WMBUS ADVANCED FEATURE B0

Document update

Version	Date	Comments
B0	27/09/2013	PTR

Reference documentation

Date/version	Description
EN 13757 part 1	*Communication system for meters and remote reading of meters . Part 1: Data exchange.*
~~EN 13757 part 2~~	~~Physical and link layer, twisted pair baseband (M-Bus)~~
EN 13757 part 3	Dedicated application layer (M-Bus) CI field definition + associated frame/packet format and content Codage used (time, BCD, floating, Listening window management week/day/hour …
EN 13757 part 4	Wireless meter readout
EN 60870-5-2	Transmission protocols, Section 5.2 Link transmission procedures
OMS-Spec_Vol2_Primary_v301c	Open Metering System Specification Vol.2 – Primary Communication (Issue 3.0.1 / 2011-01-29 (Release))

Notations / conventions

	Description
ARF-WMBUS or ARF Module	WMBUS Adeunis-RF Module
WMBUS	Wireless M-Bus
Collector	= other
ARF-AES128	Adeunis specific AES128 encryption

Hexadecimal numbers are marked with a suffix "h" or a prefix “0x”. Binary coded numbers are marked with a suffix "b". Numbers without suffix are decimal numbers except another coding is explicitly declared

Summary

Illustrations

1 Introduction

This document introduces advanced use mode of the ARF WMBUS module. The following use cases are addressed:

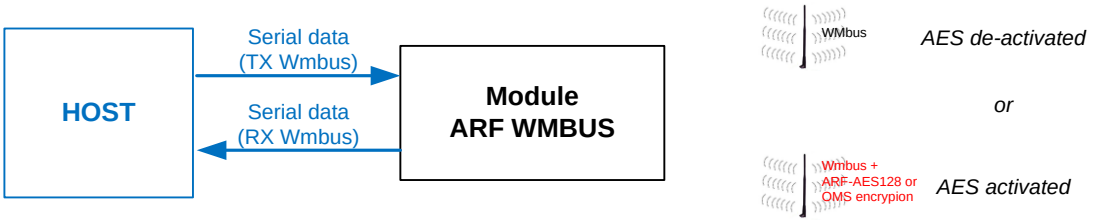
- OMS 3.0.1 mode 0 and 5 support
- Adeunis specific AES128 encryption
- Cyclic frame retransmission
- Specific output format support

The OMS or ARF-AES128 modes are not supported by bidirectional mode (T2/S2/R2).

Reminder on module ARF-WMBUS : the ARF-WMBUS module allows to send and received WMBUS frame over the air.

- The frame to be transmitted are received on the serial input of the ARF module.
- When a RF WMBUS frame is received, data are sent (to the host) by the ARF module on its serial output.

If the ARF-AES128 or the OMS5 is activated, the WMBUS transmission will be ciphered else plain WMBUS frame will be transmitted OVER THE AIR.



2 OMS

ARF Module supports OMS3.0.1, encryption mode 5: the AES-encryption uses a block size of 16 Bytes and a 128 Bit key with cipher block chaining. The CBC (Cipher Block Chaining) encryption for AES128 uses a dynamic 128 bit (16 Byte) initialisation vector to start the encryption of the first block.
The link layer header (including ID) and the fixed 4-bytes or 12-bytes header after the CI-field, are never encrypted

The module is able to encrypt / decrypt an OMS frame according to encryption mode 5. It supports:

- Ci field 0x72 (12 bytes long header) and 0x7A (4 bytes short header)
- The Configuration word (of the header) can be set to 0 (no encryption/decryption) or to 5 (encryption/decryption mode 5 activated)
- The Configuration word (of the header) also contains the number of 16 bytes blocks to encrypted. The module support partial encryption/decryption.
- The initialization vector is build dynamically according to OMS specification.
- The cipher bloc chaining is used for encryption / decryption

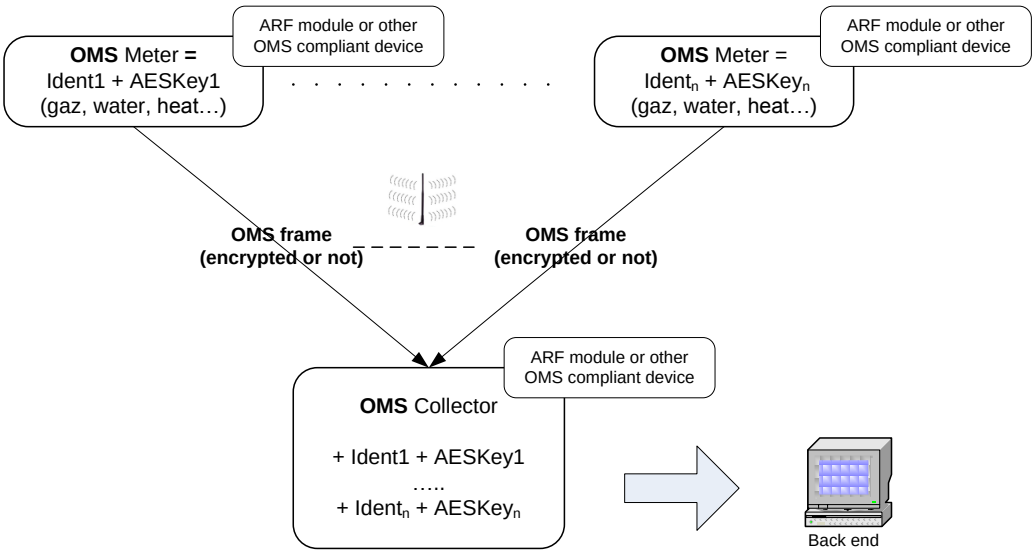


Figure 1: OMS overview

2.1 OMS Command interface

cde	Param Hex	Description	Typical response delay
h	0x30 0x35	AES mode « 0 » no AES « 5 » OMS5 encryption	
kc	4+16 hexadecimal bytes	**Add** a couple (BCD "SN" + AES key) in an array of 51 elements/couples. The couple is saved in non volatile memory when using the "kc" command. **When receiving an OMS frame** (Ci 0x72h or Ci 0x7Ah), the BCD "SN" of the OMS frame is used to find the associated AES key in the array. The AES key is then used to decrypt the OMS frame according to OMS5. (If the key is not found, see note 1) **When transmitting an OMS frame** (Ci 0x72h or Ci 0x7Ah), the BCD "SN" of the OMS frame is use to find the associated AES key in the array. The AES key is then used to encrypt the OMS frame according to OMS5. In this case the frame sent is ciphered. (If the key is not found, see note 1) When the table is full, no more element can be added. To add a new element the table must be erased ("kERA" command); All previous elements will be lost. Read back not possible.	
kr		Display list of couple (Only couple identifier, the associated AES key is not readable). Separator is <CR> 0dh.	
kERA		Erase the array of couple (BCD "SN" + AES key)	33 ms

Figure 2: OMS command

Note 1: if the couple is not found in the array of "couple" the ARF-AES 128 bits key will be used. This feature allows to use the same AES key on meter side and collector side.

Note 2: the OMS encryption is not performed if parameter h has not the value 5.
the OMS encryption is not performed if CiField is not 0x72 or 0x7A
the OMS encryption is not performed if configuration word is not set to mode 5
the OMS encryption is not performed if number of block is 0 (in configuration word)

	Meter manuf **Meter Ident** **Meter AES key**	**kc (ident + AES key)** **Command to be send (22 bytes)**	
Manuf. Ident. AES key	0x46 0x06 0x62 0x28 0x79 0x14 0x55 0x08 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x10 0x11 0x12 0x13 0x14 0x15 0x16	0x6B 0x63 0x62 0x28 0x79 0x14 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x10 0x11 0x12 0x13 0x14 0x15 0x16	kc ident AESkey

Figure 3 Example of kc command for a meter

2.2 How to use OMS on Adeunis module

You can replay the following example using the project file “OMS demo TX.ptp” and “OMS demo RX.ptp”

Transmitter side		Receiver side	
Command	**Comments**	**Command (blue) Reception (red)**	**Comments**
R0	Restore default setting	R0	Restore default setting
FD (46h44h)	Cfield=0x44	kERA	Erase OMS array
LMMT	Meter mode T	4Ah 62h 28h 79h 14h 55h 08h	J=@to be filtered
kERA	Erase OMS array	4Bh 62h 28h 79h 14h 55h 08h	K=@to be filtered
49h 46h 06h 62h 28h 79h 14h 55h 08h	I=manuf ID + ADDR 46h 06h 62h 28h…	44h 46h 06h	D=Manuf ID to be filtered
6Bh 63h 62h 28h 79h 14h 01h 02h 03h 04h 05h 06h 07h 08h 09h….	kc=Ident + AESkey	H1	RX Filter activation
h5	Activate OMS	LOMTGO	Other mode T, continuous reception
Z	Goto data mode	Z	Goto data mode
FF FF 1C 72 62 28 79 15 46 06 55 08 A2 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00	Tx with config word = 0	FF 1C 44 46 06 62 28 79 14 55 08 72 62 28 79 15 46 06 55 08 A2 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00 20	Frame received without encryption
FF FF 1C 72 62 28 79 15 46 06 55 08 A2 00 10 05 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00	Tx with config word = 5	FF 1C 44 46 06 62 28 79 14 55 08 72 62 28 79 15 46 06 55 08 A2 00 10 05 7D CE D7 C5 00 7F 75 0B 06 13 E5 EF 0C 0E ED 03 29	Frame received with OMS encryption
		Go back to command mode on receiver	
		h5	Activate OMS
		6Bh 63h 62h 28h 79h 14h 01h 02h 03h 04h 05h 06h 07h 08h 09h….	kc=Ident + AESkey
		Z	Goto data mode
FF FF 1C 72 62 28 79 15 46 06 55 08 A2 00 10 05 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00	Tx with config word = 5	FF 1C 44 46 06 62 28 79 14 55 08 72 62 28 79 15 46 06 55 08 A2 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00 20	The frame is received decrypted

3 Specific ARF-AES128 encryption

ARF-AES encryption (decryption) is performed:

- On data field,
- for each block of 16 data bytes ; If the last block is not 16 bytes long, the block will be transmitted without encryption.
- The encryption of each block is performed using and the preset initialization vector.

3.1 Typical ARF-AES128 use cases

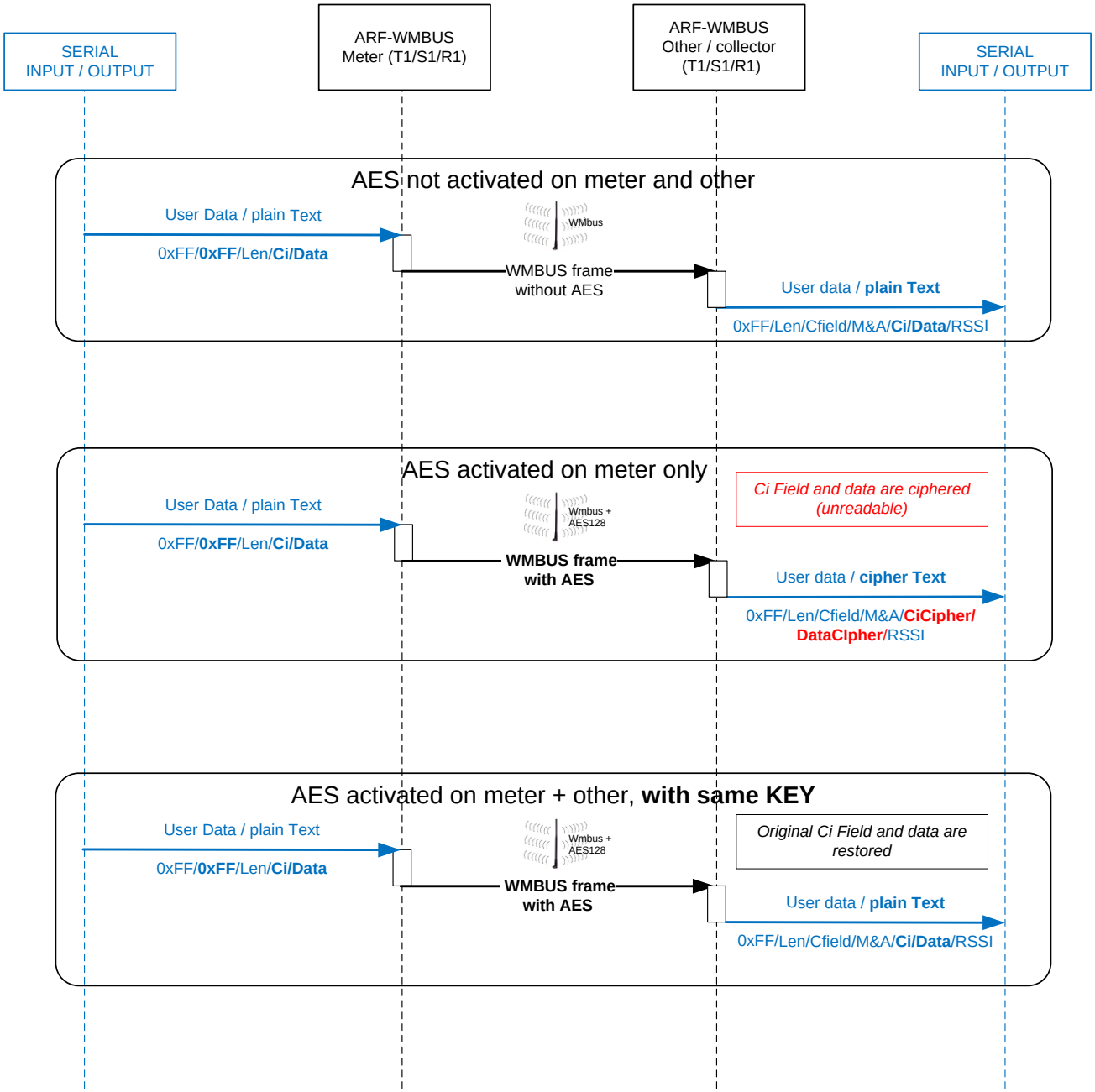


Figure 4: ARF-AES128 data exchange on ARF-WMBUS

3.2 ARF-AES128 Command interface

	Param Hex	Description	Typical response delay
h	0x30 0x31 0x35	AES mode « 0 » no AES « 1 » ARF-AES128	
kk	16 hexadecimal bytes	Set and **SAVE** the ARF-AES 128bits key Read back not possible.	
ki	16 hexadecimal bytes	Set the initialization vector 128bits (default:00…..)	
k?		Read back of initialization vector	
ks		**SAVE** the initialization vector	

Tableau 1: ARF-AES128 setup

3.3 How to use ARF-AES

You can replay the following example using the project file "AESARF demo TX.ptp" and "AESARF demo RX.ptp"

Transmitter side		Receiver side	
Command	**Comments**	**Command (blue)** **Reception (red)**	**Comments**
R0	Restore default setting	R0	Restore default setting
LMMT	Meter mode T	4Ah 62h 28h 79h 14h 55h 08h	J=@to be filtered
49h 46h 06h 62h 28h 79h 14h 55h 08h	I=manuf ID + ADDR 46h 06h 62h 28h…	4Bh 62h 28h 79h 14h 55h 08h	K=@to be filtered
		44h 46h 06h	D=Manuf ID to be filtered
6Bh 6Bh 22h 02h 03h 04h 05h 06h 07h 08h 09h….	kk=AESkey Set AES key	H1	RX Filter activation
h1	Activate ARF-AES	LOMTGO	Other mode T, continuous reception
Z	Goto data mode	Z	Goto data mode
FF FF 1C 72 62 28 79 15 46 06 55 08 A2 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00	Tx with AES	FF 1C 44 46 06 62 28 79 14 55 08 72 A8 CA 87 7F 8A E1 26 5C 51 12 69 7C BA D2 52 37 01 00 00 42 6E 02 00 82 01 6E 03 00 71	Frame received with AES encryption
		Go back to command mode on receiver	
		h1	Activate ARF-AES
		6Bh 6Bh 22h 02h 03h 04h 05h 06h 07h 08h 09h….	kk=AESkey Set AES key
		Z	Goto data mode
FF FF 1C 72 62 28 79 15 46 06 55 08 A2 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00	Tx with AES	FF 1C 44 46 06 62 28 79 14 55 08 72 62 28 79 15 46 06 55 08 A2 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00 1B	The frame is received decrypted
Go back to command mode on Transmitter		Go back to command mode on receiver	
6B 69 22 22 22 22 08 09 10 11 12 13 14 15 16 11 11 11	Update initialization vector ki=new value		
Z	Goto data mode		
FF FF 1C 72 62 28 79 15 46 06 55 08 A2 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00	Tx with AES and new initialization vector	FF 1C 44 46 06 62 28 79 14 55 08 72 40 0A 5B 37 4E 0F 45 19 B0 13 04 15 39 3E 1A 7F 01 00 00 42 6E 02 00 82 01 6E 03 00 25	Frame received with AES encryption To decrypt the frame, update the initialization vector on collector side

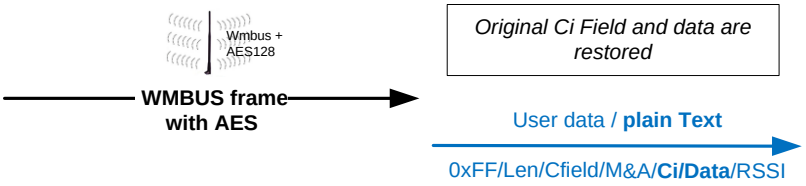
3.4 AES Serial frame format

Three kinds of serial frame :

1. Output format (RX/other side) ➔ frame received on RF link and output through the serial TXD line
2. Standard input format (TX/meter side) ➔ frame to be send over WMbus RF link
3. Extended input format (FMT_EXTENDED) ➔ for bidirectional mode (on other side), frame to be send over WMbus RF link

3.4.1 Output format (on RX side)

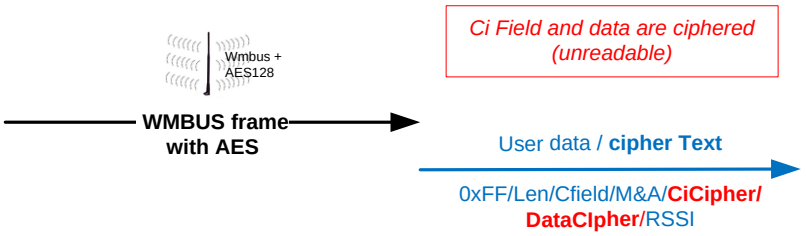
❖ If the **AES128 is activated on both side** (meter+ other) the received CI field and data payload are plain text.



0xFF	Len	C	M&A	CI	Data	RSSI
1 byte	1 byte	1 byte	2 + 6 bytes	1 byte	1➔ 127 bytes	1 byte
Wake up	Payload len	Cfield	Manuf ID & Address	**Plain** CI field	Payload=**Plain Text** Max = 127 mode T Max = 95 modes R/S	RSSI level

Example : FF 04 **5A** 06 46 41 52 46 5F 30 32 **49** 64 55 44 52 **47**
➔ 4 bytes of data payload, C=**5A**, M&A = 0x06FARF_02, CI=**0x49**, Data = dUDR, RSSI = 0x47

❖ If the **AES128 is activated on meter side only** (not on other side), the received CI field and the data payload are encrypted.



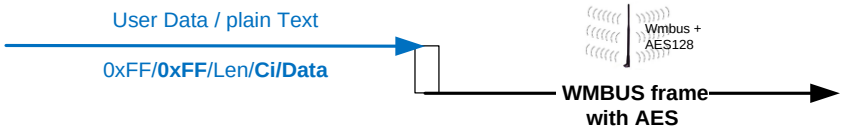
- CI field and data payload can be decrypted by the host controller with the AES128 key used by the meter
- Padding is added after the CI and data payload to insure a length modulo 16 bytes (AES block length)

0xFF	Len	C	M&A	CI	Data	RSSI
1 byte	1 byte	1 byte	2 + 6 bytes	1 byte	1➔ 127 bytes	1 byte
Wake up	Payload len	Cfield	Manuf ID & Address	**Crypted** CI field	Payload=**Crypted Text** Max = 127 mode T Max = 95 modes R/S	RSSI level

3.4.2 Standard input format (TX side)

❖ Unchanged : the data payload send to the ARF-WMBUS is plain text.

- Padding is added (if required) to CI and data payload to insure a length modulo 16 bytes (AES processing block length)
- The padding bytes will be received on collector side.



0xFF	0xFF	Len	CI	Data
1 byte	1 byte	1 byte	1 byte	1➔ 127 bytes
Wake up	SOF standard	Payload len	CI field	Payload Max = 127 mode T Max = 95 modes R/S

Example : FF **FF** 0C **49** 30 55 44 41 35 36 37 38 39 41 42 43
➔ 12 bytes of data payload, CI=**0x49**, Data=1UDA56789ABC

3.4.3 Extended input format (other side)

AES128 activation is not guaranteed / tested for bidirectional mode ; using AES128 will affect bidirectional timing and can lead to protocol error.

0xFF	0xFE	Len	M&A	CI	Data
1 byte	1 byte	1 byte	2 + 6 bytes	1 byte	1➔ 127 bytes
Wake up	SOF étendu	Payload len	Manuf ID & Address	CI field	Payload Max = 127 mode T Max = 95 modes R/S

Example : FF **FE** 04 **06 46 41 52 46 5F 30 32** 49 **64 55 44 52**
➔ Extended frame (**0xFE**), 4 bytes of data payload, M&A = 0x06FARF_02, CI=**0x49**, Data = **dUDR**

4 Cyclic synchronous retransmission

Cyclic retransmission feature allows to transmit synchronously a WMBUS frame. The frame is transmitted with a predefined interval. The frame content can be updated via the serial link.

This feature is usable:

- On meter mode unidirectional, T1/S1/R1. (cannot be used in bidirectional mode).
- The long preamble is not supported

To preserve the synchronization during a frame update, avoid to update the frame when the frame transmission is in progress (end of interval). A simple way to do this is to refresh the frame after the transmission, when the module feedbacks the character '>' (transmission performed).

In this mode, you can activate the following options:

- Usage of a transmission jitter. The transmission jitter is based on OMS requirement (allow to have a jitter from +/- 0.62 ms up to +/- 166s with duration from 2s up to 5400s). For durations > 5400s, the jitter is limited to +/-166s.
- Auto-incrementation of the access number for OMS frame content (with Ci 0x72 and 0x7A) ➔ activating OMS 5 allows to transmit an encrypted message using each time a different access number.

4.1 Jitter

For synchronous transmission, the jitter calculation uses OMS specification:

- It is based on a nominal transmission time point (predefined interval) and an additional scatter.
- The next nominal transmission time point is given by
  - ✓ the last nominal transmission time point and the nominal transmission interval.
  - ✓ The scatter is the deviation from the nominal transmission time point
- Note: cyclic retransmission (Tnom) must be even (Tnom = N x 2 seconds)

The next individual transmission time point « t_{TX} (n+1) » is calculated by the last transmission time point « t_{TX} (n) » and the individual transmission interval for next transmission (n+1) based on the access number:

$$tTX(n+1) = tTX(n) + TACC(n+1)$$

with

$$TACC(n+1) = (1 + (|ACC - 128| - 64) / 2048) \times Tnom$$

$$Tnom = N \times 2 \text{ seconds}$$

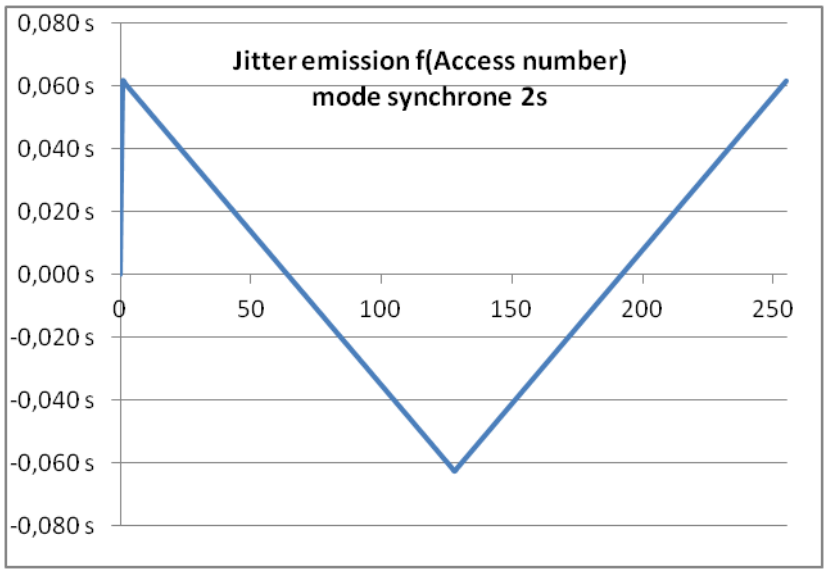


Figure 5: Transmission time point (Jitter) for 2s interval

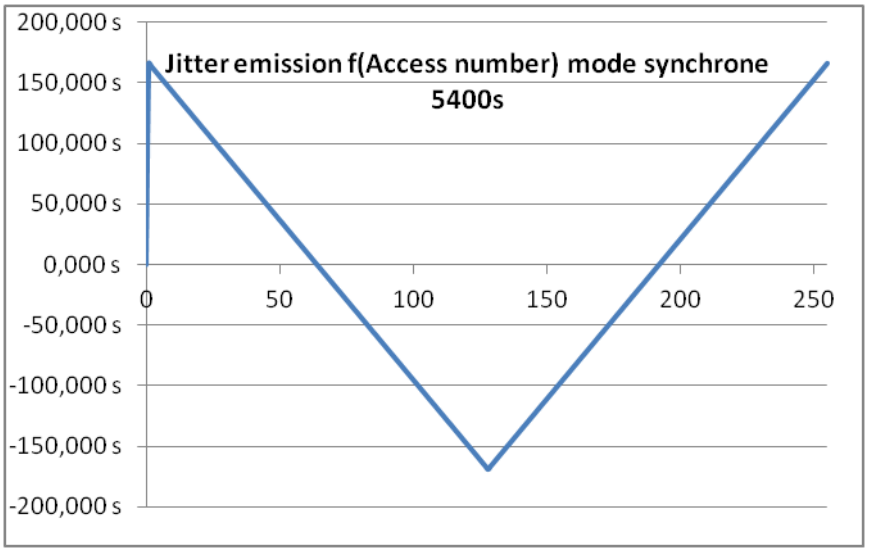


Figure 6: Maximum transmission time point (Jitter) for 5400s interval

4.2 Command interface

	Param Hex	Description	Typical response delay
c	3 hexa bytes	**(In unidirectional meter mode only)** delay for cyclic retransmission of the last frame (host to module). Unit 1 s, from 1s up to 86400s (24 hours) 0 ➔ no retransmission (0x000001-0x015180) ➔ cyclic retransmission ON	
j	0x30 0x31	Jitter activation "0" no jitter "1" jitter management	

Tableau 2: cyclic repetition setup

4.3 How to use Cyclic synchronous transmission without jitter

You can replay the following example using the project file "SYNCH demo TX.ptp" and "SYNCH demo RX.ptp"

Transmitter side		Receiver side	
Command	**Comments**	**Command (blue)** **Reception (red)**	**Comments**
R0	Restore default setting	Configure the receiver in RX mode (LOMT + filtering) according to "SYN demo RX.ptp"	
LMMT	Meter mode T		
49h 46h 06h 62h 28h 79h 14h 55h 08h	I=manuf ID + ADDR 46h 06h 62h 28h…		
63h 00h 00h 02h	c=2 secondes ➔retransmission ON		
Z	Goto to data mode		
FF FF 1C 72 62 28 79 15 46 06 55 08 A2 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00	First frame transmission		
10:50:41.83 [RX]- 3E	First TX time		
10:50:43.83 [RX]- 3E 10:50:45.83 [RX]- 3E 10:50:47.83 [RX]- 3E	Second TX time …		

Note: the transmission jitter can be activated using the "j1" command.

4.4 How to use Cyclic synchronous transmission with OMS frame

You can replay the following example using the project file “SYNCHOMS demo TX.ptp” and “SYNCH demo RX.ptp”

Transmitter side		Receiver side	
Command	**Comments**	**Command (blue) Reception (red)**	**Comments**
R0	Restore default setting	R0	Restore default setting
FD (46h44h)	Cfield=0x44	kERA	Erase OMS array
LMMT	Meter mode T	4Ah 62h 28h 79h 14h 55h 08h	J=@to be filtered
kERA	Erase OMS array	4Bh 62h 28h 79h 14h 55h 08h	K=@to be filtered
49h 46h 06h 62h 28h 79h 14h 55h 08h	I=manuf ID + ADDR 46h 06h 62h 28h…	44h 46h 06h	D=Manuf ID to be filtered
6Bh 63h 62h 28h 79h 14h 01h 02h 03h 04h 05h 06h 07h 08h 09h….	kc=Ident + AESkey	H1	RX Filter activation
h5	Activate OMS	LOMTGO	Other mode T, continuous reception
Z	Goto data mode	Z	Goto data mode
FF FF 1C 72 62 28 79 15 46 06 55 08 A2 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00	First frame transmission Ci 0x72h with config word = 0	FF 1C 44 46 06 62 28 79 14 55 08 72 62 28 79 15 46 06 55 08 00 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00 67	Frame received without encryption Access Number = 00
11:07:53.50 [RX] - >	Second TX time	FF 1C 44 46 06 62 28 79 14 55 08 72 62 28 79 15 46 06 55 08 01 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00 74	Access Number = 01
11:07:55.50 [RX] - >	3rd TX time	FF 1C 44 46 06 62 28 79 14 55 08 72 62 28 79 15 46 06 55 08 02 00 10 00 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00 27	Access Number + 1
11:07:57.50 [RX] - > 11:07:59.50 [RX] - >	Next TxTime		Access number +1

Note: the transmitted frame will be encrypted if configuration word = 05.

4.4.1 Illustration of access number update for Ci 0x7A + encryption operational

Transmitter side		Receiver side	
Command	**Comments**	**Command (blue) Reception (red)**	**Comments**
FF FF 23 7A A2 00 10 05 2F 2F 0B 6E 01 00 00 42 6E 02 00 82 01 6E 03 00 C2 01 6E 04 00 82 02 6E 05 00 C2 02 6E 06 00	1st TX time Ci 0x7A with config word = 5	FF 23 44 46 06 62 28 79 14 55 08 7A 00 00 10 05 D8 AB E6 07 8D 4A BD 18 30 D6 09 29 F3 3D 62 54 C2 01 6E 04 00 82 02 6E 05 00 C2 02 6E 06 00 70	First reception Encrypted data Access Number = 00
11:39:57.23 [RX] - >	Second TX time	FF 23 44 46 06 62 28 79 14 55 08 7A 01 00 10 05 49 6D 55 7E CD 7F FF E5 78 A7 0A 98 21 66 6A F5 C2 01 6E 04 00 82 02 6E 05 00 C2 02 6E 06 00 2B	Access Number = 01
11:39:59.23 [RX] - >	3rd TX time	FF 23 44 46 06 62 28 79 14 55 08 7A 02 00 10 05 04 76 2B 40 04 5D 46 A7 46 19 95 19 78 D2 5C 31 C2 01 6E 04 00 82 02 6E 05 00 C2 02 6E 06 00 75	Access Number + 1
	Next TxTime		Access number +1

5 Specific frame output format

A specific output format is available.

cde	Param Hex	Description	Typical response delay
o	0x30 0x35	Output format « 0 » standard Adeunis frame output format « 1 » specific output format mode '1'	
U	4	Allow to use 19200 serial baud rate	

Convention used :

REd = SOF/EOF : format specific
Len : format specific
WBUS data : **same value for all format**
RSSI

5.1 Output format "0", default format:

0xFF	Len	C	M&A	CI	Data	RSSI
1 byte	1 byte	1 byte	2 + 6 bytes	1 byte	1➔ 127 bytes	1 byte
Wake up	Payload len	Cfield	Manuf ID & Address	**Plain** CI field	Payload=**Plain Text** Max = 127 mode T Max = 95 modes R/S	RSSI level

Example

```
FF 23 44 46 06 62 28 79 14 55 08 7A 78 00 10 05 85 92 18 BE 6F 86 2A 2F 2F DA A2 21 E3
15 E1 5E C2 01 6E 04 00 82 02 6E 05 00 C2 02 6E 06 28
```

5.2 Output format "1":

0x68	Len	C	M&A	CI	Data	RSSI	0x16
1 byte	1 byte	1 byte	2 + 6 bytes	1 byte	1➔ 127 bytes	1 byte	1 byte
SOF	Len From C to RSSI	Cfield	Manuf ID & Address	**Plain** CI field	Payload=**Plain Text** Max = 127 mode T Max = 95 modes R/S	RSSI level	EOF

Example

```
68 2E 44 46 06 62 28 79 14 55 08 7A 7D 00 10 05 1B 79 3F 9D 32 F7 7D 21 C8 90 17 3C 34
9C 30 50 C2 01 6E 04 00 82 02 6E 05 00 C2 02 6E 06 00 28 16
```