Fludia
Smart Energy Components

FM432 – User Guide

LoRaWAN IoT sensors for remote monitoring

Document Ref FLD10068 version 1.2.3

Product references:

FM432e: electricity meter optical reading
Ref: FM432e_nc_1mn, FM432e_nc_10mn, FM432e_nc_15mn, FM432e_ap

FM432ir: electricity infrared meter optical reading (SML protocol)
Ref: FM432ir_nc_1mn, FM432ir_nc_15mn, FM432ir_ap

FM432g: gas meter optical reading (ATEX)
Ref: FM432g_nc_10mn, FM432g_nc_15mn, FM432g_ap

FM432p-a: pulse reading (ATEX)
Ref: FM432p-a_nc_10mn, FM432p-a_nc_15mn, FM432p-a_ap

FM432p-n: pulse reading (non-ATEX)
Ref: FM432p-n_nc_10mn, FM432p-n_nc_15mn, FM432p-n_ap

FM432t: temperature measurement
Ref: FM432t_1mn, FM432t_10mn, FM432t_15mn

Firmware versions

This documentation refers specifically to products including the following firmware versions. In most cases, data formats are the same for previous versions, but if you have a doubt, do not hesitate to contact support@fludia.com and we will provide you with the correct document version and/or specific indications.

Product family	Element	Product references	Firmware version
FM432e	Optical head	FM432e_nc_1mn	FM210em_v5.4
		FM432e_nc_10mn, FM432e_nc_15mn	FM210em_v5.7
		FM432e_ap	FM210em_v6.0
	Radio/battery box	All references	BPR07_V3.3.7
FM432ir	Optical head	FM432ir_nc_1mn	FM210ir_v1.8
		FM432ir_nc_15mn	FM210ir_v2.0
		FM432ir_ap	FM210ir_v2.1
	Radio/battery box	All references	BPR07_v3.3.7
FM432g	Optical head	FM432e_nc_10mn, FM432e_nc_15mn	FM210g_v3.2
		FM432g_ap	FM210g_v3.3
	Radio/battery box	All references	BPR07_v3.3.8
FM432p	Radio/battery box	All references	BPR07_v3.3.6 BPR07_v3.3.9
FM432t	Radio/battery box	All references	BPR07_v5.0.2

Fludia
Smart Energy Components

Revision history

Version	Notes	Date
1.0.1	New doc replacing ref FLD9492 (FM432e User Guide EN_v1.2.1), ref FLD9177 (FM432g User Guide EN_v1.2.1), ref FLD9899 (FM432p User Guide EN_v1.2.2), and a series of documents related to message decoding and data collect API	2022-05-16
1.0.2	Corrections related to T1 message headers; Additional code examples.	2022-05-25
1.0.3	Completion of FM432t sections	2022-05-30
1.0.4	Adding FM432ir code examples	2022-05-31
1.0.5	FM432 instead of FM232 in the illustration (overview chapter)	2022-06-01
1.0.6	Firmware version FM210ir_v1.6 replaces v1.2. Clarification (chapter 7.1): message contains 15 or 20 values depending on the type of meter.	2022-08-24
1.0.7	A Byte was missing in the T1 description for FM432ir (chapters 7.2 and 8.2)	2022-10-11
1.0.8	Adding JavaScript code examples for FM432e, FM432ir, FM432g, FM432p and FM432t.	
1.1.0	Correcting php code for FM432t. Technical messages headers. New FW versions for BPR07	2023-06-07
1.1.1	§7.6. correction: index is divided by 10. §8.2.6. correction: index field is signed. §8.6. correction: index is divided by 10. §6.3. Bytes #10 and #11 are « not used ». Page 2: firmware versions are specified in a table. « Battery status » is now called « Low battery ». §9.2. hex data example changed. §12.3. and §13.3. correction T2 message structure. §7.2.4. correction: OBIS 2.8.0 and not 2.0.8 §9.5.3 correction: « Javascript » instead of « PHP »	2024-07-05
1.2.0	§6, §9, §12, §14. new FM432x_ap firmware versions (easy remote parameter adjustments)	2024-07-31
1.2.1	§4. description of Join and rejoin strategies	2024-08-20
1.2.2	§9.2.4. Only 27 Bytes in the message and not 28	2024-09-20
1.2.3	§9.4. corrections in the T2 structure table	2024-09-23

Fludia
Smart Energy Components

Table of Contents

1. Overview

The basic principle of this remote monitoring solution is to have IoT sensors (FM432x on the graph) using LoRa long range radio to transfer measurement messages to a LoRaWAN Network.

The **different types of sensors** available are:

- FM432e: optical reading of electricity meters (detecting rotating disk, or blinking light)
- FM432ir: optical reading of electricity meters in Germany (detecting rotating disk, or infrared port with SML protocol)
- FM432g: optical reading of gas meters (detecting digit rotation)
- FM432p: detecting pulse outputs of some meters (gas, water, electricity, heat…)
- FM432t: measuring temperature

The newest references with "_ap" suffix (FM432_ap, FM432ir_ap…) include the possibility to remotely change parameters like measurement frequency and remotely set a redundancy feature.

The **installation process** consists of two main steps:

1. Installing the sensors: sticking the optical sensor to the existing meter, or connecting the wires, or just positioning the box, depending on the type of sensor.
2. Joining the LoRaWAN Network: starting up the sensor so that it sends join requests to the LoRaWAN Network (on which it must have been previously registered)

Retrieving the data can be done in one of three ways:

- Directly decoding the messages received by the LoRaWAN Network
- Connecting to the server's API https://fm430-api.fludia.com/v1/API/ (if a callback has been set up between the LoRaWAN Network and Fludia data server)
- Selecting data and downloading related files from the dashboard (if a callback has been set up between the LoRaWAN Network and Fludia data server)

2. Installing the sensors

2.1 Installing the FM432e (optical reading of electricity meters)

First identify the type of meter you want to measure. It could be either an electronic electricity meter (with a blinking light) or an electromechanical electricity meter (with a rotating disk).

In case the **white cable** is plugged in the radio/battery box, it is recommended to **unplug it (**and leave only the optical side connected**) before proceeding** with the installation.

If it is an **electronic meter**, you must position the optical head **switch on B**, and if it is an **electromechanical meter**, you must position the **switch on A**:

If it is an **electronic meter**, stick the adhesive plastic mount on the meter in front of the meter blinking light (aim through the hole)

Then, position the optical head on top of the plastic mount and tight the black screw.

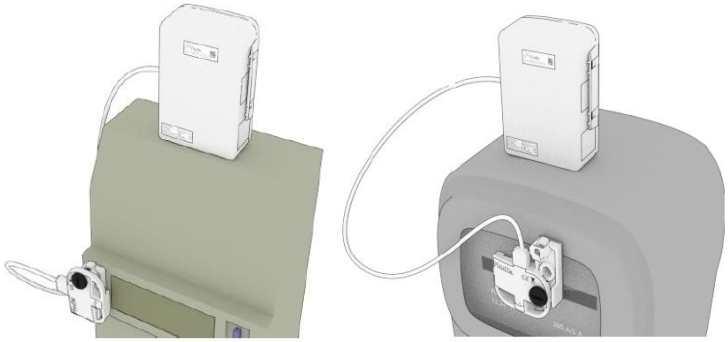
If it is an **electromechanical meter**, make sure the optical head is properly attached to the adhesive plastic mount with the help of the black screw:

Stick the system on the meter glass panel, making sure the two arrows are perfectly aligned with the meter disk (face the meter and keep your eyes at disc level for better result):

If the arrows are not completely lined up with the disc, loosen the screw, adjust position, and tighten the screw.

Connect the white cable between the optical head and the FM432 battery/radio box.

In the case of an electromechanical meter, make sure the optical head is still perfectly aligned with the disk.



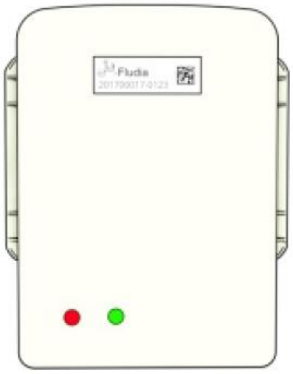
The FM432e starts measuring and tries to join the LoRaWAN network.

The optical head LEDs (lights) should blink this way:

1. Calibration: red LED blinks for 20 seconds.
2. Validation: green LED blinks every time the meter light blinks (electronic meter) or every time the disk mark (black or red) comes in front of the sensor (electromechanical meter).
3. After 3 minutes, the green light stops completely to avoid wasting battery load.

The LEDs on the radio/battery box show the progress of **the join process**:

- Red and Green LED blinking together (join in progress)
- Green LED blinking alone (join successful)
- Red LED blinking alone (join failed)



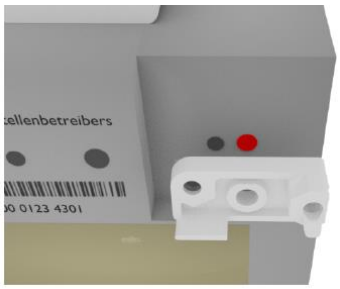
If the LEDs on the radio/battery side blink green twice, red once, green twice, red once and so on, it means that the sensor has previously joined a LoRaWAN Network (but not necessarily the one you want your sensor to join now). If you want to make sure, remove the batteries, wait for 10 seconds, and put them back on. It resets the join sequence so that your sensor can join from a fresh start.

2.2 Installing the FM432ir (optical reading of electricity meters in Germany)

First identify the type of meter you want to measure. It could be either an mME electricity meter (with an infrared port supporting SML protocol) or an electromechanical electricity meter (with a rotating disk).

In case the **white cable** is plugged in the radio/battery box, it is recommended to **unplug it (**and leave only the optical side connected**) before proceeding** with the installation.

If it is an **mME meter**, you must position the optical head **switch on B**, and if it is an **electromechanical meter**, you must position the **switch on A**:



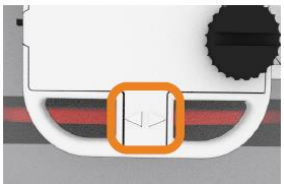
If it is an **mME meter**, stick the adhesive plastic mount on the meter in front of the meter infrared receiving LED (aim through the hole)

Then, fasten the optical head to the plastic mount.
Use the black screw to tighten the optical head to the mount.

If it is an **electromechanical meter**, make sure the optical head is properly attached to the adhesive plastic mount with the help of the black screw:

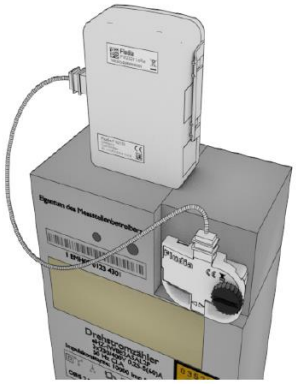
Stick the system on the meter glass panel, making sure the two arrows are perfectly aligned with the meter disk (face the meter and keep your eyes at disc level for better result):

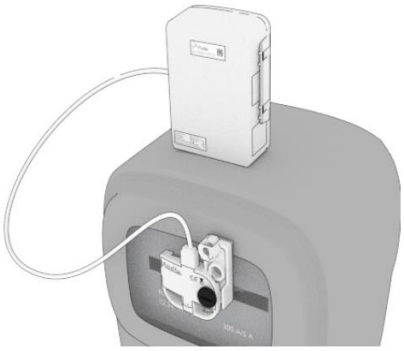
If the arrows are not completely lined up with the disc, loosen the screw, adjust position, and tighten the screw.



Connect the white cable between the optical head and the FM432 battery/radio box.

In the case of an electromechanical meter, make sure the optical head is still perfectly aligned with the disk.





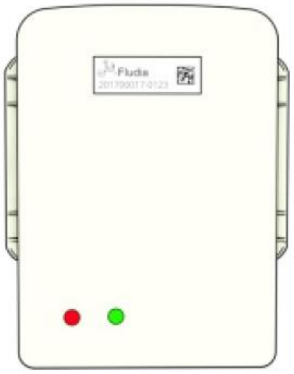
The FM432ir starts measuring and tries to join the LoRaWAN network.

The **optical head LEDs** (lights) should blink this way:

1. Calibration: red LED blinks for 20 seconds.
2. Validation: green LED each time the disk mark (black or red) comes in front of the sensor (electromechanical meter) or each time a correct infrared signal is detected (mME meter).
3. The green light stops completely to avoid wasting battery load (after 3 minutes for electromechanical, after 1 minute for mME).

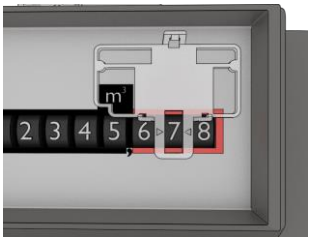
The LEDs on the radio/battery box show the progress of **the join process**:

- Red and Green LED blinking together (join in progress)
- Green LED blinking alone (join successful)
- Red LED blinking alone (join failed)



If the LEDs on the radio/battery side blink green twice, red once, green twice, red once and so on, it means that the sensor has previously joined a LoRaWAN Network (but not necessarily the one you want your sensor to join now). If you want to make sure, remove the batteries, wait for 10 seconds, and put them back on. It resets the join sequence so that your sensor can join from a fresh start.

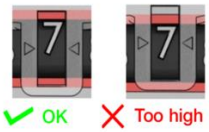
2.3 Installing the FM432g (optical reading of gas meters)



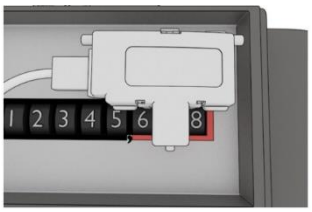
Clean the meter window.

Paste the holder on the glass of the meter, in front of the **prior-to-last digit** of the index, using the provided adhesive (already fixed on the holder).

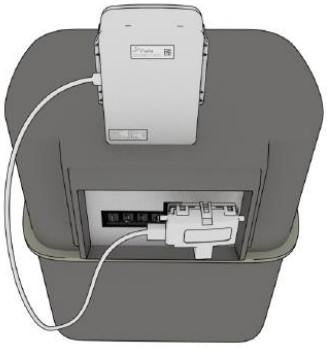
>> Make sure the arrows of the holder are perfectly at **mid-height of the frame**.



>> In some cases, the digit can be higher or lower than normal. Position the arrows at **mid-height of the frame**, **not** at mid-height of the **digit.**



Clip the optical sensor to the holder.

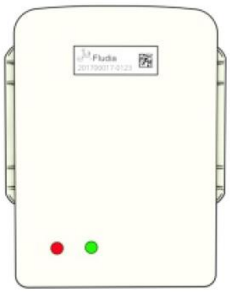


Connect the white cable between the optical head and the FM432 battery/radio box.

The FM432g starts measuring and tries to join the LoRaWAN network.

The optical head red LED (light) should blink for 20 seconds (optical head calibration period)

The **LEDs on the radio/battery box** show the progress of **the join process**:



- Red and Green LED blinking together (join in progress)
- Green LED blinking alone (join successful)
- Red LED blinking alone (join failed)

If the LEDs on the radio/battery side blink green twice, red once, green twice, red once and so on, it means that the sensor has previously joined a LoRaWAN Network (but not necessarily the one you want your sensor to join now). If you want to make sure, remove the batteries, wait for 10 seconds, and put them back on. It resets the join sequence so that your sensor can join from a fresh start.

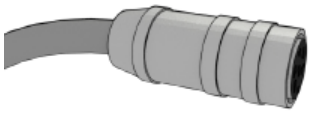
2.4 Installing the FM432p (detecting meter pulse outputs: gas, water, elec, heat…)

The FM432p comes with a cable that needs to be connected to the meter pulse interface.

By default, the end of the cable shows two wires: red wire (pulse), black wire (ground).

If the meter output shows a polarity, make sure to connect the cable accordingly.

If the cable is equipped with a “binder” connector, just connect it to the pulse output interface. The “binder” plug is wired so that either 3-5 or 4-6 configuration will work.

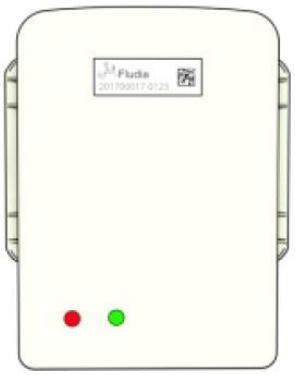


"binder" connector

The FM432p starts measuring and tries to join a LoRaWAN network either once it has detected a first pulse or once the batteries have been installed.

The **LEDs on the radio/battery box** show the progress of **the join process**:

- Red and Green LED blinking together (join in progress)
- Green LED blinking alone (join successful)
- Red LED blinking alone (join failed)



If the LEDs on the radio/battery side blink green twice, red once, green twice, red once and so on, it means that the sensor has previously joined a LoRaWAN Network (but not necessarily the one you want your sensor to join now). If you want to make sure, remove the batteries, wait for 10 seconds, and put them back on. It resets the join sequence so that your sensor can join from a fresh start.

2.5 Installing the FM432t (measuring temperature)

Start the join process by removing the batteries, waiting for 10 seconds, and putting them back on.

The **LEDs on the radio/battery box** show the progress of **the join process**:

- Red and Green LED blinking together (join in progress)
- Green LED blinking alone (join successful)
- Red LED blinking alone (join failed)

Position the FM432t box wherever you want to measure temperature. For measuring in-house temperature, it is recommended to position the FM432t box around 1 meter above the floor and, if possible, not too close to a wall (especially outside walls).

Warning: the FM432t box is not “rain-proof”. So, to measure outside temperatures, it is recommended either to position it in a place protected from the rain, or inside a “rain-proof” additional box (a full waterproof box should be avoided because of the risk of water vapor infiltration and condensation).

3. Replacing batteries in a sensor

Batteries used in FM432x products are special Lithium 3.6V batteries (Lithium thionyl chloride or Li-SoCl2). You should only replace them by similar batteries, of good quality, form renown manufacturers, and make sure that they are positioned the right way (+ side and – side are indicated both on each battery and at the bottom of the battery compartment).

The usual format of the batteries is A, but AA works also (though the energy is lower).

It is possible to use only one battery instead of two, but the expected lifetime will of course be reduced accordingly.

In case of an ATEX certified product such as the FM432g, when operating in an ATEX zone you should only use certified batteries provided by Fludia, so that the product is still considered certified.

4. Join and rejoin

4.1 Join strategy

Join request is sent when the sensor is started.

For FM432e, FM432ir and FM432g products, it is the smart optical head that is initiating the join. When the smart optical head is powered up (i.e. at the moment when it is connected to a radio-box or, if it was already connected to the radio-box without battery, at the moment when the batteries are positioned in the radio-box), it sends instruction to the radio-box to send a join request. In case this join request fails, there are two additional attempts, 10 seconds apart.

After that, there will be 3 new attempts every 12 hours.

For the FM432p product, the initial join attempt occurs when the product is powered up (i.e. at the moment when the batteries are positioned in the radio-box). In case this join request fails, there are two additional

attempts, 10 seconds apart. There are no additional retries, as long as the sensor has not detected any pulse.

When a pulse is detected, three new join attempts are made. If these attempts fail, then there will be 3 new attempts every 12 hours. The logic here is to be able to put batteries in a product beforehand (so that it can be shipped this way and easy to use right away) without it making join attempts every 12 hours. Only when the product is connected to a pulse meter will it be able to retry join requests.

4.2 Rejoin strategy

Rejoin strategy refers to a situation when a product has already joined a network but for some reason the networked has changed and does not recognize the product anymore (or if the sensor needs to be “transferred” from one network to another without having to physically initiate a join sequence on the product).

The rejoin strategy is available for product references FM432e_ap, FM432g_ap and FM432p_ap. The rejoin strategy is optional and must be activated with the help of a specific “REJOIN downlink” containing the activation value (0 or 1) and the number of unconfirmed messages before a rejoin must be initiated.

The rejoin logic is the following one:

- Every 24 hours, the product signals the LoRaWAN network that the next message will be a “CONFIRMED message”, meaning that the network should acknowledge its reception.
- As long as the product does not receive this acknowledgement, all the following messages will be “CONFIRMED messages”.
- If the product does not receive any acknowledgement after having sent n “CONFIRMED messages” (n being the number of unconfirmed messages specified in the REJOIN downlink), it initiates a new join sequence
- It will then try again every 12 hours until it manages to join.

4.3 Make a sensor forget its join

To make a sensor forget its join, you simply must cut the energy source: remove the batteries.

5. Remotely changing parameters

For products with the new -ap firmware version, some parameters can be remotely changed with the help of a downlink.

Related product references are: FM432e_ap, FM432ir_ap, FM432g_ap and FM432p_ap.

Main parameters that can be changed:

- Measurement frequency
- Number of measurement values in each uplink
- Redundancy: if activated, previous measurements are repeated.

Online tool to create the appropriate downlinks: https://www.fludia.com/resources/en-US/downlink

6. Decoding FM432e_ap messages

6.1 Types of messages

Sensor default configuration	10 minutes	15 minutes
Data message (T1)	Every 80 minutes (8 x 10 minutes). Measurement and uplink frequencies can be remotely configured with a downlink.	Every 2 hours (8 x 15 minutes). Measurement and uplink frequencies can be remotely configured with a downlink.
Service message (T2)	Every 24 hours	Every 24 hours
Technical message 1 (TT1)	Every 24 hours	Every 24 hours
Technical message 2 (TT2)	Every hour for 1 day	Every hour for 1 day

FM432e generates four kinds of messages:

- T1: contains the consumption data and is generated several times per day with a frequency determined by the sensor default time-step (can be changed with a configuration downlink).
- T2: contains useful information such as the sensor type, the firmware version, a long index. It is generated once per day.
- TT1: contains technical information (for technical feedback purposes) and is generated once per day.
- TT2: contains technical information (for technical feedback purposes) and is generated each hour for one day after startup.

After start time (t_start), the sequence of messages (for configuration with 15 minutes time-step) is the following one:

- t_start + 30 seconds: T2
- t_start + 1 minute: TT1
- t_start + 5 minutes: 010203
- t_start + 1 hour + 1 minute + 30 seconds: TT2
- t_start + 2 hours: T1
- t_start + 2 hours + 1 minute + 30 seconds: TT2
- t_start + 3 hours + 1 minute + 30 seconds: TT2
- t_start + 4 hours: T1
- etc.
- t_start + 24 hours: T1
- t_start + 24 hours + 30 seconds: T2
- t_start + 24 hours + 1 minute: TT1
- t_start + 26 hours: T1
- t_start + 28 hours: T1
- etc.

6.2 Data message (T1) structure

6.2.1 Introduction

The T1 data message transmits **1 index value** and **n increments**.

The **index value** is the cumulated number of optical detections since the start of the sensor. One optical detection can be one LED flash (in the case of an electronic electricity meter) or one disk rotation (in the case of an electromechanical electricity meter). In simple cases, 1 detection represents 1Wh. In other cases, a multiplication factor is related to the meter and 1 detection can represent xWh.

An **increment** is the index difference between a time -step and the next time-step.

T1 data messages are sent:

- every 80 minutes for sensors with the default 10-minutes time-step configuration
- every 2 hours for sensors with the default 15-minutes time-step configuration
- many other configurations with the help of a configuration downlink (or before shipping for special requests). Measurement time-step can be 5 minutes, 10 minutes, 15 minutes, 30 minutes, 60 minutes. Number of measurement values (number of increments) in each message can be any value. Almost any value, because there are restrictions related to max number of Bytes or radio duty-cycles.

There is a redundancy option that can be activated remotely with a downlink. When redundancy is activated, previous increments are repeated. For example, if the time-step is 15 minutes and there are 4 new increments before the message is sent, the previous 4 time-steps will also be sent. This way, no increment is lost, even if a message is lost (only when there are not several successive missing messages).

6.2.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

For example, in the case of 8 measurements in each message, each **increment** should be time stamped as follows: $t_i=t_{received} - ((8 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,7}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two measurements.

6.2.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22
Signification	Header	Time-step	Index				Incr. (t0)		Incr. (t1)		Incr. (t2)		Incr. (t3)		Incr. (t4)		Incr. (t5)		Incr. (t6)		Incr. (t7)	

Header

Byte	value (Hexa)	Signification
#1	0x69	FM432e ap version (adjustable parameters)

Time step

Byte	value (Hexa)	Signification
#2	0xXX	Number of minutes for each measurement

Index

Byte	value (Hexa)	Signification
#3	Byte3	Index = (hex2dec(**Byte3**) * 2^24) + (hex2dec(**Byte4**) * 2^16) + (hex2dec(**Byte5**) * 2^8) + hex2dec(**Byte6**)
#4	Byte4	
#5	Byte4	
#6	Byte6	

Incr. (ti)

Byte	value (Hexa)	Signification
#7	Byte7	Increment(t0) = (hex2dec(**Byte7**) * 2^8) + hex2dec(**Byte8**)
#8	Byte8	
#9	Byte9	Increment(t1) = (hex2dec(**Byte7**) * 2^8) + hex2dec(**Byte8**)
#10	Byte10	
…	…	…

6.2.4 Example

Given the following T1 message: 690F00006F920178017B0181018C01980196019C019F

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**	**#21**	**#22**
Signification	**Header***	**Time - step**	**Index**				**Incr. (t0)**		**Incr. (t1)**		**Incr. (t2)**		**Incr. (t3)**		**Incr. (t4)**		**Incr. (t5)**		**Incr. (t6)**		**Incr. (t7)**	
Hexa value	69	0F	00	00	6F	92	01	78	01	7B	01	81	01	8C	01	98	01	96	01	9C	01	9F
Decoded value		15	28562				376		379		385		396		408		406		412		415	

6.3 Service message (T2) structure

The T2 data message transmits technical information about the sensor (firmware, meter type, low battery indicator) and more info about the measurements (a longer index, the time-step, and a max power value). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16
Signif.	Header	Number of starts	NA	Synchro	Reserved	Optical head firmware	Meter type	Low battery	Index				Time-step	Number of values	Redundancy	Sensitivity

Header

Byte	value (Hexa)	Signification
#1	0x6A	FM432e ap version (adjustable parameters)

Number of starts

Byte	value (Hexa)	Signification
#2	0xXX	Number of starts (optical head starts again each time a parameter is changed) of the optical head since last powered-up

Synchro

Byte	value (Hexa)	Signification
#4	0x00	no synchronization
	0x01	synchronization request

Optical Head firmware

Byte	value (Hexa)	Signification
#6	0xXX	Exemple: 3C => 60 => firmware v6.0

Meter type

Byte	value (Hexa)	Signification
#7	0x00	Electromechanical meter (spinning disk)
	0x01	Electronic meter (flashing LED)

Low battery

Byte	value (Hexa)	Signification
#8	0x00	battery OK
	0x01	battery NOK (Battery voltage < 2.8 Volts)

Index

Byte	value (Hexa)	Signification
#9	Byte9	Index = (hex2dec(**Byte9**) * 2^24) + (hex2dec(**Byte10**) * 2^16) + (hex2dec(**Byte11**) * 2^8) + hex2dec(**Byte12**)
#10	Byte10	
#11	Byte11	
#12	Byte12	

Time step

Byte	value (Hexa)	Signification
#13	0xXX	Number of minutes for each measurement

Number of values

Byte	value (Hexa)	Signification
#14	0xXX	Number of measurement values in each T1 uplink

Redundancy

Byte	value (Hexa)	Signification
#15	0x00	No redundancy
	0x01	Redundancy

Sensitivity

Byte	value (Hexa)	Signification
#16	0x00	highest sensitivity
	0x01	high intermediate sensitivity
	0x02	low intermediate sensitivity
	0x03	lowest sensitivity

6.4 Technical message #1 (TT1) structure

Header (hexa): 12

Size: 19 Bytes

6.5 Technical message #2 (TT2) structure

Header (hexa): 13

Size: 11 Bytes

6.6 Javascript decoder

Decoder for the FM432e_ap product:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432e_ap-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

7. Decoding FM432e_nc_1mn messages

7.1 Types of messages

	Time-step
Message Type	1 minute
Data message (T1)	Every 20 minutes (20 x 1 minute)
Service message (T2)	Every 24 hours

FM432e_nc_1mn generates two kinds of messages:

- T1: contains twenty power values and the last index. It is generated every 20 minutes.
- T2: contains additional information such as the sensor type and the firmware version. It is generated once per day.

7.2 Data message (T1) structure

7.2.1 Introduction

The T1 data message transmits 1 **index** value (cumulated energy value) and 20 **power** values.

A power value is the average power over 1 minute. The average power calculation is based on the elapsed time between two consecutive optical detections and involves interpolation in order to create 1 minute values. One optical detection can be one LED flash (in the case of an electronic electricity meter) or one disk rotation (in the case of an electromechanical electricity meter). In simple cases, 1 detection represents 1Wh (Watt-hour) and the power values are in W (Watts). In other cases, a multiplication factor x is related to the meter and 1 detection can represent xWh (in this case the power values should be multiplied by this factor once received).

The T1 data message behaves as follows:

1. First message is sent 30 minutes after the sensor start-up.
2. Subsequent messages are sent periodically every 20 minutes.

Every T1 data message includes data related to a period of 20 minutes.

7.2.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$ *- delay*

Each **power** should be time stamped as follows: $t_i=t_{received}$ *- delay - (20 - i)*

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,19}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *delay is a delay of 10 minutes due to the power computation mechanism. It means that the message is sent 10 minutes after the time of the last 1 minute power. For example, if the message contains power values between 9:00am and 9:20am, the message will only be sent at 9:30am. This way, even*

if energy is low, there is time to wait for a late optical detection, necessary to compute average power, and attribute their share to the power values before sending the message.

7.2.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	...	#40	#41	#42	#43	#44	#45
Signification	Header*	Index				P(t0)		P(t1)		P(t2)		...	P(t17)		P(t18)		P(t19)	

7.2.4 Header

* Header value (Hexa)	Signification
5b	FM432e with 1-min time step

7.2.5 Index calculation

The index can be obtained this way:

Index = (**Byte_2** * 2^24) + (**Byte_3** * 2^16) + (**Byte_4** * 2^8) + **Byte_5**

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

7.2.6 Power values calculation

Power values are labelled (t0) to (t19).

The formulas for obtaining power values are:

Power(t0) = (Byte_6 * 2^8) + Byte_7
Power(t1) = (Byte_8 * 2^8) + Byte_9
Power(t2) = (Byte_10 * 2^8) + Byte_11
…

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

7.2.7 Example

Given the following T1 message:

5b0afdff00068f068f0649066a067e0682057a04ad049f04bd04c204c004c604bf04ae04a504a304b0049b04ac

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**...**	**#40**	**#41**	**#42**	**#43**	**#44**	**#45**
Signification	**Header***	**Index**				**P(t0)**		**P(t1)**		**P(t2)**		**...**	**P(t17)**		**P(t18)**		**P(t19)**	
Hexa value	5b	0a	fd	ff	00	06	8f	06	8f	06	49		04	b0	04	9b	04	ac
Decoded value		184418048				1679		1679		1609			1200		1179		1196	

7.3 Service message (T2) structure

The T2 data message transmits technical information about the sensor (firmware, meter type, low battery indicator) and more info about the measurements (a longer index, the time-step, and a max power value). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
Signification	Header (51)	Number of starts	Time synchronization (see below)	Not used	Optical head information (see below)	Index				Not used		Time step (see below)

Number of starts

Byte	**Signification**
#2	Number of starts since initial configuration (= number of times the sensor is unplugged from the radio module)

Time synchronization information

Byte	**Bit**	**Signification**
#3	1 - 7	Jitter (in seconds) set to zero for 1 minute data
	8	Synchro querying: not available for 1 minute data

Optical head information

Byte	**Bit**	**Signification**
#5	1 - 6	Firmware version
	7	Meter type: 0 = electromechanical meter 1 = electronic meter
	8	Low battery: 0 = battery OK 1 = battery NOK (Battery voltage < 2.8 Volts)

Index

Bytes	**Index calculation**
#6, #7, #8, #9	**Index** = (**Byte_#6** * 2^24) + (**Byte_#7** * 2^16) + (**Byte_#8** * 2^8) + **Byte_#9**

Time-step

Byte	**Value (hexa)**	**Signification**
#12	02	1-minute interval

7.4 Javascript decoder

Decoder for the FM432e_nc_1mn product:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432e_nc_1mn-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

8. Decoding FM432e_nc_10mn and FM432e_nc_15mn messages

8.1 Types of messages

Sensor version	10 minutes	15 minutes	1 hour
Data message (T1)	Every 80 minutes (8 x 10 minutes)	Every 2 hours (8 x 15 minutes)	Every 8 hours (8 x 1 hour)
Service message (T2)	Every 24 hours	Every 24 hours	Every 24 hours
Technical message 1 (TT1)	Every 24 hours	Every 24 hours	Every 24 hours
Technical message 2 (TT2)	Every hour for 1 day	Every hour for 1 day	Every hour for 1 day

FM432e generates four kinds of messages:

- T1: contains the consumption data and is generated several times per day with a frequency determined by the sensor time-step.
- T2: contains useful information such as the sensor type, the firmware version, a long index. It is generated once per day.
- TT1: contains technical information (for technical feedback purposes) and is generated once per day.
- TT2: contains technical information (for technical feedback purposes) and is generated each hour for one day after startup.

After start time (t_start), the sequence of messages (for version 15 minutes time-step) is the following one:

- t_start + 30 seconds: T2
- t_start + 1 minute: TT1
- t_start + 5 minutes: 010203
- t_start + 1 hour + 1 minute + 30 seconds: TT2
- t_start + 2 hours: T1
- t_start + 2 hours + 1 minute + 30 seconds: TT2
- t_start + 3 hours + 1 minute + 30 seconds: TT2
- t_start + 4 hours: T1
- etc.
- t_start + 24 hours: T1
- t_start + 24 hours + 30 seconds: T2
- t_start + 24 hours + 1 minute: TT1
- t_start + 26 hours: T1
- t_start + 28 hours: T1
- etc.

8.2 Data message (T1) structure

8.2.1 Introduction

The T1 data message transmits **1 index value** and **8 increments**.

The **index value** is the cumulated number of optical detections since the start of the sensor. One optical detection can be one LED flash (in the case of an electronic electricity meter) or one disk rotation (in the case of an electromechanical electricity meter). In simple cases, 1 detection represents 1Wh. In other cases, a multiplication factor is related to the meter and 1 detection can represent xWh.

An **increment** is the index difference between a time -step and the next time-step.

T1 data messages are sent:

- every 80 minutes for 10-minutes time-step sensors
- every 2 hours for 15-minutes time-step sensors
- every 8 hours for 1-hour time-step sensors

8.2.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

Each **increment** should be time stamped as follows: $t_i=t_{received} - ((8 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,7}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two measurements.

8.2.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20
Signification	Header*	Index			Incr. (t0)		Incr. (t1)		Incr. (t2)		Incr. (t3)		Incr. (t4)		Incr. (t5)		Incr. (t6)		Incr. (t7)	

8.2.4 Header

* Header values (Hexa)	Signification
20	10-min time step
21	15-min time step
22	1-hour time step

8.2.5 Index calculation

The index can be obtained this way:

Index = (**Byte_2** * 2^16) + (**Byte_3** * 2^8) + **Byte_4**

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

8.2.6 Increment calculation

Increment values are labelled (t0) to (t7) and represent successive increments.

The formulas for obtaining increment values are:

Increment(t0) = (Byte_5 * 2^8) + Byte_6
Increment(t1) = (Byte_7 * 2^8) + Byte_8
Increment(t2) = (Byte_9 * 2^8) + Byte_10
…

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

8.2.7 Example

Given the following T1 message: 21006f920178017b0181018c01980196019c019f

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**
Signification	**Header***	**Index**			**Incr. (t0)**		**Incr. (t1)**		**Incr. (t2)**		**Incr. (t3)**		**Incr. (t4)**		**Incr. (t5)**		**Incr. (t6)**		**Incr. (t7)**	
Hexa value	21	00	6F	92	01	78	01	7B	01	81	01	8C	01	98	01	96	01	9C	01	9F
Decoded value		28562			376		379		385		396		408		406		412		415	

8.3 Service message (T2) structure

The T2 data message transmits technical information about the sensor (firmware, meter type, low battery indicator) and more info about the measurements (a longer index, the time-step, and a max power value). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
Signification	Header (0E)	Number of startups	Time synchronization (see below)	Param. Id	Optical head information (see below)	Long index				Not used		Time step (see below)

Time synchronization information

Byte	**Bit**	**Signification**
#3	1 - 7	Jitter (in seconds)
	8	Synchro querying: 0 = no synchro querying 1 = synchro querying

Optical head information

Byte	**Bit**	**Signification**
#5	1 - 6	Firmware version
	7	Meter type: 0 = electromechanical meter 1 = electronic meter
	8	Low battery: 0 = battery OK 1 = battery NOK (Battery voltage < 2.8 Volts)

Time step

Byte	Value	Signification
#12	00	10-minutes time step
	03	15-minutes time step
	01	1-hour time step

8.3.1 Long Index calculation

The index can be obtained this way (make sure to first convert to decimal):

Index = (**Byte_6** * 2^24) + (**Byte_7** * 2^16) + (**Byte_8** * 2^8) + **Byte_9**

8.3.2 Max power value calculation

The max power in W can be obtained this way (make sure to first convert to decimal):

Pmax = (Byte_10 * 2^8) + Byte_11

8.4 Technical message #1 (TT1) structure

Header (hexa): 12

Size: 19 Bytes

8.5 Technical message #2 (TT2) structure

Header (hexa): 13

Size: 11 Bytes

8.6 Javascript decoder

Decoder for the FM432e_nc_10mn and FM432e_nc_15mn products:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432e_nc_10mn-15mn-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

9. Decoding FM432ir_ap messages

FM432ir works both with mME electricity meters with SML protocol and with electromechanical electricity meters (physical switch on the optical head to change meter type).

9.1 Types of messages

Sensor default configuration	15 minutes
Data message (T1)	Every 2 hours (8 x 15 minutes) Measurement and uplink frequencies can be remotely configured with a downlink
Service message (T2)	Every 24 hours

FM432ir generates two kinds of messages:

- T1: contains the consumption data and is generated several times per day with a frequency determined by the sensor default time-step (can be changed with a configuration downlink).
- T2: contains useful information such as the sensor type, the firmware version, a long index. It is generated once per day.

After start time (t_start), the sequence of messages (for configuration with 15 minutes time-step) is the following one:

- t_start + 30 seconds: T2
- t_start + 5 minutes: 010203
- t_start + 2 hours: T1
- t_start + 4 hours: T1
- etc.
- t_start + 24 hours: T1
- t_start + 24 hours + 30 seconds: T2
- t_start + 26 hours: T1
- t_start + 28 hours: T1
- etc.

9.2 Data message (T1) structure in the case of an infrared meter (mME)

9.2.1 Introduction

The T1 data message transmits **1 index value** and **n increments**, but in the case of the special configuration for self-consumption, there are **2 index values** and **2xn increments** (1 index and n increments for positive values, i.e. imported power, and 1 index and n increments for negative values, i.e exported power)

The **index value** is the cumulated energy since the start of the sensor.

An **increment** is the index difference between a time -step and the next time-step.

T1 data messages are sent:

- every 2 hours for sensors with the default 15-minutes time-step configuration
- many other configurations with the help of a configuration downlink (or before shipping for special requests). Measurement time-step can be 5 minutes, 10 minutes, 15 minutes, 30 minutes, 60 minutes. Number of measurement values (number of increments) in each message can be any value. Almost any value, because there are restrictions related to max number of Bytes or radio duty-cycles.

There is a redundancy option that can be activated remotely with a downlink. When redundancy is activated, previous increments are repeated. For example, if the time-step is 15 minutes and there are 4 new increments before the message is sent, the previous 4 time-steps will also be sent. This way, no increment is lost, even if a message is lost (only when there are not several successive missing messages).

9.2.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

For example, in the case of 8 measurements in each message, each **increment** should be time stamped as follows: $t_i=t_{received} - ((8 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,7}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two measurements.

9.2.3 Byte structure

Byte	#1	#2	#3	#4	#5	…	#10	#11	#12	#13	#14	#15	…	…	#24	#25	#26	#27
Signif.	Header	Time -step	Signed	Index					Incr. (t0)		Incr. (t1)		…		Incr. (t6)		Incr. (t7)	

With E-POS and E-NEG values in self-consumption cases:

Byte	#1	#2	#3	#4	…	#11	#12	#13	…	#26	#27	#28	…	#35	#36	#37	…	#50	#52
Signif.	Header	Time -step	Signed	Index E-POS			Incr. (t0)		…	Incr. (t7)		Index E-NEG			Incr. (t0)		…	Incr. (t7)	

Header

Byte	value (Hexa)	Signification
#1	0x71	FM432ir ap version (adjustable parameters) / E-SUM
	0x72	FM432ir ap version (adjustable parameters) / E-POS
	0x73	FM432ir ap version (adjustable parameters) / E-NEG
	0x74	FM432ir ap version (adjustable parameters) / E-POS and E-NEG

Just after starting up, the sensor looks for specific OBIS codes in the following order:
OBIS code 1.8.0 (E-POS): positive active energy (A+)
OBIS code 16.8.0 (E-SUM): sum active energy without reverse blockade (A+ - A-)
OBIS code 2.8.0 (E-NEG): negative active energy (A-)
The first OBIS code recognized defines what will be read.
In self-consumption cases with E-POS and E-NEG both needed, the option must be specifically set with help of a downlink (or pre-configured by Fludia if specified in the order)

Time step

Byte	value (Hexa)	Signification
#2	0xXX	Number of minutes for each measurement

Signed

Byte	value (Hexa)	Signification
#3	0x00	Unsigned (values can only be positive)
	0x01	Signed (values can be positive or negative)

Index

Byte	value (Hexa)	Signification (decoding)
#4	Byte4	Index = ((hex2dec(**Byte4**) * 2^56) + (hex2dec(**Byte5**) * 2^48) + (hex2dec(**Byte6**) * 2^40) + (hex2dec(**Byte7**) * 2^32) + (hex2dec(**Byte8**) * 2^24) + (hex2dec(**Byte9**) * 2^16) + (hex2dec(**Byte10**) * 2^8) + hex2dec(**Byte11**)) /10 If signed, decoding is a bit special, see javascript decoding example
#5	Byte5	
#6	Byte6	
#7	Byte7	
#8	Byte8	
#9	Byte9	
#10	Byte10	
#11	Byte11	

Incr. (ti)

Byte	value (Hexa)	Signification (decoding)
#12	Byte12	Increment(t0) = ((hex2dec(**Byte12**) * 2^8) + hex2dec(**Byte13**)) /10
#13	Byte13	
#14	Byte14	Increment(t1) = ((hex2dec(**Byte14**) * 2^8) + hex2dec(**Byte15**)) /10
#15	Byte15	
…	…	…

9.2.4 Example

Given the following T1 message: 72 0F 00 0000000000107900 0A0A 0A0B … 0E17 0C11

The decoded values should be:

Byte	**#1**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**...**	**#25**	**#26**	**#27**	**#28**
Signif.	**Header**	**Time step**	**Signed**	**Index**								**Incr.t0)**		**Incr.(t1)**		**...**	**Incr.(t13)**		**Incr.(t14)**	
Hexa value	72	0F	00	00	00	00	00	00	10	79	00	0A	0A	0A	0B		0E	17	0C	11
Decoded value		15	00	107955.2 Wh								257.0 Wh		257.1 Wh			360.7 Wh		308.9 Wh	

9.3 Data message (T1) structure in the case of an electromechanical meter

9.3.1 Introduction

The T1 data message transmits **1 index value** and **n increments**.

The **index value** is the cumulated energy since the start of the sensor.

An **increment** is the index difference between a time -step and the next time-step.

T1 data messages are sent:

- every 2 hours for sensors with the default 15-minutes time-step configuration
- many other configurations with the help of a configuration downlink (or before shipping for special requests). Measurement time-step can be 5 minutes, 10 minutes, 15 minutes, 30 minutes, 60 minutes. Number of measurement values (number of increments) in each message can be any value. Almost any value, because there are restrictions related to max number of Bytes or radio duty-cycles.

There is a redundancy option that can be activated remotely with a downlink. When redundancy is activated, previous increments are repeated. For example, if the time-step is 15 minutes and there are 4 new increments before the message is sent, the previous 4 time-steps will also be sent. This way, no increment is lost, even if a message is lost (only when there are not several successive missing messages).

9.3.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

For example, in the case of 8 measurements in each message, each **increment** should be time stamped as follows: $t_i=t_{received} - ((8 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,7}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two measurements.

9.3.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	…	…	#19	#20	#21	#22
Signif.	Header	Time -step	Index				Incr. (t0)		Incr. (t1)		…		Incr. (t6)		Incr. (t7)	

Header

Byte	value (Hexa)	Signification
#1	0x6F	FM432ir ap version (adjustable parameters) / electromechanical

Time step

Byte	value (Hexa)	Signification
#2	0xXX	Number of minutes for each measurement

Index

Byte	value (Hexa)	Signification (decoding)
#3	Byte3	Index = (hex2dec(**Byte3**) * 2^24) + (hex2dec(**Byte4**) * 2^16) + (hex2dec(**Byte5**) * 2^8) + hex2dec(**Byte6**)
#4	Byte4	
#5	Byte5	
#6	Byte6	

Incr. (ti)

Byte	value (Hexa)	Signification (decoding)
#7	Byte7	Increment(t0) = (hex2dec(**Byte7**) * 2^8) + hex2dec(**Byte8**)
#8	Byte8	
#9	Byte9	Increment(t1) = (hex2dec(**Byte9**) * 2^8) + hex2dec(**Byte10**)
#10	Byte10	
…	…	…

9.3.4 Example

Given the following T1 message

6F 0F 005A962B 0035 0B34 0B34 0A1F 00A1 007B 2206 1968

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**	**#21**	**#22**
Signification	**Header***	**Time step**	**Index**				**Incr. (t0)**		**Incr. (t1)**		**Incr. (t2)**		**Incr. (t3)**		**Incr. (t4)**		**Incr. (t5)**		**Incr. (t6)**		**Incr. (t7)**	
Hexa value	6F	0F	00	5A	96	2B	00	35	0B	34	0B	34	0A	1F	00	A1	00	7B	22	06	19	68
Decoded value		15	5936683				53		2868		2868		2591		161		123		8710		6504	

9.4 Service message (T2) structure in the case of an infrared meter (mME)

The T2 data message transmits technical information about the sensor (firmware version…) and more info about the measurements (scaler value, reminder of the current index…). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	…	#20
Signif.	Header		Time step	Number of values	Redundancy	Type of measure	Forced Type?	firmware version	Sensor sensitivity	Scaler E-POS	Scaler E-SUM	Scaler E-NEG	Index		

Header

Byte	value (Hexa)	Signification
#1	0xF03A	FM432ir ap version (adjustable parameters) T2 in infrared mode
#2		

Time step

Byte	value (Hexa)	Signification
#3	0xXX	Number of minutes for each measurement

Number of values

Byte	value (Hexa)	Signification
#4	0xXX	Number of measurement values in each T1 uplink

Redundancy

Byte	value (Hexa)	Signification
#5	0x00	No redundancy
	0x01	Redundancy

Type of measure

Byte	Value	Signification
#6	00	E-POS values (OBIS code 1.8.0)
	01	E-SUM values (OBIS code 16.8.0)
	02	E-NEG values (OBIS code 2.8.0)
	03	E-POS and E-NEG (OBIS codes 1.8.0 and 2.8.0)

Forced Type?

Byte	Value	Signification
#7	00	Type of measure **hasn't been forced** via a downlink since starting up
	01	Type of measure **has been forced** via a downlink since starting up

Firmware version

Byte	Value	Signification
#8	0A	Optical head firmware version 1.0 (0A=>10=>1.0 / 0B=>11=>1.1…)

Sensor sensitivity

Byte	Value	Signification
#9	00	highest sensitivity
	01	high intermediate sensitivity
	02	low intermediate sensitivity
	03	lowest sensitivity

Scaler E-POS

Byte	Value (signed hexa)	Signification
#10	FF / 01 / 02 / 03 / 7F	Multiplication factor that has been applied to index and increment values in case the type is E-POS (FF: 10^-1 ; 01: 10^1 ; 02: 10^2 ; 03: 10^3 ; 7F: OBIS code 1.8.0 not found)

Scaler E-SUM

Byte	Value (signed hexa)	Signification
#11	FF / 01 / 02 / 03 / 7F	Multiplication factor that has been applied to index and increment values in case the type is E-SUM (FF: 10^-1 ; 01: 10^1 ; 02: 10^2 ; 03: 10^3; 7F: OBIS code 16.8.0 not found)

Scaler E-NEG

Byte	Value (signed hexa)	Signification
#12	FF / 01 / 02 / 03 / 7F	Multiplication factor that has been applied to index and increment values in case the type is E-NEG (FF: 10^-1 ; 01: 10^1 ; 02: 10^2 ; 03: 10^3; 7F: OBIS code 2.8.0 not found)

Index

Byte	value (Hexa)	Signification (decoding)
#13	Byte13	Index = ((hex2dec(**Byte13**) * 2^56) + (hex2dec(**Byte14**) * 2^48) + (hex2dec(**Byte15**) * 2^40) + (hex2dec(**Byte16**) * 2^32) + (hex2dec(**Byte17**) * 2^24) + (hex2dec(**Byte18**) * 2^16) + (hex2dec(**Byte19**) * 2^8) + hex2dec(**Byte20**)) /10 If signed,first bit gives the sign and decoding is a bit special, see javascript decoding example
#14	Byte14	
#15	Byte15	
#16	Byte16	
#17	Byte17	
#18	Byte18	
#19	Byte19	
#20	Byte20	

9.5 Service message (T2) structure in the case of an electromechanical meter

The T2 data message transmits technical information about the sensor (firmware, meter type, low battery indicator..). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16
Signif.	Header	Number of starts	NA	Synchro	Reserved	Optical head firmware	Meter type	Low battery	Index				Time-step	Number of values	Redundancy	NA

Header

Byte	value (Hexa)	Signification
#1	0x70	FM432ir ap version (adjustable parameters) / electromechanical

Number of starts

Byte	value (Hexa)	Signification
#2	0xXX	Number of starts (optical head starts again each time a parameter is changed) of the optical head since last powered-up

Synchro

Byte	value (Hexa)	Signification
#4	0x00	no synchronization
	0x01	synchronization request

Optical Head firmware

Byte	value (Hexa)	Signification
#6	0xXX	Exemple: 15 => 21 => firmware v2.1

Meter type

Byte	value (Hexa)	Signification
#7	0x00	Electromechanical meter (spinning disk)
	0x01	Infrared meter (mME with SML protocol)

Low battery

Byte	value (Hexa)	Signification
#8	0x00	battery OK
	0x01	battery NOK (Battery voltage < 2.8 Volts)

Index

Byte	value (Hexa)	Signification
#9	Byte9	Index = (hex2dec(**Byte9**) * 2^24) + (hex2dec(**Byte10**) * 2^16) + (hex2dec(**Byte11**) * 2^8) + hex2dec(**Byte12**)
#10	Byte10	
#11	Byte11	
#12	Byte12	

Time step

Byte	value (Hexa)	Signification
#13	0xXX	Number of minutes for each measurement

Number of values

Byte	value (Hexa)	Signification
#14	0xXX	Number of measurement values in each T1 uplink

Redundancy

Byte	value (Hexa)	Signification
#15	0x00	No redundancy
	0x01	Redundancy

9.6 Javascript decoder

Decoder for the FM432ir_ap product:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432ir_ap-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

10. Decoding FM432ir_nc_1mn messages

10.1 Types of messages

	Time-step
Message Type	1 minute
Data message (T1)	Every 15 minutes (15 x 1 minute)
Service message (T2)	Every 24 hours

FM432ir_nc_1mn generates two kinds of messages:

- T1: contains 15 or 20 successive values (depending on the type of meter) and the last index. It is generated every 15 or 20 minutes. The successive values are average power values in the case of an electromechanical meter. They are index increments in the case of an infrared meter (typically an mME meter).
- T2: contains additional technical information. It is generated once per day.

10.2 Data message (T1) structure in the case of an infrared meter (mME)

10.2.1 Introduction

The T1 data message transmits 1 **index** value (cumulated energy value) and 15 **increments**.

The **index value** is the cumulated energy since the start of the optical head.

An **increment** is the index difference between a time -step and the next time-step.

T1 data messages are sent every 15 minutes.

10.2.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

Each **increment** should be time stamped as follows: $t_i=t_{received} - ((15 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,14}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two measurements (here, 1 minute).

10.2.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	...	#11	#12	#13	#14	#15	#16	#17	#18	...	#39	#40	#41	#42
Signification	Header		Time-step	Sign	Index					Incr.(t0)		Incr.(t1)		Incr.(t2)		...	Incr(t13)		Incr(t14)	

10.2.4 Header

Header value (Hexa)	Signification
0xF02E	mME meter delivering E-SUM values (OBIS code 16.8.0 available)
0xF02F	mME meter delivering E-POS values (OBIS code 16.8.0 not available, but 1.8.0 available
0xF030	mME meter delivering E-NEG values (OBIS codes 16.8.0 and 1.8.0 not available, but 2.8.0 available)

Just after starting up, the sensor looks for specific OBIS codes in the following order:

- OBIS code 1.8.0 (E-POS): positive active energy (A+)
- OBIS code 16.8.0 (E-SUM): sum active energy without reverse blockade (A+ - A-)
- OBIS code 2.8.0 (E-NEG): negative active energy (A-)

10.2.5 Time-step

For FM432ir_nc_1mn product reference, Time-step is set to 1.

10.2.6 Sign

Indicates if the increment values are signed or unsigned.

Byte	Value	Signification
#4	00	Increment values are unsigned
	01	Increment values are signed

10.2.7 Index calculation

The index in Wh can be obtained this way:

Index = ((**Byte_4** * 2^56) + (**Byte_5** * 2^48) + (**Byte_6** * 2^40) + (**Byte_7** * 2^32) + (**Byte_8** * 2^24) + (**Byte_9** * 2^16) + (**Byte_10** * 2^8) + **Byte_11)** / 10

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

10.2.8 Increment values calculation

Increment values are labelled (t0) to (t19) and represent successive increments.

The formulas for obtaining increment values in Wh are:

Increment(t0) = ((Byte_12 * 2^8) + Byte_13) / 10
Increment(t1) = ((Byte_14 * 2^8) + Byte_15) / 10
Increment(t2) = ((Byte_16 * 2^8) + Byte_17) / 10
…

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

Special values 0xFFFB, 0xFFFC, 0xFFFD, 0xFFFE and 0xFFFF are error codes to be communicated to Fludia for support.

10.2.9 Example

Given the following T1 message: F02F 01 00 0000000000107900 0A0A 0A0B … 0E17 0C11

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**...**	**#39**	**#40**	**#41**	**#42**
Signification	**Header***		**Time step**	**Sign**	**Index**								**Incr.t0)**		**Incr.(t1)**		**...**	**Incr.(t13)**		**Incr.(t14)**	
Hexa value	F02F		01	00	00	00	00	00	00	10	79	00	0A	0A	0A	0B		0E	17	0C	11
Decoded value			01	00	107955.2 Wh								257.0 Wh		257.1 Wh			360.7 Wh		308.9 Wh	

10.3 Data message (T1) structure in the case of an electromechanical meter

10.3.1 Introduction

The T1 data message transmits 1 **index** value (cumulated energy value) and 20 **power** values.

A power value is the average power over 1 minute. The average power calculation is based on the elapsed time between two consecutive optical detections and involves interpolation in order to create 1 minute values. One optical detection occurs for each disk rotation, when the black mark (on the spinning disk) passes in front of the sensor. A multiplication factor (ratio or constant of the meter) is generally related to the meter and 1 detection can represent x Wh. In this case, the power values should be multiplied by this factor once received.

The T1 data message behaves as follows:

1. First message is sent 30 minutes after the sensor start-up.
2. Subsequent messages are sent periodically every 15 minutes.

Every T1 data message includes data related to a period of 15 minutes.

10.3.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

Each **power** should be time stamped as follows: $t_i=t_{received} - delay - ((20 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,19}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *delay is a delay of 10 minutes due to the power computation mechanism. It means that the message is sent 10 minutes after the time of the last 1 minute power. For example, if the message contains power values between 9:00am and 9:20am, the message will only be sent at 9:30am. This way, even if energy is low, there is time to wait for a late optical detection, necessary to compute average power, and attribute their share to the power values before sending the message.*
- *Time_step* is the time-step, i.e the interval between two measurements (here, 1 minute).

10.3.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	...	#40	#41	#42	#43	#44	#45
Signification	Header	Index				P(t0)		P(t1)		P(t2)		...	P(t17)		P(t18)		P(t19)	

10.3.4 Header

Header value (Hexa)	**Signification**
5b	1 min time step

10.3.5 Index calculation

The index can be obtained this way:

Index = (**Byte_2** * 2^24) + (**Byte_3** * 2^16) + (**Byte_4** * 2^8) + **Byte_5**

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

10.3.6 Power values calculation

Power values are labelled (t0) to (t19).

The formulas for obtaining power values are:

Power(t0) = (Byte_6 * 2^8) + Byte_7
Power(t1) = (Byte_8 * 2^8) + Byte_9
Power(t2) = (Byte_10 * 2^8) + Byte_11
…

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

10.3.7 Example

Given the following T1 message:

50 0afdff00 068f 068f 0649 … 04b0 049b 04ac

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**...**	**#40**	**#41**	**#42**	**#43**	**#44**	**#45**
Signification	**Header***	**Index**				**P(t0)**		**P(t1)**		**P(t2)**		**...**	**P(t17)**		**P(t18)**		**P(t19)**	
Hexa value	50	0A	FD	FF	00	06	8F	06	8F	06	49		04	B0	04	9B	04	AC
Decoded value		184418048				1679		1679		1609			1200		1179		1196	

10.4 Service message (T2) structure in the case of an infrared meter (mME)

The T2 data message transmits technical information about the sensor (firmware version…) and more info about the measurements (scaler value, reminder of the current index…). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	…	#18
Signification	Header (F02A)		Time step	Type of measure	Forced Type?	firmware version	Sensor sensitivity	Scaler E-POS	Scaler E-SUM	Scaler E-NEG	Index		

Time-step

Byte	**Value**	**Signification**
#3	01	1-minute time step

Type of measure

Byte	**Value**	**Signification**
#4	00	E-POS values (OBIS code 1.8.0)
	01	E-SUM values (OBIS code 16.8.0)
	02	E-NEG values (OBIS code 2.8.0)

Forced Type?

Byte	Value	Signification
#5	00	Type of measure **hasn't been forced** via a downlink since starting up
	01	Type of measure **has been forced** via a downlink since starting up

Firmware version

Byte	Value	Signification
#6	0A	Optical head firmware version 1.0 (0A=>10=>1.0 / 0B=>11=>1.1…)

Sensor sensitivity

Byte	Value	Signification
#7	00	highest sensitivity
	01	high intermediate sensitivity
	02	low intermediate sensitivity
	03	lowest sensitivity

Scaler E-POS

Byte	Value (signed hexa)	Signification
#8	FF / 01 / 02 / 03 / 7F	Multiplication factor that has been applied to index and increment values in case the type is E-POS (FF: 10^-1 ; 01: 10^1 ; 02: 10^2 ; 03: 10^3 ; 7F: OBIS code 1.8.0 not found)

Scaler E-SUM

Byte	Value (signed hexa)	Signification
#9	FF / 01 / 02 / 03 / 7F	Multiplication factor that has been applied to index and increment values in case the type is E-SUM (FF: 10^-1 ; 01: 10^1 ; 02: 10^2 ; 03: 10^3; 7F: OBIS code 16.8.0 not found)

Scaler E-NEG

Byte	Value (signed hexa)	Signification
#10	FF / 01 / 02 / 03 / 7F	Multiplication factor that has been applied to index and increment values in case the type is E-NEG (FF: 10^-1 ; 01: 10^1 ; 02: 10^2 ; 03: 10^3; 7F: OBIS code 2.8.0 not found)

Index

Bytes	Index calculation
#11…. #18	**Index** = ((**Byte_11** * 2^56) + (**Byte_12** * 2^48) + (**Byte_13** * 2^40) + (**Byte_14** * 2^32) + (**Byte_15** * 2^24) + (**Byte_16** * 2^16) + (**Byte_17** * 2^8) + **Byte_18)** **/ 10**

10.5 Service message (T2) structure in the case of an electromechanical meter

The T2 data message transmits technical information about the sensor (firmware, low battery indicator) and reminder of key values (index and time-step). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
Signification	Header (4B)	Number of starts	Time synchronization	Not used	Optical head information	Index				Not used		Time step

Number of starts

Byte	Signification
#2	Number of starts since initial configuration (= number of times the sensor has been unplugged from the radio module)

Time synchronization information

Byte	Bit	Signification
#3	1 - 7	Jitter (in seconds)
	8	Synchro querying: 0 = no synchro querying 1 = synchro querying

Optical head information

Byte	Bit	Signification
#5	1 - 6	Firmware version
	7	Meter type: 0 = electromechanical meter
	8	Low battery: 0 = battery OK 1 = battery NOK (Battery voltage < 2.8 Volts)

Index

Bytes	Index calculation
#6, #7, #8, #9	**Index** = (**Byte_#6** * 2^24) + (**Byte_#7** * 2^16) + (**Byte_#8** * 2^8) + **Byte_#9**

Time-step

Byte	Value (hexa)	Signification
#12	02	1 minute time step

10.6 Javascript decoder

Decoder for the FM432ir_nc_1mn product:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432ir_nc_1mn-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

11. Decoding FM432ir_nc_15mn messages

11.1 Types of messages

	Time-step
Message Type	15 minutes
Data message (T1)	Every 2 hours (8 x 15 minutes)
Service message (T2)	Every 24 hours

FM432ir_nc_15mn generates two kinds of messages:

- T1: contains eight successive values and the last index. It is generated every 2 hours. The successive values are index increments.
- T2: contains additional technical information. It is generated once per day.

11.2 Data message (T1) structure in the case of an infrared meter (mME)

11.2.1 Introduction

The T1 data message transmits 1 **index value** (cumulated energy value) and 8 **increments**.

The **index value** is the cumulated energy since the start of the optical head.

An **increment** is the index difference between a time -step and the next time-step.

T1 data messages are sent every 2 hours for 15-minutes time-step sensors.

11.2.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

Each **increment** should be time stamped as follows: $t_i=t_{received} - ((8 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,7}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two measurements.

11.2.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	...	#11	#12	#13	#14	#15	#16	#17	#18	...	#25	#26	#27	#28
Signification	Header		Time-step	Sign	Index					Incr.(t0)		Incr.(t1)		Incr.(t2)		...	Incr.(t6)		Incr.(t7)	

11.2.4 Header

Header value (Hexa)	Signification
0xF02E	mME meter delivering E-SUM values (OBIS code 16.8.0 available)
0xF02F	mME meter delivering E-POS values (OBIS code 16.8.0 not available, but 1.8.0 available
0xF030	mME meter delivering E-NEG values (OBIS code 16.8.0 and 1.8.0 not available, but 2.8.0 available

Just after starting up, the sensor looks for specific OBIS codes in the following order:

- OBIS code 1.8.0 (E-POS): positive active energy (A+)
- OBIS code 16.8.0 (E-SUM): sum active energy without reverse blockade (A+ - A-)
- OBIS code 2.8.0 (E-NEG): negative active energy (A-)

11.2.5 Time-step

For FM432ir_nc_15mn product reference, Time-step is set to 0F.

11.2.6 Sign

Indicates if the index and increment values are signed or unsigned.

Byte	Value	Signification
#4	00	Index and Increment values are unsigned
	01	Index and Increment values are signed

11.2.7 Index calculation

The index in Wh can be obtained this way:

Index = ((**Byte_4** * 2^56) + (**Byte_5** * 2^48) + (**Byte_6** * 2^40) + (**Byte_7** * 2^32) + (**Byte_8** * 2^24) + (**Byte_9** * 2^16) + (**Byte_10** * 2^8) + **Byte_11)** **/ 10**

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

11.2.8 Increment value calculation

Increment values are labelled (t0) to (t7) and represent successive increments.

The formulas for obtaining increment values in Wh are:

Increment(t0) = ((Byte_12 * 2^8) + Byte_13) **/ 10**
Increment(t1) = ((Byte_14 * 2^8) + Byte_15) **/ 10**
Increment(t2) = ((Byte_16 * 2^8) + Byte_17) **/ 10**
…

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

Special values 0xFFFB, 0xFFFC, 0xFFFD, 0xFFFE and 0xFFFF are error codes to be communicated to Fludia for support.

11.2.9 Example

Given the following T1 message for a 15 minutes time-step sensor:

F02F 0F 00 0000000000107900 0A0A 0A0B … 0E17 0C11

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**...**	**#25**	**#26**	**#27**	**#28**
Signification	**Header***		**Time step**	**Sign**	**Index**								**Incr.t0)**		**Incr.(t1)**		**...**	**Incr.(t13)**		**Incr.(t14)**	
Hexa value	F02F		0F	00	00	00	00	00	00	10	79	00	0A	0A	0A	0B		0E	17	0C	11
Decoded value			15	00	107955.2 Wh								257.0 Wh		257.1 Wh			360.7 Wh		308.9 Wh	

11.3 Data message (T1) structure in the case of an electromechanical meter

11.3.1 Introduction

The T1 data message transmits **1 index value** and **8 increments**.

The **index value** is the cumulated number of optical detections since the start of the sensor. One optical detection occurs for each disk rotation, when the black mark (on the spinning disk) passes in front of the sensor.

Important: a multiplication factor (ratio or constant of the meter) is generally related to the meter and 1 detection can represent xWh.

An **increment** is the index difference between a time -step and the next time-step.

T1 data messages are sent every 2 hours for 15-minutes time-step sensors.

11.3.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

Each **increment** should be time stamped as follows: $t_i=t_{received} - ((8 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,7}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two measurements.

11.3.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21
Signification	Header*	Index				Incr. (t0)		Incr. (t1)		Incr. (t2)		Incr. (t3)		Incr. (t4)		Incr. (t5)		Incr. (t6)		Incr. (t7)	

11.3.4 Header

* Header values (Hexa)	Signification
49	15-min time step

11.3.5 Index calculation

The index can be obtained this way:

Index = (**Byte_2** * 2^24) + (**Byte_3** * 2^16) + (**Byte_4** * 2^8) + **Byte_5**

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

11.3.6 Increment calculation

Increment values are labelled (t0) to (t7) and represent successive increments.

The formulas for obtaining increment values are:

Increment(t0) = (Byte_6* 2^8) + Byte_7
Increment(t1) = (Byte_8 * 2^8) + Byte_9
Increment(t2) = (Byte_10 * 2^8) + Byte_11
…

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

11.3.7 Example

Given the following T1 message for a 15 minutes time-step sensor:

49 005A962B 0035 0B34 0B34 0A1F 00A1 007B 2206 1968

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**	**#21**
Signification	**Header***	**Index**				**Incr. (t0)**		**Incr. (t1)**		**Incr. (t2)**		**Incr. (t3)**		**Incr. (t4)**		**Incr. (t5)**		**Incr. (t6)**		**Incr. (t7)**	
Hexa value	49	00	5A	96	2B	00	35	0B	34	0B	34	0A	1F	00	A1	00	7B	22	06	19	68
Decoded value		5936683				53		2868		2868		2591		161		123		8710		6504	

11.4 Service message (T2) structure in the case of an infrared meter (mME)

The T2 data message transmits technical information about the sensor (firmware version…) and more info about the measurements (scaler value, reminder of the current index…). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	…	#18
Signification	Header (F02A)		Time step	Type of measure	Forced Type?	firmware version	Sensor sensitivity	Scaler E-POS	Scaler E-SUM	Scaler E-NEG	Index		

Time-step

Byte	**Value**	**Signification**
#3	0F	15 minutes time step

Type of measure

Byte	**Value**	**Signification**
#4	00	E-POS values (OBIS code 1.8.0)
	01	E-SUM values (OBIS code 16.8.0)
	02	E-NEG values (OBIS code 2.8.0)

Forced Type?

Byte	Value	Signification
#5	00	Type of measure **hasn't been forced** via a downlink since starting up
	01	Type of measure **has been forced** via a downlink since starting up

Firmware version

Byte	Value	Signification
#6	9	Optical head firmware version

Sensor sensitivity

Byte	Value	Signification
#7	00	highest sensitivity
	01	high intermediate sensitivity
	02	low intermediate sensitivity
	03	lowest sensitivity

Scaler E-POS

Byte	Value (signed hexa)	Signification
#8	FF / 01 / 02 / 03 / 7F	Multiplication factor that has been applied to index and increment values in case the type is E-POS (FF: 10^-1 ; 01: 10^1 ; 02: 10^2 ; 03: 10^3 ; 7F: OBIS code 1.8.0 not found)

Scaler E-SUM

Byte	Value (signed hexa)	Signification
#9	FF / 01 / 02 / 03 / 7F	Multiplication factor that has been applied to index and increment values in case the type is E-SUM (FF: 10^-1 ; 01: 10^1 ; 02: 10^2 ; 03: 10^3; 7F: OBIS code 16.8.0 not found)

Scaler E-NEG

Byte	Value (signed hexa)	Signification
#10	FF / 01 / 02 / 03 / 7F	Multiplication factor that has been applied to index and increment values in case the type is E-NEG (FF: 10^-1 ; 01: 10^1 ; 02: 10^2 ; 03: 10^3; 7F: OBIS code 2.8.0 not found)

Index

Bytes	Index calculation in Wh
#11…. #18	**Index** = ((**Byte_11** * 2^56) + (**Byte_12** * 2^48) + (**Byte_13** * 2^40) + (**Byte_14** * 2^32) + (**Byte_15** * 2^24) + (**Byte_16** * 2^16) + (**Byte_17** * 2^8) + **Byte_18)** / 10

11.5 Service message (T2) structure in the case of an electromechanical meter

The T2 data message transmits technical information about the sensor (firmware, low battery indicator) and reminder of key values (index and time-step). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
Signification	Header (4B)	Number of starts	Time synchronization	Not used	Optical head information	Long index				Not used		Time step (see below)

Number of starts

Byte	Signification
#2	Number of starts since initial configuration (= number of times the sensor has been unplugged from the radio module)

Time synchronization information

Byte	Bit	Signification
#3	1 - 7	Jitter (in seconds)
	8	Synchro querying: 0 = no synchro querying 1 = synchro querying

Optical head information

Byte	Bit	Signification
#5	1 - 6	Firmware version
	7	Meter type: 0 = electromechanical meter
	8	Low battery: 0 = battery OK 1 = battery NOK (Battery voltage < 2.8 Volts)

Index

Bytes	Index calculation
#6, #7, #8, #9	**Index** = (**Byte_#6** * 2^24) + (**Byte_#7** * 2^16) + (**Byte_#8** * 2^8) + **Byte_#9**

Time step

Byte	Value	Signification
#12	03	15 minutes time step

11.6 Javascript decoder

Decoder for the FM432ir_nc_15mn product:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432ir_nc_15mn-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

12. Decoding FM432g_ap messages

12.1 Types of messages

Sensor default configuration	10 minutes	15 minutes
Data message (T1)	Every 80 minutes (8 x 10 minutes) Measurement and uplink frequencies can be remotely configured with a downlink	Every 2 hours (8 x 15 minutes) Measurement and uplink frequencies can be remotely configured with a downlink
Service message (T2)	Every 24 hours	Every 24 hours
Technical message 1 (TT1)	Every 24 hours	Every 24 hours

FM432g generates three kinds of messages:

- T1: contains the consumption data and is generated several times per day with a frequency determined by the sensor default time-step (can be changed with a configuration downlink).
- T2: contains useful information such as the firmware version. It is generated once per day.
- TT1: contains technical information (for technical feedback purposes) and is generated once per day..

After start time (t_start), the sequence of messages (for configuration with 15 minutes time-step) is the following one:

- t_start + 30 seconds: T2
- t_start + 1 minute: TT1
- t_start + 5 minutes: 010203
- t_start + 2 hours: T1
- t_start + 4 hours: T1
- etc.
- t_start + 24 hours: T1
- t_start + 24 hours + 30 seconds: T2
- t_start + 24 hours + 1 minute: TT1
- t_start + 26 hours: T1
- t_start + 28 hours: T1
- etc.

12.2 Data message (T1) structure

12.2.1 Introduction

The T1 data message transmits **1 index value** and **n increments**.

The **index value** is the cumulated number of optical detections since the start of the sensor. One optical detection occurs when the digit changes in front of the optical sensor. Since the sensor is positioned in front of the second digit from the right, it counts 1 when 10 dm3 are consumed (unless the meter has a different unit than dm3 for the last digit, which is rare). Therefore, the sensor multiplies the number of detections by ten, so that the values are in dm3, which is usually more convenient.

An **increment** is the index difference between a time -step and the next time-step.

T1 data messages are sent:

- every 80 minutes for sensors with the default 10-minutes time-step configuration
- every 2 hours for sensors with the default 15-minutes time-step configuration
- many other configurations with the help of a configuration downlink (or before shipping for special requests). Measurement time-step can be 5 minutes, 10 minutes, 15 minutes, 30 minutes, 60 minutes. Number of measurement values (number of increments) in each message can be any value. Almost any value, because there are restrictions related to max number of Bytes or radio duty-cycles.

There is a redundancy option that can be activated remotely with a downlink. When redundancy is activated, previous increments are repeated. For example, if the time-step is 15 minutes and there are 4 new increments before the message is sent, the previous 4 time-steps will also be sent. This way, no increment is lost, even if a message is lost (only when there are not several successive missing messages).

12.2.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

For example, in the case of 8 measurements in each message, each **increment** should be time stamped as follows: $t_i=t_{received} - ((8 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,7}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two measurements.

12.2.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22
Signification	Header	Time-step	Index				Incr. (t0)		Incr. (t1)		Incr. (t2)		Incr. (t3)		Incr. (t4)		Incr. (t5)		Incr. (t6)		Incr. (t7)	

Header

Byte	value (Hexa)	Signification
#1	0x6D	FM432g ap version (adjustable parameters)

Time step

Byte	value (Hexa)	Signification
#2	0xXX	Number of minutes for each measurement

Index

Byte	value (Hexa)	Signification
#3	Byte3	Index = (hex2dec(**Byte3**) * 2^24) + (hex2dec(**Byte4**) * 2^16) + (hex2dec(**Byte5**) * 2^8) + hex2dec(**Byte6**)
#4	Byte4	
#5	Byte4	
#6	Byte6	

Incr. (ti)

Byte	value (Hexa)	Signification
#7	Byte7	Increment(t0) = (hex2dec(**Byte7**) * 2^8) + hex2dec(**Byte8**)
#8	Byte8	
#9	Byte9	Increment(t1) = (hex2dec(**Byte7**) * 2^8) + hex2dec(**Byte8**)
#10	Byte10	
…	…	…

12.2.4 Example

Given the following T1 message: 690F00006F920178017B0181018C01980196019C019F

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**	**#21**	**#22**
Signification	**Header***	**Time - step**	**Index**				**Incr. (t0)**		**Incr. (t1)**		**Incr. (t2)**		**Incr. (t3)**		**Incr. (t4)**		**Incr. (t5)**		**Incr. (t6)**		**Incr. (t7)**	
Hexa value	6D	0F	00	00	6F	92	01	78	01	7B	01	81	01	8C	01	98	01	96	01	9C	01	9F
Decoded value		15	28562				376		379		385		396		408		406		412		415	

12.3 Service message (T2) structure

The T2 data message transmits technical information about the sensor (firmware, meter type, low battery indicator) and more info about the measurements (a longer index, the time-step, and a max power value). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
Signif.	Header	Number of starts	NA	Synchro	NA	Optical head firmware	NA	NA	Index				Time-step	Number of values	Redundancy

Header

Byte	value (Hexa)	Signification
#1	0x6E	FM432g ap version (adjustable parameters)

Number of starts

Byte	value (Hexa)	Signification
#2	0xXX	Number of starts (optical head starts again each time a parameter is changed) of the optical head since last powered-up

Synchro

Byte	value (Hexa)	Signification
#4	0x00	no synchronization
	0x01	synchronization request

Optical Head firmware

Byte	value (Hexa)	Signification
#6	0xXX	Exemple: 0x3C => 60 => firmware v6.0

Index

Byte	value (Hexa)	Signification
#9	Byte9	Index = (hex2dec(**Byte9**) * 2^24) + (hex2dec(**Byte10**) * 2^16) + (hex2dec(**Byte11**) * 2^8) + hex2dec(**Byte12**)
#10	Byte10	
#11	Byte11	
#12	Byte12	

Time step

Byte	value (Hexa)	Signification
#13	0xXX	Number of minutes for each measurement

Number of values

Byte	value (Hexa)	Signification
#14	0xXX	Number of measurement values in each T1 uplink

Redundancy

Byte	value (Hexa)	Signification
#15	0x00	No redundancy
	0x01	Redundancy

12.4 Technical message (TT1) structure

Byte	Signification
#1	Header (hexa): 2E
#2-#22	Reserved technical information

12.5 Javascript decoder

Decoder for the FM432g_ap product:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432g_ap-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

13. Decoding FM432g_nc_10mn and FM432g_nc_15mn messages

13.1 Types of messages

Sensor version	10 minutes	15 minutes	1 hour
Data message (T1)	Every 80 minutes (8 x 10 minutes)	Every 2 hours (8 x 15 minutes)	Every 8 hours (8 x 1 hour)
Service message (T2)	Every 24 hours	Every 24 hours	Every 24 hours
Technical message 1 (TT1)	Every 24 hours	Every 24 hours	Every 24 hours

FM432g generates three kinds of messages:

- T1: contains the consumption data and is generated several times per day with a frequency determined by the sensor time-step.
- T2: contains useful information such as the sensor type, the firmware version, a long index. It is generated once per day.
- TT1: contains technical information (for technical feedback purposes) and is generated once per day.

13.2 Data message (T1) structure

13.2.1 Introduction

The T1 data message transmits **1 index value** and **8 increments**.

The **index value** is the cumulated number of optical detections since the start of the sensor. One optical detection occurs when the digit changes in front of the optical sensor. Since the sensor is positioned in front of the second digit from the right, it counts 1 when 10 dm3 are consumed (unless the meter has a different unit than dm3 for the last digit, which is rare). Therefore, the sensor multiplies the number of detections by ten, so that the values are in dm3, which is usually more convenient.

An **increment** is the index difference between a time -step and the next time-step.

T1 data messages are sent:

- every 80 minutes for 10-minutes time-step sensors
- every 2 hours for 15-minutes time-step sensors
- every 8 hours for 1-hour time-step sensors

13.2.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

Each **increment** should be time stamped as follows: $t_i=t_{received} - ((8 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,7}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two measurements.

13.2.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20
Signification	Header*	Index			Incr. (t0)		Incr. (t1)		Incr. (t2)		Incr. (t3)		Incr. (t4)		Incr. (t5)		Incr. (t6)		Incr. (t7)	

13.2.4 Header

* Header values (Hexa)	Signification
1D	10-min time step
1E	15-min time step
1F	1-hour time step

13.2.5 Index calculation

The index can be obtained this way:

Index = (**Byte_2** * 2^16) + (**Byte_3** * 2^8) + **Byte_4**

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

13.2.6 Increment calculation

Increment values are labelled (t0) to (t7) and represent successive increments.

The formulas for obtaining increment values are:

Increment(t0) = (Byte_5 * 2^8) + Byte_6
Increment(t1) = (Byte_7 * 2^8) + Byte_8
Increment(t2) = (Byte_9 * 2^8) + Byte_10
…

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

13.2.7 Example

Given the following T1 message: 1E006F900170017C0190018601AE019A018601B8

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**
Signification	**Header***	**Index**			**Incr. (t0)**		**Incr. (t1)**		**Incr. (t2)**		**Incr. (t3)**		**Incr. (t4)**		**Incr. (t5)**		**Incr. (t6)**		**Incr. (t7)**	
Hexa value	1E	00	6F	90	01	70	01	7C	01	90	01	86	01	AE	01	9A	01	86	01	B8
Decoded value		28560			370		380		400		390		430		410		390		440	

13.3 Service message (T2) structure

The T2 data message transmits information about the sensor (optical head firmware, number of starts) and more info about the measurements (a longer index, and a reminder of the time-step). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
Signification	Header (10)	Number of starts	Time synchronization (see below)	Param. Id	Optical head information (see below)	Long index				Not used		Time step (see below)

Number of starts

Byte	Signification
#2	Number of starts since initial configuration (= number of times the sensor is unplugged from the radio module)

Time synchronization information

Byte	Bit	Signification
#3	1 - 7	Jitter (in seconds)
	8	Synchro querying (needs specific implementation on the server side): 0 = no synchro querying 1 = synchro querying

Optical head information

Byte	Bit	Signification
#5	1 - 6	Firmware version
	7	Meter type: 1 = gas meter
	8	not used

Long index

Bytes	Index calculation
#6, #7, #8, #9	**Index** = (**Byte_#6** * 2^24) + (**Byte_#7** * 2^16) + (**Byte_#8** * 2^8) + **Byte_#9**

Time step

Byte	Value	Signification
#12	00	10-minutes time-step
	03	15-minutes time-step
	01	1-hour time-step

13.4 Technical message (TT1) structure

Byte	Signification
#1	Header (hexa): 2E
#2-#22	Reserved technical information

13.5 Javascript decoder

Decoder for the FM432g_nc_10mn and FM432g_nc_15mn products:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432g_nc_10mn-15mn-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

14. Decoding FM432p_ap messages (FM432p-a_ap and FM432p-n_ap)

14.1 Types of messages

Sensor default configuration	10 minutes	15 minutes
Data message (T1)	Every 80 minutes (8 x 10 minutes) Measurement and uplink frequencies can be remotely configured with a downlink	Every 2 hours (8 x 15 minutes) Measurement and uplink frequencies can be remotely configured with a downlink
Service message (T2)	Every 24 hours	Every 24 hours

FM432p generates two kinds of messages:

- T1: contains the consumption data and is generated several times per day with a frequency determined by the sensor default time-step (can be changed with a configuration downlink).
- T2: contains useful information such as the firmware version. It is generated once per day.

After start time (t_start), the sequence of messages (for configuration with 15 minutes time-step) is the following one:

- t_start + 30 seconds: T2
- t_start + 5 minutes: 010203
- t_start + 2 hours: T1
- t_start + 4 hours: T1
- etc.
- t_start + 24 hours: T1
- t_start + 24 hours + 30 seconds: T2
- t_start + 26 hours: T1
- t_start + 28 hours: T1
- etc.

14.2 Data message (T1) structure

14.2.1 Introduction

The T1 data message transmits **1 index value** and **n increments**.

The **index value** is the cumulated number of detected pulses since the start of the sensor.

An **increment** is the index difference between a time -step and the next time-step.

T1 data messages are sent:

- every 80 minutes for sensors with the default 10-minutes time-step configuration
- every 2 hours for sensors with the default 15-minutes time-step configuration
- many other configurations with the help of a configuration downlink (or before shipping for special requests). Measurement time-step can be 5 minutes, 10 minutes, 15 minutes, 30 minutes, 60 minutes. Number of measurement values (number of increments) in each message can be any value. Almost

any value, because there are restrictions related to max number of Bytes or radio duty-cycles.

There is a redundancy option that can be activated remotely with a downlink. When redundancy is activated, previous increments are repeated. For example, if the time-step is 15 minutes and there are 4 new increments before the message is sent, the previous 4 time-steps will also be sent. This way, no increment is lost, even if a message is lost (only when there are not several successive missing messages).

14.2.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

For example, in the case of 8 measurements in each message, each **increment** should be time stamped as follows: $t_i=t_{received} - ((8 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,7}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two measurements.

14.2.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22
Signification	Header	Time-step	Index				Incr. (t0)		Incr. (t1)		Incr. (t2)		Incr. (t3)		Incr. (t4)		Incr. (t5)		Incr. (t6)		Incr. (t7)	

Header

Byte	value (Hexa)	Signification
#1	0x6B	FM432p ap version (adjustable parameters) T1

Time step

Byte	value (Hexa)	Signification
#2	0xXX	Number of minutes for each measurement

Index

Byte	value (Hexa)	Signification
#3	Byte3	Index = (hex2dec(**Byte3**) * 2^24) + (hex2dec(**Byte4**) * 2^16) + (hex2dec(**Byte5**) * 2^8) + hex2dec(**Byte6**)
#4	Byte4	
#5	Byte4	
#6	Byte6	

Incr. (ti)

Byte	value (Hexa)	Signification
#7	Byte7	Increment(t0) = (hex2dec(**Byte7**) * 2^8) + hex2dec(**Byte8**)
#8	Byte8	
#9	Byte9	Increment(t1) = (hex2dec(**Byte7**) * 2^8) + hex2dec(**Byte8**)
#10	Byte10	
…	…	…

14.2.4 Example

Given the following T1 message: 6B 0F 00006F92 0178 017B 0181 018C 0198 0196 019C 019F

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**	**#21**	**#22**
Signification	**Header***	**Time - step**	**Index**				**Incr. (t0)**		**Incr. (t1)**		**Incr. (t2)**		**Incr. (t3)**		**Incr. (t4)**		**Incr. (t5)**		**Incr. (t6)**		**Incr. (t7)**	
Hexa value	6B	0F	00	00	6F	92	01	78	01	7B	01	81	01	8C	01	98	01	96	01	9C	01	9F
Decoded value		15	28562				376		379		385		396		408		406		412		415	

14.3 Service message (T2) structure

The T2 data message transmits technical information about the sensor (firmware, meter type, low battery indicator) and more info about the measurements (a longer index, the time-step, and a max power value). The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
Signification	Header	Firmware version			Index				Time-step	number of increment values	Redundancy

Header

Byte	**value (Hexa)**	**Signification**
#1	0x6C	FM432p ap version (adjustable parameters) T2

Firmware version

Bytes	**Value example (to be considered as ASCII)**	**Signification**
#2, #3, #4	33, 31, 36	33 is ASCII code for 3, 31 is ASCII code for 1, 36 is ASCII code for 6. So, in this example, version would be 3.1.6

Index

Byte	**value (Hexa)**	**Signification**
#5	Byte5	Index = (hex2dec(**Byte5**) * 2^24) + (hex2dec(**Byte6**) * 2^16) + (hex2dec(**Byte7**) * 2^8) + hex2dec(**Byte8**)
#6	Byte6	
#7	Byte7	
#8	Byte8	

Time step

Byte	**value (Hexa)**	**Signification**
#9	0xXX	Number of minutes for each measurement

Number of increment values

Byte	**value (Hexa)**	**Signification**
#10	0xXX	Number of measurement values in each T1 uplink

Redundancy

Byte	**value (Hexa)**	**Signification**
#11	0x00	No redundancy
	0x01	Redundancy

14.4 Javascript decoder

Decoder for the FM432p-a_ap and FM432p-n_ap products:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432p_ap-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

15. Decoding FM432p-(a | n)_nc_10mn and FM432p-(a | n)_nc_15mn

15.1 Types of messages

Sensor version	10 minutes	15 minutes	1 hour
Data message (T1)	Every 80 minutes (8 x 10 minutes)	Every 2 hours (8 x 15 minutes)	Every 8 hours (8 x 1 hour)
Service message (T2)	Every 24 hours	Every 24 hours	Every 24 hours

FM432p generates two kinds of messages:

- T1: contains the consumption data and is generated several times per day with a frequency determined by the time-step related to the product reference.
- T2: contains additional information such as the firmware version and a long index. It is generated once per day.

15.2 Data message (T1) structure

15.2.1 Introduction

The T1 data message transmits **1 index value** and **8 increments**.

The **index value** is the cumulated number of input pulses since the product start-up.

An **increment** is the index difference between a time-step and the next time-step.

T1 data messages are sent:

- every 80 minutes for a 10-minutes time-step product
- every 2 hours for a 15-minutes time-step product
- every 8 hours for a 1-hour time-step product

15.2.2 Timestamping

Index value should be time stamped as follows: $t=t_{received}$

Each **increment** should be time stamped as follows: $t_i=t_{received} - ((8 - i)*Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,7}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two measurements.

15.2.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20
Signification	Header*	Index			Incr. (t0)		Incr. (t1)		Incr. (t2)		Incr. (t3)		Incr. (t4)		Incr. (t5)		Incr. (t6)		Incr. (t7)	

15.2.4 Header

* Header values (Hexa)	Signification
2B	10-min time step
2C	15-min time step
2D	1-hour time step

15.2.5 Index calculation

The index can be obtained this way:

Index = (**Byte_2** * 2^16) + (**Byte_3** * 2^8) + **Byte_4**

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

15.2.6 Increment calculation

Increment values are labelled (t0) to (t7) and represent successive increments.

The formulas for obtaining increment values are:

Increment(t0) = (Byte_5 * 2^8) + Byte_6
Increment(t1) = (Byte_7 * 2^8) + Byte_8
Increment(t2) = (Byte_9 * 2^8) + Byte_10
…

Make sure to first convert from hexadecimal to decimal: **Byte_i** <= hex2dec(**Byte_i**)

15.2.7 Example

Given the following T1 message:

2C006f920178017b0181018c01980196019c019f

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**	**#11**	**#12**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**
Signification	**Header***	**Index**			**Incr. (t0)**		**Incr. (t1)**		**Incr. (t2)**		**Incr. (t3)**		**Incr. (t4)**		**Incr. (t5)**		**Incr. (t6)**		**Incr. (t7)**	
Hexa value	2C	00	6F	92	01	78	01	7B	01	81	01	8C	01	98	01	96	01	9C	01	9F
Decoded value		28562			376		379		385		396		408		406		412		415	

15.3 Service message (T2) structure

The T2 data message transmits additional information: firmware version, long index, number of increment values in each T1 message, time-step value. The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Signification	Header (29)	Firmware version			Long index				number of increment values in each T1 message	Time-step

Firmware version

Bytes	Value example (to be considered as ASCII)	Signification
#2, #3, #4	33, 31, 36	33 is ASCII code for 3, 31 is ASCII code for 1, 36 is ASCII code for 6. So, in this example the firmware version is 3.1.6

Long index

Bytes	Index calculation
#5, #6, #7, #8	**Index** = (**Byte_#5** * 2^24) + (**Byte_#6** * 2^16) + (**Byte_#7** * 2^8) + **Byte_#8**

Number of increment values in T1

Byte	Value (Hexa)	Signification
#9	08	8 increment values in each T1 message

Time step

Byte	Value (Hexa)	Signification
#12	0A	10-minutes time step
	0F	15-minutes time step
	3C	1-hour time step (60 minutes)

15.4 Javascript decoder

Decoder for the FM432p-a_nc_10mn, FM432p-a_nc_15mn, FM432p-n_nc_10mn and FM432p-n_nc_15mn product references:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432p_nc_10mn-15mn-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

16. Decoding FM432t_nc_1mn

16.1 Types of messages

	Time-step
Message Type	1 minute
Data message (T1)	Every 20 minutes (20 x 1 minute)
Service message (T2)	Every 24 hours

FM432t_1mn generates two kinds of messages:

- T1: contains twenty successive values. It is generated every 20 minutes. The successive values are temperatures.
- T2: contains additional technical information. It is generated once per day.

All hexadecimal values are unsigned, except Temperature values (they are signed).

16.2 Data message (T1) structure

16.2.1 Introduction

The T1 data message transmits **20 Temperature** values.

T1 data messages are sent every 20 minutes.

16.2.2 Timestamping

Each **Temperature** should be time stamped as follows: $t_i = t_{received} - (20 - i)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,19}.
- $t_{received}$ is the timestamp when T1 data message is received.

16.2.3 Byte structure

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**...**	**#37**	**#38**	**#39**	**#40**	**#41**	**#42**
Signification	**Header**	**Time-step**	**Temp(t0)**		**Temp(t1)**		**Temp(t2)**		**...**	**Temp(t17)**		**Temp(t18)**		**Temp(t19)**	

16.2.4 Header

Header value (Hexa)	**Signification**
57	FM432t with 1-min time step

16.2.5 Time-step

Time-step (Hexa)	Signification
01	1 minute

16.2.6 Temperature values calculation

Temperature values are labelled (t0) to (t19) and represent successive increments.

The formulas for obtaining increment values are:

Temp(t0) = ((Byte_3 * 2^8) + Byte_4)/100
Temp(t1) = ((Byte_5 * 2^8) + Byte_6)/100
Temp(t2) = ((Byte_7 * 2^8) + Byte_8)/100
…

Make sure to first convert from hexadecimal (signed) to decimal: **Byte_i** <= hex2dec(**Byte_i**)

16.2.7 Example

Given the following T1 message:

57 01 06f5 0708 0714 … 0484 046b 044c

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**...**	**#37**	**#38**	**#39**	**#40**	**#41**	**#42**
Signification	**Header***	**Time-step**	**Temp(t0)**		**Temp(t1)**		**Temp(t2)**		**...**	**Temp(t17)**		**Temp(t18)**		**Temp(t19)**	
Hexa value	57	01	06	F5	07	08	07	14		04	84	04	6B	04	4C
Decoded value			17.81°C		18.00°C		18.12°C			11.56°C		11.31°C		11.00°C	

16.3 Service message (T2) structure

The T2 data message transmits additional information such as the minimum and maximum Temperature over the last 24 hours. The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
Signification	Header (58)	Firmware version			Time-step	Max Temp		Min Temp		Max Temp variation		Average/ Instant

Time step

Byte	Value (Hexa)	Signification
#9	0A	10-minute time step
	0F	15-minute time step

Max Temp

Bytes	Value example (Hexa)	Signification
#2, #3	07BE	Maximum Temperature over the last 24 hours. Convert to decimal and divide by 100. Example: 07BE => 19.82°C

Min Temp

Bytes	Value example (Hexa)	Signification
#4, #5	FF06	Minimum Temperature over the last 24 hours. Convert to decimal and divide by 100. Example: FF06 => -2.50°C

Max Temp variation

Bytes	Value example (Hexa)	Signification
#6, #7	FDEA	Maximum Temperature variation between two successive measurements over the last 24 hours. Convert to decimal and divide by 100. Example: FF06 => -5.34°C (temperature drop of 5.34°C in one minute)

16.4 Javascript decoder

Decoder for the FM432t_nc_1mn product reference:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432t_nc_1mn-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

17. Decoding FM432t_nc_10mn and FM432t_nc_15mn

17.1 Types of messages

Sensor version	10 minutes	15 minutes
Data message (T1)	Every 80 minutes (8 x 10 minutes)	Every 2 hours (8 x 15 minutes)
Service message (T2)	Every 24 hours	Every 24 hours

FM432t generates two kinds of messages:

- T1: contains the Temperature data and is generated several times per day with a frequency determined by the time-step related to the product reference.
- T2: contains additional technical information. It is generated once per day.

17.2 Data message (T1) structure

17.2.1 Introduction

The T1 data message transmits **8 Temperature** values.

Each of these Temperature values is an average of several measurements. Individual measurements are performed every minute. If the data time-step is 15 minutes (for example), 15 successive individual measurements are averaged to create one 15 minute Temperature value.

T1 data messages are sent:

- every 80 minutes for a 10-minutes time-step product
- every 2 hours for a 15-minutes time-step product.

17.2.2 Timestamping

Each **Temperature** should be time stamped as follows: $t_i = t_{received} - ((8 - i) * Time\_step)$

Where:

- t_i is the timestamp for one element of the data message i ϵ {0,7}.
- $t_{received}$ is the timestamp when T1 data message is received.
- *Time_step* is the time-step, i.e the interval between two values.

17.2.3 Byte structure

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18
Signification	Header	Time-step	Temp (t0)		Temp (t1)		Temp (t2)		Temp (t3)		Temp (t4)		Temp (t5)		Temp (t6)		Temp (t7)	

17.2.4 Header

Header value (Hexa)	Signification
57	FM432t

17.2.5 Time-step

Time-step (Hexa)	Signification
0A	10 minutes
0F	15 minutes

17.2.6 Temperature calculation

Temperature values are labelled (t0) to (t7).

The formulas for obtaining Temperature values are:

Increment(t0) = ((Byte_3 * 2^8) + Byte_4)/100
Increment(t1) = ((Byte_5 * 2^8) + Byte_6)/100
Increment(t2) = ((Byte_7 * 2^8) + Byte_8)/100
…

Make sure to first convert from hexadecimal (signed) to decimal: **Byte_i** <= hex2dec(**Byte_i**)

17.2.7 Example

Given the following T1 message:

57 01 06f5 0708 0714 … 0484 046b 044c

The decoded values should be:

Byte	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**...**	**#13**	**#14**	**#15**	**#16**	**#17**	**#18**
Signification	**Header***	**Time-step**	**Temp(t0)**		**Temp(t1)**		**Temp(t2)**		**...**	**Temp(t5)**		**Temp(t6)**		**Temp(t7)**	
Hexa value	57	01	06	F5	07	08	07	14		04	84	04	6B	04	4C
Decoded value			17.81°C		18.00°C		18.12°C			11.56°C		11.31°C		11.00°C	

17.3 Service message (T2) structure

The T2 data message transmits additional information such as the minimum and maximum Temperature over the last 24 hours. The T2 data message is sent on sensor start and then every 24 hours.

Byte	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
Signification	Header (58)	Firmware version			Time-step	Max Temp		Min Temp		Max Temp variation		Average/ Instant

Time step

Byte	**Value (Hexa)**	**Signification**
#9	0A	10-minute time step
	0F	15-minute time step

Max Temp

Bytes	**Value example (Hexa)**	**Signification**
#2, #3	07BE	Maximum Temperature over the last 24 hours. Convert to decimal and divide by 100. Example: 07BE => 19.82°C

Min Temp

Bytes	Value example (Hexa)	Signification
#4, #5	FF06	Minimum Temperature over the last 24 hours. Convert to decimal and divide by 100. Example: FF06 => -2.50°C

Max Temp variation

Bytes	Value example (Hexa)	Signification
#6, #7	FDEA	Maximum Temperature variation between two successive measurements over the last 24 hours. Convert to decimal and divide by 100. Example: FF06 => -5.34°C (temperature drop of 5.34°C in one minute)

17.4 Javascript decoder

Decoder for the FM432t_nc_10mn and FM432t_nc_15mn product references:

https://github.com/Fludia-IoT/LoRaWAN-decoders/blob/master/fm432t_nc_10mn-15mn-decoder.js

All decoders:

https://github.com/Fludia-IoT/LoRaWAN-decoders

18. Accessing the dashboard to check communication and data

In case a callback has been set up to forward messages to Fludia data server, it is possible to access the dashboard with a Web browser, by typing the following URL: https://fm400-api.fludia.com/console

The login and password have been provided by email at the moment of purchase.

The dashboard displays the list of sensors and some indicators (Version number, time-step, elapsed time since last connection…).

By clicking on the + sign, you can unstack two graphs, one showing measurement data and the other message counts.

By clicking on the ID (Serial) number, you can access a more detailed page, where you can graphically navigate the measurement data, download corresponding data files and have a look at the raw messages.

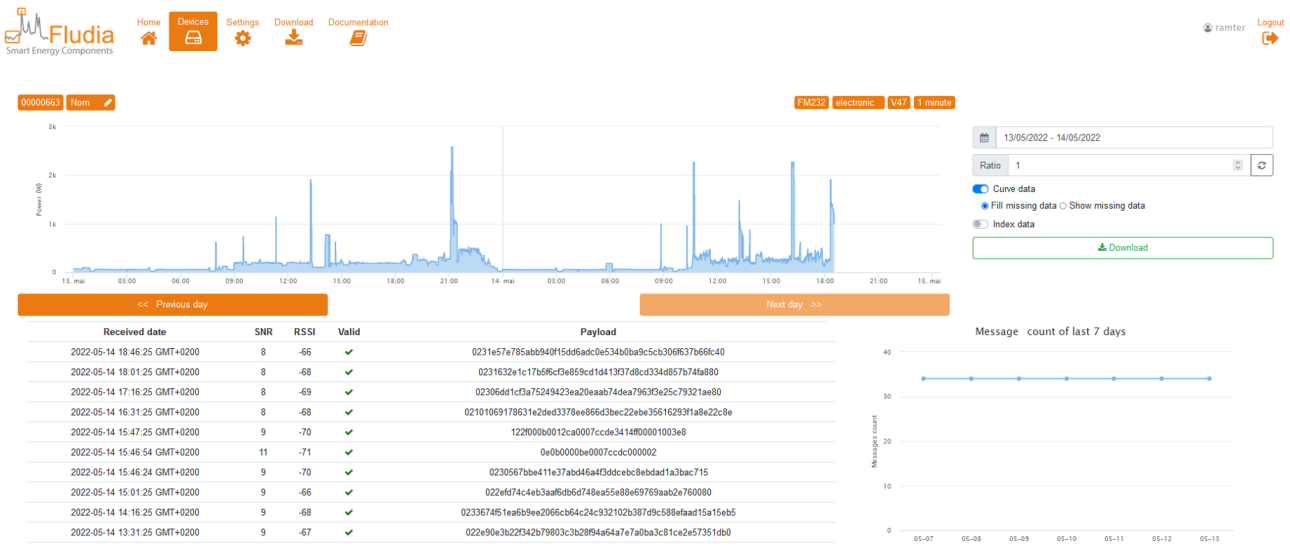
The top part of this page shows a graph with the measurement data. By default, the displayed period is two days, but a different period can be selected in the calendar.

It is possible to go back and forward in time, one day at a time, by clicking on the “Previous day” or “Next day” button.

The type of data on display can be chosen and be either curve data (could be called interval data, can refer to electric power or gas volume for example) or index data (cumulative quantity, can refer to cumulated energy or cumulated volume for example).

There are two options for curve data: “fill missing data” or “show missing data”. The “fill missing data” option creates values by spreading the energy evenly over the missing points. The “show missing value” does not create new values and the mssing periods appear clearly as blanks on the graph.

Index data is always provided without filling missing values.



By clicking on the “download” button a csv file is being created and downloaded containing the measurement data on display, with the selected option. Once opened in a spreadsheet, measurement data looks like the example below.

Timestamp (s)	Power without ratio (W)	Power with ratio (W)	Ratio	Type	Step (minute)	Date
1652468580	272	272	1	FM232	1	2022-05-13 21:03:00 GMT+0200
1652468640	647	647	1	FM232	1	2022-05-13 21:04:00 GMT+0200
1652468700	2079	2079	1	FM232	1	2022-05-13 21:05:00 GMT+0200
1652468760	1749	1749	1	FM232	1	2022-05-13 21:06:00 GMT+0200
1652468820	1409	1409	1	FM232	1	2022-05-13 21:07:00 GMT+0200
1652468880	2580	2580	1	FM232	1	2022-05-13 21:08:00 GMT+0200

19. Retrieving data from the server with the API

In case a callback has been set up to forward messages to Fludia data server, it is possible to use Fludia API.

The access is done through HTTPS with the following URL:

https://fm430-api.fludia.com/v1/API/

Each query to the API needs authentication using the Basic Auth access. The login and password are provided separately by email (they are the same as the ones used to access the dashboard).

Exploring the API before implementation.

If needed as a first step, there are several ways to explore the API:

a. CURL command line: curl -u "login:password" "https://fm430-api.fludia.com/v1/API/..."

b. Postman tool

c. swagger.fludia.com (list of requests and test capability)

19.1 Device list request

GET https://fm430-api.fludia.com/v1/API/FM400

returns the list of devices (Last 8 characters of the device ID), the timestamp of the last received message for each device and some additional attributes.

Result example:


```
[
  {
    "Nom": "",
    "SerialNumber": "a000596c",
    "Type": "FM430",
    "Type_decodage": "V2",
    "Type_capteur": "Electronique",
    "Version_firmware": "21",
    "DerniereReceptionOK_TS": "1649066589",
    "Pas": "10 minutes",
    "Actif": "actif",
    "Pile_faible": "0",
    "Ratio": "1"
  },
  {
    "Nom": "",
    "SerialNumber": "a000596d",
    "Type": "FM430",
    "Type_decodage": "V2",
    "Type_capteur": "Electromécanique",
    "Version_firmware": "21",
    "DerniereReceptionOK_TS": "1649067662",
    "Pas": "10 minutes",
    "Actif": "actif",
    "Pile_faible": "0",
    "Ratio": "1"
  }
]
```


Attributes :

SerialNumber: last 8 characters of the device ID (device ID is found on the device front panel label, not to be confused with SN number… a bit confusing, but it comes from older ways to deal with device identification).

Type: FM430 refers to all LoRa devices (FM232x and FM432x).

Type_decodage: version of message decoding performed by the server (no use for normal API usage).

Type_capteur: Electronique (electronic electricity meter), Electromécanique (electromechanical electricity meter), IR (infrared SML german electricity meter), gaz (gas meter), pulse (pulse detection), Température (temperature).

Version_firmware: firmware version

DerniereReceptionOK_TS: timestamp of the last received message

Pas: measurement time step (1 minute, 10 minutes, 15 minutes, 1 heure)

Actif: “actif” signifies that the device has sent at least 1 message in the last three days (otherwise, it is “inactive”.

Pile faible: low battery indicator. Equals 1 when the battery voltage is under a threshold, 0 otherwise (indicator available only for some of the sensors… if not available, the value is NULL)

Ratio: indicates the value of the constant that might be used to multiply the data (taken into account only in a specific request not presented in this document). This constant can be set through the API (for example to take into account the constant of an electromechanical meter in Wh per disk turn). Default value is 1.

19.2 Interval data request

GET
https://fm430-api.fludia.com/v1/API/pxm/**SerialNumber**[?limit=**n**[&tsDeb=**x**&tsFin=**x**]&show_missing=true]

Input data

SerialNumber	
limit	Maximum number of messages to be collected
&tsDeb=	First Timestamp
&tsFin=	Last Timestamp
show_missing	Option to process missing data

Returns the list of interval data, with related UTC timestamps.

&tsDeb= et &tsFin= are timestamps (in seconds) to filter data from tsDeb to tsFin (not mandatory).

&limit=n to limit to n last values.

show_missing chooses the way to process data. It is "false" by default.

Query « pxm »	**show_missing=false (default)**	**show_missing=true**
Missing radio message	Missing data are filled-up by averaged values	Missing data are not filled-up

Result example (first value is the TimeStamp, second value is the measurement value, without taking any ratio into account):

```
[
  [
    1628860800,
    57.2
  ],
  [
    1628861400,
    52.8
  ]
]
```

Depending on the sensor type (Type_capteur), the measurement values have different meaning:

- Electronique (optical reading of electronic electricity meter): the value is an average power (in Watt, to be multiplied by a constant if the meter comes with such a constant)
- Electromécanique (optical reading of electromechanical electricity meter): the value is an average

power (in Watt, to be multiplied by a constant if the meter comes with such a constant)

- IR (optical reading of infrared SML german electricity meter): the value is an average power (in Watt)
- gaz (optical reading of gas meter): the value is a volume (in tens of dm3, if the sensor has been correctly positioned on the digit left to the dm3 digit)
- pulse (pulse detection): the value is the number of pulses detected over the time step
- Température (temperature): the value is the average temperature over the time step

19.3 Index data

GET https://fm430-api.fludia.com/v1/API/index_brut/**SerialNumber**[?limit=**n**[&tsDeb=**x**&tsFin=**x**]]

Input data

SerialNumber	
limit	Maximum number of messages to be collected
tsDeb	First Timestamp
tsFin	Last Timestamp

&tsDeb= et &tsFin= for timestamps (in seconds) to filter data from tsDeb to tsFin (not mandatory).

&limit=n to limit to n last values.

Result example (first value is the TimeStamp, second value is the index value):

```
  [
    1628860800,
    578956
  ],
  [
    1628861400,
    578958
  ]
]
```

Depending on the sensor type (Type_capteur), the index values have different meaning:

- Electronique (optical reading of electronic electricity meter): the value is the cumulative energy since installation (in Watt.hour, to be multiplied by a constant if the meter comes with such a constant)
- Electromécanique (optical reading of electromechanical electricity meter): the value is the cumulative energy since installation (in Watt.hour, to be multiplied by a constant if the meter comes with such a constant)
- IR (optical reading of infrared SML german electricity meter): the value is the cumulative energy provided by the meter (in Watt.hour)
- gaz (optical reading of gas meter): the value is the cumulative volume since installation (in tens of dm3, if the sensor has been correctly positioned on the digit left to the dm3 digit)

- pulse (pulse detection): the value is the cumulative number of pulses since installation
- Température (temperature): the value is the average temperature over the time step

19.4 Message list

GET https://fm430-api.fludia.com/v1/API/trames?SerialNumber=**SerialNumber**[&limit=**n**&offset=**m**]

Input data

SerialNumber	
limit	Maximum number of messages to be collected
offset	Number of last messages to avoid

&limit=n to limit to n last messages. &offset=m to avoid the last m messages.

returns the last n messages received from the device:


```
[
  {
    "SerialNumber": "000017c5",
    "TsReception": 1649080309,
    "Valide": true,
    "Data":
"5b000615330fe30b120b030b660af7107e142a1600163015e40b870b1f0ec90be2067509df0daa0fca131016
1e",
    "Variables": {},
    "Rssi": -77,
    "Snr": 11
  },
  {
    "SerialNumber": "000017c5",
    "TsReception": 1649079109,
    "Valide": true,
    "Data":
"5b00061037095b04bf06d504bf04a307760a880c6a0c5a0c74074406800f870f7a151d18171a461be6191c14
14",
    "Variables": {},
    "Rssi": -77,
    "Snr": 8
  }
]
```


20. Contact

For further information or advice please contact us:

Fludia

support@fludia.com

01 83 64 13 94

4 ter rue Honoré d'Estienne d'Orves

92150 Suresnes, France

21. Annex A: product references and what they mean

FM432e: electricity meter optical reading

- Ref: FM432e_nc_1mn => no compression, average power measured over 1 minute; message sent every 20 minutes including 20 values
- Ref: FM432e_nc_10mn => no compression, index increment measured over 10 minutes; message sent every 80 minutes including 8 values
- Ref: FM432e_nc_15mn => no compression, index increment measured over 15 minutes; message sent every 120 minutes including 8 values
- Ref: FM432e_ap => adjustable parameters = measurement frequency, among other things, can be changed by sending a downlink. Default config: index increment measured over 15 minutes; message sent every 120 minutes including 8 values.

FM432ir: electricity infrared meter optical reading (SML protocol)

- Ref: FM432ir_nc_1mn => no compression, index increment measured over 1 minute, message sent every 15 minutes including 15 values.
- Ref: FM432ir_nc_15mn => no compression, index increment measured over 15 minutes, message sent every 120 minutes including 8 values
- Ref: FM432ir_ap => adjustable parameters = measurement frequency, among other things, can be changed by sending a downlink. Default config: index increment measured over 15 minutes; message sent every 120 minutes including 8 values.

FM432g: gas meter optical reading (ATEX)

- Ref: FM432g_nc_10mn => no compression, index increments measured over 10 minutes, message sent every 80 minutes including 8 values.
- Ref: FM432g_nc_15mn => no compression, index increments measured over 15 minutes, message sent every 120 minutes including 8 values.
- Ref: FM432g_ap => adjustable parameters = measurement frequency, among other things, can be changed by sending a downlink. Default config: index increment measured over 15 minutes; message sent every 120 minutes including 8 values.

FM432p-a: pulse reading (ATEX)

- Ref: FM432p-a_nc_1mn => no compression, index increment measured over 1 minute, message sent every 20 minutes including 20 values.
- Ref: FM432p-a_nc_10mn => no compression, index increment measured over 10 minutes, message sent every 80 minutes including 8 values.
- Ref: FM432p-a_nc_15mn => no compression, index increment measured over 15 minutes, message sent every 120 minutes including 8 values.
- Ref: FM432p-a_ap => adjustable parameters = measurement frequency, among other things, can be changed by sending a downlink. Default config: index increment measured over 15 minutes; message sent every 120 minutes including 8 values.

FM432p-n: pulse reading (non-ATEX)

- Ref: FM432p-n_nc_1mn => no compression, index increment measured over 1 minute, message sent every 20 minutes including 20 values.
- Ref: FM432p-n_nc_10mn => no compression, index increment measured over 10 minutes, message sent every 80 minutes including 8 values.
- Ref: FM432p-n_nc_15mn => no compression, index increment measured over 15 minutes, message sent every 120 minutes including 8 values.
- Ref: FM432p-n_ap => adjustable parameters = measurement frequency, among other things, can be changed by sending a downlink. Default config: index increment measured over 15 minutes; message sent every 120 minutes including 8 values.

FM432t: temperature measurement

- Ref: FM432t_1mn => average temperature measured over 1 minute; message sent every 20 minutes including 20 values
- Ref: FM432t_10mn => average temperature measured over 10 minutes; message sent every 80 minutes including 8 values
- Ref: FM432t_15mn => average temperature measured over 15 minutes; message sent every 120 minutes including 8 values