

ArrowWAN 868 / ArrowWAN MVM 868

**PROTOCOL REFERENCE
V 1.0
06/08/2020**

Index

References	3
Introduction	3
1. Radio frame	3
1.1 Frame encryption layer.....	3
1.2 Payload description and decoding.....	6
1.2.1 Measurement frame.....	6
1.2.2 Installation frame.....	8
1.2.3 Status frame.....	9
1.3 Encryption-less mode	10
2. ANNEX A – FORMAT FRAMES	10
2.1 Measurement frame.....	10
2.2 Installation frame.....	11
2.3 Status frame.....	11
3. ANNEX B – wMBUS TYPES	12
3.1 Type G: Date	12
3.2 Type F: Datetime.....	12
3.3 Type I: Datetime	13
3.4 Inverse compact profile without register.....	13
4. Contacts	14

REFERENCES

Reference	Document / link
[1]	EN13757-7:2018
[2]	EN13757-3:2018

INTRODUCTION

The ArrowWan 868 module is a LoRaWAN radio module available in two versions:

- ArrowWanMVM 868: Compact version with included inductive sensor for Maddalena MVM piston meters
- ArrowWan 868: External module with pulse input

This document explains how data is sent by the module and gives the information necessary for interpretation.

1. Radio frame

The module sends metering data according to the current setup (in terms of number of transmissions, duration of transmission windows, timing of index reading). Data is sent at a random interval inside the transmission window to limit collisions.

The module can transmit three different types of frames that are sent on different LoRaWAN FPorts:

Frame	Description	FPort
Measurement	Metering data (interval and content depend on configuration)	16
Installation	Installation data, sent during installation phase only	32
Status	Device information, periodically sent (by default once per week)	48

To ensure confidentiality data is encrypted at application level using AES128. This is an additional encryption on top of the LoRaWAN application layer. This means that payload after having been received and processed by the LoRaWAN Network Server must be decrypted. The following section describes how this encryption layer works.

Starting from firmware version 1.5.8 the frame encryption layer can optionally be disabled (data protection relies on LoRaWAN security mechanisms).

1.1 Frame encryption layer

The module has a set of 15 unique keys that are written into the module at production time (the keys are normally shipped together with the module or have to be requested to Maddalena).

These keys are used for the encryption and validation phases.

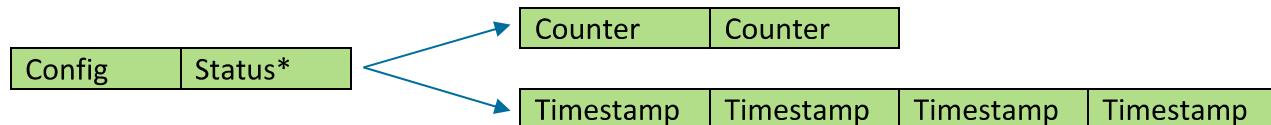
The algorithms used are:

- Encryption phase : AES128-CTR
- Validation phase: AES CMAC

The frame structure is the following

Header	Encrypted payload	Validation
See following description 1..M	Data M..N-4	MIC* N-3 N-2 N-1 N

Header structure



Header length can vary from 3 to 6 bytes, depending on the Config byte.

Config byte is encoded as follows:

Bit#	Name	Description
7	Not used	
6	MIC present	1-> MIC field is present after the payload
5	Timestamp/Counter	0 -> Counter (2 bytes) , 1 -> Timestamp (4 bytes)
4	Status	1-> Status byte is present
3		
2		
1	Key ID	Index of the key used (1-15)
0		

NOTE: even if Status and MIC are optional fields they are always enabled in the current firmware versions

Status byte is encoded as follows:

Bit#	Name
7	Tamper alarm
6	Backflow alarm
5	Leakage alarm
4	Temp error
3	Permanent error
2	Power low

1	00 -> no error
0	11 -> alarm condition

Validation phase

MIC field can be present or not, according to bit 6 of the Config byte.

If MIC is not present the validation phase must be skipped.

If MIC is present, it is used to verify the integrity of received data according to the following steps:

1. Create a 16 byte block as follows:

DevEUI								FPort	Padding (0h)						
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

2. Create the following structure:

16 byte Block	Header	Payload
---------------	--------	---------

3. Take out of the 15 keys the one with index equal to Key ID (from the Config byte)
4. With the key obtained at step #3 calculate AES-CMAC on the structure created on step #2
5. If first 4 bytes of calculated AES-CMAC match with MIC value, then integrity is verified, if not payload must be considered invalid or corrupted.

Decryption phase

To decrypt the payload of received data, follow this steps:

1. Create the initialization vector (to be used for AES CTR) as follows, depending on Timestamp/Counter Config field bit:

DevEUI								FPort	Config	Counter	Padding (0h)						
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
AES CTR IV vector if Counter is present																	

DevEUI								FPort	Config	Timestamp				Padding (0h)			
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
AES CTR IV vector if Timestamp is present																	

2. Decrypt the payload part using AES-CTR128 algorithm with the key indicated by Key ID field (as obtained in Validation phase step #3)

NOTE: the module can use any of the available keys, it is mandatory that also the decoding application has all of them available.

Example of validation and decryption

Input data

DevEUI: 78d800b018863021

Frame: 520011007afc4b5d054ff10a6f196653807e56a3ede329213c9fb0ec9c5983d59f21e5cf5d39378f

FPort: 32

Header information

Config Field: 52h (MIC is present, Status byte is present, Key ID is 2, counter present)

Status: 00h

Counter: 1100h

Key used for verification (key index 2, obtained from key database):

d9e4e19e5a4aeb410bdf36ba1448ad75

Validation phase

16 byte block: 21308618b000d87820000000000000000

MIC field from frame: 5d39378f

Calculated MIC: 5d39378f -> verification ok !

Decryption phase

IV for decryption: 21308618b000d8782052110000000000 (counter present)

Decrypted payload: db28b21701000a04010c31323033363838312d44414d6703000018224c7d2100

1.2 Payload description and decoding

The payload can be decoded following wMBUS compact frame specifications (for format frames see Annex A).

If a wMBUS parser is not available, decoding can be performed according to the following descriptions, in this case distinction of frame type can be simply done using FPort and thus the fields named “Format signature” and “Full frame CRC” can be skipped.

For decoding specific field types (date time and consumption data) see Annex B.

1.2.1 Measurement frame

This frame contains metering data according to the module configuration and is sent few times a day (default 3).

N	Field	Content	Example value
1	Format signature	Compact frame DIF/VIF table CRC	33h
2			27h

3	Full frame CRC			7Eh
4				7Ch
5	Youngest profile date	wMBUS type F	29/01/2015 00:00	00h
6				20h
7				FEh
8				11h
9	Youngest profile volume	4 bytes signed integer, LSB first, liters	53523 L	11h
10				D4h
11				00h
12				00h
13	Consumption data (wMBUS inverse compact profile without register)	LVAR		0Eh
14		Spacing control	Signed difference, hourly, delta size 2 byte	E2h
15		Spacing value	4 hours	04h
16		Profile value (H-1, LSB)	Delta = 33 L	21h
17		Profile value (H-1, MSB)		00h
18		Profile value (H-2, LSB)	Delta = 70 L	46h
19		Profile value (H-2, MSB)		00h
20		Profile value (H-3, LSB)	Delta = 145 L	91h
21		Profile value (H-3, MSB)		00h
22		Profile value (H-4, LSB)	Delta = 120 L	78h
23		Profile value (H-4, MSB)		00h
24		Profile value (H-5, LSB)	Delta = 22 L	16h
25		Profile value (H-5, MSB)		00h
26		Profile value (H-6, LSB)	Delta = 48 L	30h
27		Profile value (H-6, MSB)		00h

1.2.2 Installation frame

This frame is used only in the initial phase of radio activation, to send data relevant to the installation phase.

The device is shipped from the factory in a state that is called *storage* in which the radio part is fully shutdown. After installation using the NFC application, the module switches to *installation* state sending for six times this specific frame, and then it goes into the normal working state (called *nominal*).

N	Field	Content	Example value
1	Format signature	Compact frame DIF/VIF table CRC	DBh
2			28h
3	Full frame CRC		B7h
4			18h
5	Hardware version	2 byte binary LSB first	01h
6			00h
9	Firmware version	3 byte binary LSB first	07h
10			05h
11			01h
13	Meter ID	LVAR	12 ASCII chars
14		ASCII value	0Ch
15			50h
16			37h
17			30h
18			39h
19			30h
20			33h
21			36h
22			41h
23			46h
24			34h
25			31h
26	Current volume	4 bytes signed integer, LSB first, liters	41h
27			61h
28			BCh
29			00h
30	Current date time	wmBUS type I	30h
31			32h
32			10h
33			FEh
34			11h
35			00h

1.2.3 Status frame

This frame is sent once a week (default configuration) and contains some diagnostic information.

N	Field	Content	Example value	
1	Format signature	Compact frame DIF/VIF table CRC		AFh
2				B0h
3	Full frame CRC			B9h
4				1Bh
5	Error flag	4 byte binary LSB first (see error flag table)	2 (battery low)	02h
6				00h
7				00h
8				00h
9	Date	wmBUS type G	30/01/2015	FEh
10				11h
11	Hardware version	2 byte binary LSB first	0001	01h
12				00h
13	Firmware version	3 byte binary LSB first	1.5.7	07h
14				05h
15				01h
16	Reset counter	2 bytes signed integer, LSB first	562	32h
17				02h
18	Battery	2 bytes signed integer, LSB first	3652 days left	44h
19				0Eh
20	RSSI	1 byte unsigned integer, LSB first (interpreted as negative value)	-93 dBm	5Dh
21	TX Power	1 byte unsigned integer, LSB first	20dBm	14h
22	TX Mode	LVAR	“Tm” (mode T, meter)	02h
23		wMBUS Mode name (ascii)		6Dh
24				54h
25	RX Mode	LVAR	“Tm” (mode T, meter)	02h
26		wMBUS Mode name (ascii)		6Dh
27				54h
28	Radio TX count	4 byte unsigned integer	89563	DBh
29				5Dh
30				01h
31				00h

Error flag table

Byte 0	Bit 0 : Tamper
	Bit 1 : Battery low

	Bit 2: Reserved
	Bit 3: Power reset occurred
	Bit 4: Reserved
	Bit 5: Reserved
	Bit 6: Reserved
	Bit 7: Reserved
Byte 1	Reserved
Byte 2	Reserved
Byte 3	Reserved

1.3 Encryption-less mode

If the module is configured to disable the additional encryption layer, only the payload is sent (no header nor MIC).

An additional byte is also added, to express the status value. This status byte is encoded as described in 1.1.

To distinguish encryption-less mode from standard mode, the frames are sent on a different port as follows:

Frame	FPort
Measurement	116
Installation	132
Status	148

2. ANNEX A – FORMAT FRAMES

2.1 Measurement frame

Standard mode format signature: 3327h

Encryption-less mode format signature: 5752h

Data record	DIF/DIFE	VIF/VIFE
Youngest profile date	84h 04h 32 bit signed integer, storage #8	6Dh Date type F
Youngest profile volume	84h 04h 32 bit signed integer, storage #8	13h Volume liters
Inverse compact profile without register	8Dh 04h Variable length binary, storage #8	93h 13h Volume liters, inverse compact profile without register

Status (only in encryption-less mode)	01h 8 bit unsigned integer	FDh 17h Error flag
---------------------------------------	-------------------------------	-----------------------

2.2 Installation frame

Standard mode format signature: DB28h
 Encryption-less mode format signature: 7C32h

Data record	DIF/DIFE	VIF/VIFE
Hardware version	02h 16 bit signed integer	FDh 0Dh Hardware version
Firmware version	02h 16 bit signed integer	FDh 0Fh Other firmware version
Meter ID	0Dh Variable length binary	FDh 11h Customer
Volume	04h 32 bit signed integer	13h Volume in liters
Datetime	06h 48 bit signed integer	6Ch Date type I
Status (only in encryption-less mode)	01h 8 bit unsigned integer	FDh 17h Error flag

2.3 Status frame

Standard mode format signature: AFB0h
 Encryption-less mode format signature: E8F4h

Data record	DIF/DIFE	VIF/VIFE
Error flag	04h 32 bit signed integer	FDh 97h 1Dh Error flags
Date	02h 16 bit signed integer	6Ch Date type G
Hardware version	02h 16 bit signed integer	FDh 0Dh Hardware version
Firmware version	02h 16 bit signed integer	FDh 0Fh Other firmware version
Reset counter	02h 16 bit signed integer	FDh 60h Reset counter
Battery	02h 16 bit signed integer	FDh 74h Remaining battery lifetime (days)
Average RSSI value	01h	FDh F1h 10h

	8 bit unsigned integer	RX RF level
TX Power	01h 8 bit unsigned integer	FDh F1h 11h TX RF level
TX mode	0Dh Variable length binary	FDh 9Ah 11h Digital ouput Radio TX
RX mode	0Dh Variable length binary	FDh 9Bh 10h Digital input Radio RX
Radio TX count	04h 32 bit unsigned integer	FDh E1h 11h Cumulation counter Radio TX
Status (only in encryption-less mode)	01h 8 bit unsigned integer	FDh 17h Error flag

3. ANNEX B – wMBUS TYPES

3.1 Type G: Date

2 ⁷ 2 ⁶ 2 ⁵ 2 ⁴ 2 ³ 2 ² 2 ¹ 2 ⁰	Day: UI5 [1 to 5] < 1 to 31 > "0": every day
2 ¹⁵ 2 ¹⁴ 2 ¹³ 2 ¹² 2 ¹¹ 2 ¹⁰ 2 ⁹ 2 ⁸	Month: UI4 [9 to 12] < 1 to 12 > "15": every month
	Year: UI7 [6 to 8,13 to 16] < 0 to 99 > 127: every year

3.2 Type F: Datetime

2 ⁷ 2 ⁶ 2 ⁵ 2 ⁴ 2 ³ 2 ² 2 ¹ 2 ⁰	Min: UI6 [1 to 6] < 0 to 59 > ; 63: every minute
2 ¹⁵ 2 ¹⁴ 2 ¹³ 2 ¹² 2 ¹¹ 2 ¹⁰ 2 ⁹ 2 ⁸	Hour: UI5 [9 to 13] < 0 to 23 > ; 31: every hour
2 ²³ 2 ²² 2 ²¹ 2 ²⁰ 2 ¹⁹ 2 ¹⁸ 2 ¹⁷ 2 ¹⁶	Day: UI5 [17 to 21] < 1 to 31 > ; 0: every day
2 ³¹ 2 ³⁰ 2 ²⁹ 2 ²⁸ 2 ²⁷ 2 ²⁶ 2 ²⁵ 2 ²⁴	Month: UI4 [25 to 28] < 1 to 12 > ; 15 every month
	Year: UI7 [22 to 24 ; 29 to 32] < 0 to 99 > ; 127 every year
	Hundred year: UI2 [14 to 15] < 0 to 3 > ; this year is 1900+100*hundred year + year
	IV B1 [8] IV < 0 > = valid ; IV < 1 > = invalid
	SU B1 [16] IV < 0 > = standard time ; IV < 1 > = summer time
	RES1 B1 [7] < 0 > reserved for future use

3.3 Type I: Datetime

Byte /bit	MSBit							LSBit
LSB	8	7	6	5	4	3	2	1
	16	15	14	13	12	11	10	9
	24	23	22	21	20	19	18	17
	32	31	30	29	28	27	26	25
	40	39	38	37	36	35	34	33
MSB	48	47	46	45	44	43	42	41

Second	UI6 [1 to 6]	< 0 to 59 > ; 63: every second1)
Minute	UI6 [9 to 14]	< 0 to 59 > ; 63: every minute2)
Hour	UI5 [17 to 21]	< 0 to 23 > ; 31: every hour2)
Day	UI5 [25 to 29]	< 1 to 31 > < 0 > (0 = not specified2)
Month	UI4 [33 to 36]	< 1 to 12 > < 0 > 0 = not specified2)
Year	UI7 [30 to 32+37 to 40]	< 0 to 99 > < 127 > 127 = not specified2)
Day of the week	UI3 [22 to 24]	< 1 to 7 > 1 = Monday 7 = Sunday 0 = not specified2)
Week	UI6 [41 to 46]	< 1 to 53 > 0 = not specified 2)
Time invalid	UI1 [16]	1 = invalid ; 0 = valid
Time during daylight savings	UI1 [7]	1 = yes (summer time) ; 0 = no
Leap year	UI1 [8]	1 = leap year ; 0 = standard year
Daylight savings deviation (hour) ³⁾	UI1 [15] UI2 [47 to 48]	< 0 to 1 > (1 = + ; 0 = -) < 0 to 3 > 0 = no daylight savings

3.4 Inverse compact profile without register

The inverse compact profile contains a list, equally spaced in time, of delta values (consumption) in reverse temporal order. The base value for the deltas is taken from Youngest profile volume.

LVAR: total number of bytes following (equals to 2 + delta size * number of elements)

Spacing control byte:

bit 6..7: Increment mode	bit 4..5: Spacing unit	bit 0..3: Element size
00 _b = Absolute value	00 _b = seconds	Profile DIF, low nibble only, but except 0D _h and except 0F _h
01 _b = Increments	01 _b = minutes	
10 _b = Decrements	10 _b = hours	
11 _b = Signed difference	11 _b = days/month	

Spacing value byte:

Spacing value	Spacing unit	Meaning
0	00 _b -11 _b ^a	Elements of an array, not spacing in time
1-250	all	Number of days, hours, minutes or seconds between values
251	all	Reserved
252	all	Reserved
253	00 _b -10 _b 11 _b	Reserved; a half month between values
254	00 _b 01 _b 10 _b 11 _b	Reserved six full months between values three full months between values a full month between values
255	all	Reserved

^a The spacing unit is used to address up to four columns. If one column is needed, only the spacing unit 00b shall be used. If several columns are needed, the spacing unit describes the concerned column number by formula spacing unit +1 (e.g. spacing unit 01b indicates column 2).

4. Contacts

Distribué par :

Compteur-énergie.com

Tel : +33 (0)360 800 010

Mail : contact@compteur-énergie.com