IDEMIA
augmented identity

MorphoAccess® SIGMA Family & MorphoWave® Compact

Application note - Wiegand formats

WARNING

Revision History

The table below contains the history of changes made to the present document.

Version	Date	Description
01	October 2016	New document's reference based on 2014_0000000453_v6
02	April 2017	Add MorphoAccess® SIGMA EXTREME Series terminal
03	December 2017	Update company name (IDEMIA)
04	July 2018	Add MorphoWave® Compact support

Table of Contents

Section 1 : **Introduction**

Purpose

The purpose of this document is to describe how to configure the format of the different frames supported by the MorphoAccess® SIGMA Family & MorphoWave® Compact terminals.

MorphoAccess® SIGMA Family & MorphoWave® Compact terminals support some pre-defined Wiegand frames:

-        Standard 26 bits

-        Apollo 44 bits

-        Nothern 34 bits

-        Nothern 34 bits No Parity

-        Ademco 34 bits

-        HID Corporate 1000

-        HID 37 bits

-        Autodetect

-        Wiegand Last Format Input

If necessary, the end user can define and uses its own formats. MorphoAccess® SIGMA Family & MorphoWave® Compact terminals can handle up to 8 custom formats.

Section 2 : **Custom formats**

Parameters name

The custom frames format can be configured using following terminal's parameters:

- wiegand. custom_format_slot0

- wiegand. custom_format_slot1

- wiegand. custom_format_slot2

- wiegand. custom_format_slot3

- wiegand. custom_format_slot4

- wiegand. custom_format_slot5

- wiegand. custom_format_slot6

- wiegand. custom_format_slot7

Those parameters represent a binary buffer.

Format

Custom formats are using the same format for configuration.

The binary format is as following:

Fields	Length (in bytes)
Format ID *Shall be unique per custom format (valid value is from 0 to 7)* *Little endian*	4
Format Name *ASCII string (NULL character included)* *Null right padding*	32
Wiegand frame bit length *Shall be <= 512*	4
ID field start bit (first bit is 0th bit) *Little endian*	4
ID field length in bits *Shall be <= Wiegand frame bit length* *Cannot be 0* *Little endian*	4
"Heart Beat" ID *ID sent in case of Heart Beat wiegand frame* *Multiple of 4 bytes, 0 left padding* *Every blocs of 4 bytes little endian* <u>Example:</u> ID to send = 0x1234567890ABCDEF123 (73 bits) Heart beat ID is 000001234567890ABCDEF123	N x 4, min. 4
RFU *Shall be set to 0*	8

Number of User fields *Little endian*	4
1st user field name *ASCII string (NULL character included)* *Null right padding*	16
1st user field start bit (first bit is 0th bit) *Little endian*	4
1st user field length in bits *Little endian*	4
1st user field "Success" ID *ID sent in case of success of the control* *Multiple of 4 bytes, 0 left padding* *Every blocs of 4 bytes little endian*	N x 4, min. 4
1st user field "Failure" ID *ID sent in case of failure of the control* *Multiple of 4 bytes, 0 left padding* *Every blocs of 4 bytes little endian*	N x 4, min. 4
RFU *Shall be set to 0*	8
…	
N user field name *ASCII string (NULL character included)* *Null right padding*	16
N user field start bit (first bit is 0th bit) *Little endian*	4
N user field length in bits *Little endian*	4

N user field "Success" ID *ID sent in case of success of the control* *Multiple of 4 bytes, 0 left padding* *Every blocs of 4 bytes little endian*	N x 4, min. 4
N user field "Failure" ID *ID sent in case of failure of the control* *Multiple of 4 bytes, 0 left padding* *Every blocs of 4 bytes little endian*	N x 4, min. 4
RFU *Shall be set to 0*	8
Number of parity bits *Little endian*	4
1st parity bit start bit *Little endian*	4
1st parity bit type *0: Odd, 1: Even* *Little endian*	4
1st parity bit mask *Mask defining which bits to use in parity computation.* *All bits set to 1 mean include those bits from wiegand frame in parity computation* *Multiple of 4 bytes, 0 left padding* *Every blocs of 4 bytes little endian* *Length depends on Wiegand frame length*	N x 4, min. 4
RFU *Shall be set to 0*	4
…	

N parity bit start bit *Little endian*	4
N parity bit type *0: Odd, 1: Even* *Little endian*	4
N parity bit mask *Mask defining which bits to use in parity computation.* *All bits set to 1 mean include those bits from wiegand frame in parity computation* *Multiple of 4 bytes, 0 left padding* *Every blocs of 4 bytes little endian* *Length depends on Wiegand frame length*	N x 4, min. 4
RFU *Shall be set to 0*	4

Section 3 : **Predefined formats**

Introduction

All predefined formats of wiegand use zero as site code value. To specify different values of site code, a custom wiegand format shall be created and used.

Standard 26 bits

Overall Length (in bits)	26
Length of ID (in bits)	16
ID range	0-65535
Length of Site Code (in bits)	8
Site Code range	0-255
ID start bit	9
ID end bit	24
Site Code Start bit	1
Site Code End Bit	8
Are there any bits not covered by ID and SC?	N/A
Parity bits (comma separated)	0,25
Parity bit 0	Computed over bits 1-12 and is even
Parity bit 25	Computed over bits 13-24 and is odd

Frame bits	Data	Parity bit 1	Parity bit 2
0	Parity bit 1		
1		E	
2		E	
3		E	
4		E	
5		E	
6		E	
7		E	
8		E	
9		E	
10		E	
11		E	
12		E	
13			O
14			O
15			O
16			O
17			O
18			O
19			O
20			O
21			O
22			O
23			O
24			O
25	Parity bit 2		

Site code

ID

Apollo 44 bits

Overall Length (in bits)	44
Length of ID (in bits)	16
ID range	0-65535
Length of Site Code (in bits)	14
Site Code range	0-16383
ID start bit	21
ID end bit	36
Site Code Start bit	7
Site Code End Bit	20
Are there any bits not covered by ID and SC?	Yes. Blocks of bits 1-6 and 37-42 are filled with 0
Parity bits (comma separated)	0,43
Parity bit 0	Computed over bits 1-21 and is even
Parity bit 43	Computed over bits 22-42 and is odd

Frame bits	Data	Parity bit 1	Parity bit 2
0	Parity bit 1		
1	0	E	
2	0	E	
3	0	E	
4	0	E	
5	0	E	
6	0	E	
7		E	
8		E	
9		E	
10		E	
11		E	
12		E	
13		E	
14		E	
15		E	
16		E	
17		E	
18		E	
19		E	
20		E	
21		E	
22			O
23			O
24			O
25			O
26			O

Site code

ID

27			O
28			O
29			O
30			O
31			O
32			O
33			O
34			O
35			O
36			O
37	0		O
38	0		O
39	0		O
40	0		O
41	0		O
42	0		O
43	Parity bit 2		

Nothern 34 bits

Overall Length (in bits)	34
Length of ID (in bits)	16
ID range	0-65535
Length of Site Code (in bits)	16
Site Code range	0-65535
ID start bit	17
ID end bit	32
Site Code Start bit	1
Site Code End Bit	16
Are there any bits not covered by ID and SC?	N/A
Parity bits (comma separated)	0,33
Parity bit 0	Always 0
Parity bit 33	Computed over all bits (0-32) and is even

Frame bits	Data	Parity bit 1	Parity bit 2
0	Parity bit 1	0	E
1			E
2			E
3			E
4			E
5			E
6			E
7			E
8			E
9			E
10			E
11			E
12			E
13			E
14			E
15			E
16			E
17			E
18			E
19			E
20			E
21			E
22			E
23			E
24			E

Site code

ID
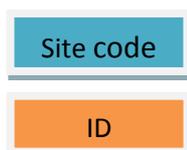
25			E
26			E
27			E
28			E
29			E
30			E
31			E
32			E
33	Parity bit 2		

Nothern 34 bits No Parity

Same as Northen 34 bits but contains no parity information.

Ademco 34 bits

Overall Length (in bits)	34
Length of ID (in bits)	20
ID range	0-1048575
Length of Site Code (in bits)	12
Site Code range	0-4095
ID start bit	13
ID end bit	32
Site Code Start bit	1
Site Code End Bit	12
Are there any bits not covered by ID and SC?	N/A
Parity bits (comma separated)	0,33
Parity bit 0	Computed over bits 1-18 and is odd
Parity bit 33	Computed over bits 15-32 and is even

Frame bits	Data	Parity bit 1	Parity bit 2
0	Parity bit 1		
1		O	
2		O	
3		O	
4		O	
5		O	
6		O	
7		O	
8		O	
9		O	
10		O	
11		O	
12		O	
13		O	
14		O	
15		O	E
16		O	E
17		O	E
18		O	E
19			E
20			E
21			E
22			E
23			E

Site code
ID

HID Corporate 1000

|	
Overall Length (in bits)	35
Length of ID (in bits)	20
ID range	0-1048575
Length of Site Code (in bits)	12
Site Code range	0-4095
ID start bit	14
ID end bit	33
Site Code Start bit	2
Site Code End Bit	13
Are there any bits not covered by ID and SC	N/A
Parity bits (comma separated)	0,1,34
Parity bit 0 (Computed third)	Computed over bits 1-34 and is odd
Parity bit 1 (Computed first)	Computed over bits : 2, 3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 18, 20, 21, 23, 24, 26, 27, 29, 30, 32, 33 and is even
Parity bit 34 (Computed second)	Computed over bits : 1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23, 25, 26, 28, 29, 31, 32 and is odd

Frame bits	Data	Parity bit 1	Parity bit 2	Parity bit 3
0	Parity bit 1			
1	Parity bit 2	O		O
2		O	E	O
3		O	E	
4		O		O
5		O	E	O
6		O	E	
7		O		O
8		O	E	O
9		O	E	
10		O		O
11		O	E	O
12		O	E	
13		O		O
14		O	E	O
15		O	E	
16		O		O
17		O	E	O
18		O	E	
19		O		O
20		O	E	O

Site code

ID

21		O	E	
22		O		O
23		O	E	O
24		O	E	
25		O		O
26		O	E	O
27		O	E	
28		O		O
29		O	E	O
30		O	E	
31		O		O
32		O	E	O
33		O	E	
34	Parity bit 3	O		

HID 37 bits

Overall Length (in bits)	37
Length of ID (in bits)	24
ID range	0-16777215
Length of Site Code (in bits)	11
Site Code range	0-2047
ID start bit	12
ID end bit	35
Site Code Start bit	1
Site Code End Bit	11
Are there any bits not covered by ID and SC?	N/A
Parity bits (comma separated)	0,36
Parity bit 0	Computed over bits 1-18 and is even
Parity bit 36	Computed over bits 18-35 and is odd

Frame bits	Data	Parity bit 1	Parity bit 2
0	Parity bit 1		
1		E	
2		E	
3		E	
4		E	
5		E	
6		E	
7		E	
8		E	
9		E	
10		E	
11		E	
12		E	
13		E	
14		E	
15		E	
16		E	
17		E	
18		E	O
19			O
20			O
21			O
22			O
23			O

Site code

ID
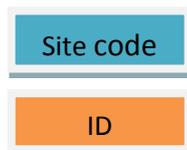
24			O
25			O
26			O
27			O
28			O
29			O
30			O
31			O
32			O
33			O
34			O
35			O
36	Parity bit 2		

130 bits Tamper wiegand format

Overall Length (in bits)	130
Length of ID (in bits)	128
ID start bit	1
ID end bit	128
Parity bits (comma separated)	0,129
Parity bit 0	Computed over bits 1-64 and is even
Parity bit 129	Computed over bits 65-128 and is odd

ID is filled with terminal serial number. Since serial number is alpha numeric, ASCII value of serial number is used to generate ID.

Example:

Serial Number of MA SIGMA terminal = 1310SMS0000011

1- ASCII Value of each digit(in hexadecimal):

ASCII character	1	3	1	0	S	M	S	0	0	0	0	0	1	1
Hex value	31	33	31	30	53	4D	53	30	30	30	30	30	31	31

2- Complete 130 bits wiegand frame (with two parity bits at first and last bits):

0000000000000000000110001001100110011000100110000010100110100110101010011001 10000001100000011000000110000001100000011000100110001100011

3- 128 bit ID (removing first and last bit from above 130 bit frame):

0000000000000000011000100110011001100010011000001010011010011010101001100110000001100000011000000110000001100000011000100110001

4- Grouping every 8 bits (1 byte) from 128 bit to get ASCII characters

8 Bits Group	Hex value	ASCII character
00000000	00	
00000000	00	
00110001	31	1

00110011	33	3
00110001	31	1
00110000	30	0
01010011	53	S
01001101	4D	M
01010011	53	S
00110000	30	0
00110000	30	0
00110000	30	0
00110000	30	0
00110000	30	0
00110001	31	1
00110001	31	1

Autodetect

This feature allows reading of different length wiegand frame in input. When this feature is enabled, Terminal detects input Wiegand Frame length and matches it with any of predefined custom slots length. If match is found, it will parse input wiegand frame for processing according to detected custom slot format otherwise input Wiegand is discarded.

Autodetect can be configured for following Wiegand input channel like external port, prox port and HID card number from iClass smartcard.

To enable Autodetect for following key value shall be configured to *"wiegand_fmt_autodetect" (18)* in any of following input configuration.

- wiegand.external_port_input_format

- wiegand.prox_port_input_format

- sc.HID_card_number_format.

Notes:

1. It is mandatory to configure at least one wiegand custom slot.

2. If there are multiple formats defined in custom slots with same length then first matching slot will be considered.

3. When autodetect is enabled at least one custom slot is maintained if tried to delete from list of 0 to 7.

Example :

Below table defines 4 different custom slots in MA terminal done by site.

Slot No.	Wiegand frame Length	Id start bit	Id length	Parity
0	37	10	16	Even,Even
1	45	16	25	Even,odd
2	22	6	11	Even,odd
3	37	8	13	Even,odd

If Autodetect is enabled and on external port terminal receives 37-bit Wiegand frame as per below

1001011101111101010111010111010111111

As input Wiegand frame length is 37 bits so it matched with custom-slot 0, so this received Wiegand Frame is parsed to extract with id length = 16 and it start position in 10th bit.

Wiegand Last Format Input

This configuration help to use last received input Wiegand Frame format to be applied on output Wiegand format before sending out control result Wiegand Frame. This enable different Wiegand format to go out as it is received in input with modification of control result.

To enable *Wiegand Last Format Input* following key value shall be set to *"wiegand_fmt_last_fmt_input " (18)* in any of following output event configuration.
- wiegand.event_verify_fail.
- wiegand.event_verify_pass.

In case the input format is the autodetect, the last custom Wiegand format selected for input will be will be as output format.

For authentication operation intiated from other trigger like a keyboard ID or distant command and if wiegand format "wiegand_fmt_last_fmt_input" is selected for the event "wiegand.event_verify_fail" or "wiegand.event_verify_pass" in that case custom slot 0 format will be used.

Notes:
It is mandatory to define custom format slot 0 to enable "wiegand_fmt_last_fmt_input".

Example :

Below table define 4 different custom slot configuration in MA terminal done by site and "wiegand_fmt_last_fmt_input" is selected for the event "wiegand.event_verify_fail" or "wiegand.event_verify_pass".

Slot No.	Wiegand frame Length	Id start bit	Id length	Parity
0	37	10	16	Even,Even
1	45	16	25	Even,odd
2	22	6	11	Even,odd
3	37	8	13	Even,odd

Input Wiegand frame is 1001011101111010101110101111010111111 and length detected is 37-bit.

If control result is OK in that wiegand event is generated and send using last recived (slot-0) Wiegand Frame in output.

Output Wiegand Frame is 1001011101111010101110101111010111111.

Section 4 : **Site Code Propagation**

Introduction

Site code propagation allow use of site code from received wiegand input frame and to be applied in output wiegand frame.

Configuration key "wiegand.site_code_propagation" is used to enable site code propagation:

0 - Disable site code propagation (Default)

1 - Enable site code propagation

Site code is stored internally each time from a Wiegand input frame, or Prox card. This site code will be used as output site code in wiegand frame for the "wiegand.event_verify_fail" and "wiegand.event_verify_pass" formats.

In case of authentication from keyboard ID or distant command, the defined site code from format custom slot 0 will be used.

Notes:

It is mandatory to define custom format slot 0 to enable the site code propagation.

When different Wiegand format are used for input and output ("wiegand_fmt_last_fmt_input" is disabled) and if site code length is different in configured input/output format in that case output site code will be truncated or padded with 0 before sending out Wiegand Frame.

Annex 1 :   **Support**

Troubleshooting

Customer service

IDEMIA

SAV Terminaux Biométriques

Boulevard Lénine - BP428

76805 Saint Etienne du Rouvray

FRANCE

Phone: +33 2 35 64 53 52

Hotline

IDEMIA

Support Terminaux Biométriques

18, Chaussée Jules César

95520 Osny

FRANCE

support.bioterminals@idemia.com

Phone: + 33 1 30 20 30 40

(9H00am to 5H30pm French Time, Monday to Friday)

http://www.biometric-terminals.com/

A login and password are required to access the full site content. If an administrator doesn't have one, please send us an email to the address above to request one.

Contact by email is preferred.

Registered Office:

IDEMIA

11, boulevard Gallieni

92130 Issy-les-Moulineaux – France

www.idemia.com