neg. HD période p1	kvarh			0		0	1	3		0	40	U24	3	134
ERNp1JA	Index énergie réac. neg. JA période p1	kvarh			0	1	3	1	3		0	41	U24	3	137
ERNp1HPE	Index énergie réac. neg. HPE période p1	kvarh			0	1	3	1	3		0	42	U24	3	140
ERNp1HPH	Index énergie réac. neg. HPH période p1	kvarh			0	1	3		0		0	43	U24	3	143
ERNp1HPD	Index énergie réac. neg. HPD période p1	kvarh			0	1	3		0		0	44	U24	3	146
ERNp1SCM	Index énergie réac. neg. SCM période p1	kvarh			0		0		0	1	3	45	U24	3	149
ERNp1HM	Index énergie réac. neg. HM période p1	kvarh			0		0		0	1	3	46	U24	3	152
ERNp1DSM	Index énergie réac. neg. DSM période p1	kvarh			0		0		0	1	3	47	U24	3	155
															158
			0	6	**32**	24	**72**	18	**54**	12	**36**	Unpacked buf size		158	
Attention 2.4: Aucune de ces info n'existe pour un 2.4. Seulement à partir de 2.7								Max size of "packed buffer" for this counter type ==>						**104**	
Cet attribut ne concerne que la version 2.7.Connecté à un ICE 2.4 cela retourne "unsupported attribute"															

3.5.4.2.6.4 « COMPTEUR LINKY (TIC-STANDARD) »

							"ZCL-like" Conversion		
Data	**Label**	**Time Stamped**	**Nb Bytes**	**Unit**	**Tri-phased only**	**Prod only**	**Bit num**	**Type**	**Size max**
Adresse Secondaire du Compteur	ADSC		12	Sans			0	Cstring	13
Version de la TIC	VTIC		2	Sans			1	U8	1
Date et heure courante	DATE	13	0	Sans			2	*SDMYhms*	7
Nom de la grille tarifaire fournisseur	NGTF		16	Sans			3	Cstring	17
Libellé tarif fournisseur en cours	LTARF		16	Sans			4	Cstring	17
Energie active soutirée totale	EAST		9	Wh			5	U32	4
Energie active soutirée Fournisseur, index 01	EASF01		9	Wh			6	U32	4
Energie active soutirée Fournisseur, index 02	EASF02		9	Wh			7	U32	4
Energie active soutirée Fournisseur, index 03	EASF03		9	Wh			8	U32	4
Energie active soutirée Fournisseur, index 04	EASF04		9	Wh			9	U32	4
Energie active soutirée Fournisseur, index 05	EASF05		9	Wh			10	U32	4
Energie active soutirée Fournisseur, index 06	EASF06		9	Wh			11	U32	4
Energie active soutirée Fournisseur, index 07	EASF07		9	Wh			12	U32	4
Energie active soutirée Fournisseur, index 08	EASF08		9	Wh			13	U32	4
Energie active soutirée Fournisseur, index 09	EASF09		9	Wh			14	U32	4
Energie active soutirée Fournisseur, index 10	EASF10		9	Wh			15	U32	4
Energie active soutirée Distributeur, index 01	EASD01		9	Wh			16	U32	4
Energie active soutirée Distributeur, index 02	EASD02		9	Wh			17	U32	4
Energie active soutirée Distributeur, index 03	EASD03		9	Wh			18	U32	4
Energie active soutirée Distributeur, index 04	EASD04		9	Wh			19	U32	4
Energie active injectée totale	EAIT		9	Wh		x	20	U32	4
Energie réactive Q1 totale	ERQ1		9	varh		x	21	U32	4
Energie réactive Q2 totale	ERQ2		9	varh		x	22	U32	4
Energie réactive Q3 totale	ERQ3		9	varh		x	23	U32	4
Energie réactive Q4 totale	ERQ4		9	varh		x	24	U32	4
Courant efficace, phase 1	IRMS1		3	A			25	U16	2
Courant efficace, phase 2	IRMS2		3	A	x		26	U16	2
Courant efficace, phase 3	IRMS3		3	A	x		27	U16	2
Tension efficace, phase 1	URMS1		3	V			28	U16	2
Tension efficace, phase 2	URMS2		3	V	x		29	U16	2
Tension efficace, phase 3	URMS3		3	V	x		30	U16	2
Puissance app. de référence (PREF)	PREF		2	kVA			31	U8	1
Puissance app. de coupure (PCOUP)	PCOUP		2	kVA			32	U8	1
Puissance app. instantanée soutirée phase 1	SINST1		5	VA			33	U24	3
Puissance app. instantanée soutirée phase 2	SINST2		5	VA	x		34	U24	3
Puissance app. instantanée soutirée phase 3	SINST3		5	VA	x		35	U24	3
Puissance app. max. soutirée n	SMAXN	13	5	VA			36	SDMYhmsU24	10
Puissance app max. soutirée n-1	SMAXN-1	13	5	VA			37	SDMYhmsU24	10
Point n de la courbe de charge active soutirée	CCASN	13	5	W			38	SDMYhmsU24	10
Point n-1 de la courbe de charge active soutirée	CCASN-1	13	5	W			39	SDMYhmsU24	10
Point n de la courbe de charge active injectée	CCAIN	13	5	W		x	40	SDMYhmsU24	10
Point n-1 de la courbe de charge active injectée	CCAIN-1	13	5	W		x	41	SDMYhmsU24	10
Tension moy. ph. 1	UMOY1	13	3	V			42	SDMYhmsU16	9
Tension moy. ph. 2	UMOY2	13	3	V	x		43	SDMYhmsU16	9
Tension moy. ph. 3	UMOY3	13	3	V	x		44	SDMYhmsU16	9
Registre de Statuts	STGE		8	Sans			45	U32xbe*	4
Debut Pointe Mobile 1	DPM1	13	2	Sans			46	SDMYhmsU8	8
Fin Pointe Mobile 1	FPM1	13	2	Sans			47	SDMYhmsU8	8
Debut Pointe Mobile 2	DPM2	13	2	Sans			48	SDMYhmsU8	8
Fin Pointe Mobile 2	FPM2	13	2	Sans			49	SDMYhmsU8	8
Debut Pointe Mobile 3	DPM3	13	2	Sans			50	SDMYhmsU8	8
Fin Pointe Mobile 3	FPM3	13	2	Sans			51	SDMYhmsU8	8
		208	322						
			530						297

* U32xbe = hexa big endian 32 bit bifield

3.5.4.3 CLIENT

3.5.4.3.1 DEPENDENCIES

None

3.5.4.3.2 ATTRIBUTES

The Client cluster has no attributes.

3.5.4.3.3 COMMANDS RECEIVED

No cluster specific commands are received by the Client cluster.

3.5.4.3.4 COMMANDS GENERATED

No cluster specific commands are generated by the Client cluster.

4 BR_UNCOMPRESS: THE UNCOMPRESS BATCH TOOL

In association with the sensors, nke Watteco provides a software tool useful to uncompress the batch report payloads send by the nke Watteco's sensors.

This software is provided as an executable and is called br_uncompress on a Linux OS or br_uncompress.exe on a Windows OS.

4.1 UNCOMPRESS TOOL USAGE

By using the –h option, it is possible to have the usage of this tool. Here below can be seen the usage of the br_uncompress utility:

```
USAGE:

  br_uncompress [-a] tagsz "taglbl,resol,sampletype" "..." ... < buf > result

  -a : Input buf must be considered as ascii hexa bytes either than usual raw bytes: 'hhhhhh...' or 'hh hh
hh...' or '$HH$HH$HH...'

      Allow following usages:

      echo '$10$27$00$80$03$93$20$18$00$80$10$81$83$07$0d$45$85$10$05' | ./br_uncompress -a  3 2,1.0,12

      or

      echo      "404780800a5800000442ca8a4048fd395c817e21cb9a40028fd5379de3768b4f816e75a6e376006e2d800066"      |
./br_uncompress -a  3 2,10,9 1,10,7 4,30,10 3,10,4 5,10,6 6,1,4
```

In order to be clearer, the second example given in the usage will be explained. In this example the **applicative payload to uncompress** is:

```
404780800a5800000442ca8a4048fd395c817e21cb9a40028fd5379de3768b4f816e75a6e376006e2d800066
```

Then, it can be seen that the utility br_uncompress is called with the option **–a** (which says to the program to **consider the buffer as ASCII hex bytes**).

After this option, it can be seen the number **3**. As the usage tell us, this number is the **tag size**, it is actually the number of bits used for the label (cf. §3.3.1 for further explanations).

Finally, the last argument of this command is **a triple argument list**. Each element of this list corresponds to an attribute which is batch reported. For each element, there are **three parameters**, in order these parameters are:

- **The label** used to identify the attribute reported
- **The resolution** of the attribute
- **The sample type** of the attribute

The label and the resolution are the same as the ones used when the batch reporting has been configured (cf. §3.3.1 and §3.4 for further explanations). On the contrary, the sample type is not defined by the user but is proper to the attribute. Thus, this sample type can be seen for each batchable attribute in each cluster in the paragraph: §3.4.

Here below is the table giving the corresponding number to use in br_uncompress for each sample type.

Label	Explicit type	Size (bytes)	Type n°
ST_UNDEF	Type undefined	?	**0**
ST_BL	Boolean	1 (1 lsb used)	**1**
ST_U4	Unsigned int	1 (4 lsbs used)	**2**
ST_I4	Signed int	1 (4 lsbs used)	**3**
ST_U8	Unsigned int	1	**4**
ST_I8	Signed int	1	**5**
ST_U16	Unsigned int	2	**6**
ST_I16	Signed int	2	**7**
ST_U24	Unsigned int	3	**8**
ST_I24	Signed int	3	**9**
ST_U32	Unsigned int	4	**10**
ST_I32	Signed int	4	**11**
ST_FL	Float	4	**12**

4.2 How to Interpret the uncompress tool results

After executing the command, the tool will give back quite a lot of text as answer. The results of the uncompression will be at the end of this answer after the label “UNCOMPRESS SERIE”. An example of result can be seen here:

```
UNCOMPRESS SERIE

cnt: 7
71146
71134 2 2214810

71090 1 2180
71100 1 2190
71110 1 2230
71120 1 2780
71130 1 2150
71140 1 2160

71088 4 2180
71104 4 2210
71118 4 2780
71128 4 2600

71112 5 3671
```

In this answer, the first value is 7, it appears after the text “cnt :”. This value is the frame counter of the batch frame send by the sensor. A frame counter allows to identify a frame. For further informations about the frame counter please refers to the LoRaWAN specification.

After the frame counter, all the other results can be seen as column.

The **first column** gives **timestamps**. The first timestamp corresponds to the **sending time of the batch frame.** After this one, all the others are to the **sampling time of the corresponding values** (3rd column). The time stamp is actually the time in second since the sensor had been turned on.

The **second column** contains **the label** of each batch reported attribute.

Finally, the **third column** contains **the values** of the batch reported attribute.

Suggested client behavior: In order to know the absolute sampling time for all the attribute values it is recommended to use the timestamp of the frame sending as a reference. Indeed, the time of reception is known by the client, therefore from these two data and by doing a difference with the timestamp of each sample it is quite trivial to find out the absolute sampling time for each value.

5 LIMITATIONS

It is recommended to wait the response after requesting a remote device or wait more than 500 ms to send the next one.

6 TECHNICAL SUPPORT

6.1 FREQUENTLY ASKED QUESTIONS

<Reserved>

6.2 CONTACTS

For any technical request, please contact the technical support by email: info.watteco@nke.fr