

# MorphoAccess<sup>®</sup> SIGMA Family

## Remote Message Specification



COPYRIGHT© 2018 IDEMIA

Osny, France

# WARNING

---

COPYRIGHT© 2018 IDEMIA. All rights reserved.

Information in this document is subject to change without notice and do not represent a commitment on the part of IDEMIA. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying or recording, for any purpose without the express written permission of IDEMIA.

This legend is applicable to all pages of this document.

This manual makes reference to names and products that are trademarks of their respective owners.

# Revision History

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

| Version   | Date       | Description   |
|-----------|------------|---|
| <b>01</b> | 2016-11-02 | New document's reference based on 2015_2000014298_v2<br><br>Features added or updated: <ul style="list-style-type: none"><li>- Enhanced MMI answer</li><li>- RS485/RS422 - Message format</li><li>- RS485: Annex 3</li><li>- RS422: Annex 4</li></ul> |
| <b>02</b> | 2016-11-24 | Features updated: <ul style="list-style-type: none"><li>- Enhanced MMI answer:<br/>Relay activation duration unit change</li></ul>  |
| <b>03</b> | 2017-04-12 | Replace 5G Series with SIGMA Family<br>Add the MorphoAccess® SIGMA Extreme Series product<br>Change the contact information   |
| <b>04</b> | 2017-06-21 | Add note about text display duration in Enhanced MMI  |
| <b>05</b> | 2017-12-27 | Update company name (IDEMIA)  |
| <b>06</b> | 2017-04-17 | Features updated: <ul style="list-style-type: none"><li>- Section 2: IP Remote messages</li><li>- Section 5: Extended remote message format and real time mode</li></ul>  |

# Table of Contents

|  |           |
|--|-----------|
| <b>REVISION HISTORY .....</b>                                | <b>3</b>  |
| <b>SECTION 1 : INTRODUCTION .....</b>                        | <b>9</b>  |
| <b><i>MorphoAccess® SIGMA Family Terminal.....</i></b>       | <b>10</b> |
| <b><i>Scope of the document.....</i></b>                     | <b>11</b> |
| <b><i>Remote Message Overview .....</i></b>                  | <b>14</b> |
| <i>Supported Protocols.....</i>                              | 15        |
| <b>SECTION 2 : TCP/SSL/UDP REMOTE MESSAGES .....</b>         | <b>17</b> |
| <b><i>IP Remote Messages .....</i></b>                       | <b>18</b> |
| <i>Presentation.....</i>                                     | 18        |
| <i>Activation of Remote Message .....</i>                    | 19        |
| <i>Controller Capabilities.....</i>                          | 21        |
| <i>Default Controller Definition.....</i>                    | 21        |
| <i>Alternative Controller Definition.....</i>                | 22        |
| <i>Remote Message Sending Flow .....</i>                     | 23        |
| <b>SECTION 3 : SERIAL REMOTE MESSAGES .....</b>              | <b>26</b> |
| <b><i>Serial Remote Messages .....</i></b>                   | <b>27</b> |
| <i>RS485 Presentation.....</i>                               | 27        |
| <i>RS422 Presentation.....</i>                               | 27        |
| <i>Activation.....</i>                                       | 28        |
| <i>Terminal Identifier.....</i>                              | 28        |
| <i>Data Format.....</i>                                      | 29        |
| <i>RS485/RS422 - Message format .....</i>                    | 29        |
| <i>Serial Link Settings .....</i>                            | 30        |
| <b>SECTION 4 : WIEGAND / DATACLOCK REMOTE MESSAGES .....</b> | <b>31</b> |
| <b><i>Wiegand Remote Messages .....</i></b>                  | <b>32</b> |
| <i>Presentation.....</i>                                     | 32        |
| <i>External Port Wiegand Protocol selection .....</i>        | 32        |
| <i>Activation.....</i>                                       | 32        |
| <b><i>Setting up Wiegand Interface.....</i></b>              | <b>33</b> |
| <i>Duress finger event Wiegand frame definition.....</i>     | 33        |
| <i>User control failure event frame definition.....</i>      | 34        |
| <i>User control success event frame definition .....</i>     | 36        |
| <i>Tamper event Wiegand frame definition.....</i>            | 38        |

|   |           |
|---|-----------|
| <i>Wiegand Frame Timing</i> .....                                     | 39        |
| <b>DataClock Remote Messages</b> .....                                | <b>40</b> |
| <i>Presentation</i> .....   | 40        |
| <i>External Port Dataclock Protocol selection</i> .....               | 40        |
| <i>Activation</i> .....   | 40        |
| <b>DataClock: failure messages</b> .....                              | <b>41</b> |
| <i>Presentation</i> .....   | 41        |
| <i>Software Configuration</i> .....                                   | 41        |
| <b>SECTION 5 : REMOTE MESSAGE FRAMES</b> .....                        | <b>42</b> |
| <b>Message Format</b> .....   | <b>43</b> |
| <i>Basic format</i> .....   | 43        |
| <i>Extended format</i> .....  | 43        |
| <b>Control is OK</b> .....  | <b>45</b> |
| <i>Description</i> .....  | 45        |
| <i>Frame sent when control is OK</i> .....                            | 45        |
| <i>Example</i> .....  | 46        |
| <b>Control Failed</b> .....   | <b>47</b> |
| <i>Description</i> .....  | 47        |
| <i>Command: sent frame</i> .....                                      | 47        |
| <i>Examples</i> .....   | 48        |
| <b>Tamper Alarm sent by MorphoAccess® SIGMA Family terminal</b> ..... | <b>50</b> |
| <i>Description</i> .....  | 50        |
| <i>Sending</i> .....  | 50        |
| <i>Command: sent frame</i> .....                                      | 50        |
| <b>Internal log file full Message</b> .....                           | <b>51</b> |
| <i>Description</i> .....  | 51        |
| <i>Message sent</i> .....   | 51        |
| <b>Door opened for too long</b> .....                                 | <b>52</b> |
| <i>Description</i> .....  | 52        |
| <i>Message sent</i> .....   | 52        |
| <b>Forced door open</b> .....   | <b>53</b> |
| <i>Description</i> .....  | 53        |
| <i>Message sent</i> .....   | 53        |
| <b>Door closed after alarm</b> .....                                  | <b>54</b> |
| <i>Description</i> .....  | 54        |
| <i>Message sent</i> .....   | 54        |
| <b>Door unlocked</b> .....  | <b>55</b> |

|   |           |
|---|-----------|
| <i>Description</i> .....                          | 55        |
| <i>Message sent</i> .....                         | 55        |
| <b>Door locked back</b> .....                     | <b>56</b> |
| <i>Description</i> .....                          | 56        |
| <i>Message sent</i> .....                         | 56        |
| <b>Management menu login</b> .....                | <b>57</b> |
| <i>Description</i> .....                          | 57        |
| <i>Message sent</i> .....                         | 57        |
| <b>Management menu logout</b> .....               | <b>58</b> |
| <i>Description</i> .....                          | 58        |
| <i>Message sent</i> .....                         | 58        |
| <b>Database deleted</b> .....                     | <b>59</b> |
| <i>Description</i> .....                          | 59        |
| <i>Message sent</i> .....                         | 59        |
| <b>Enrolment completed</b> .....                  | <b>60</b> |
| <i>Description</i> .....                          | 60        |
| <i>Message sent</i> .....                         | 60        |
| <b>Deletion completed</b> .....                   | <b>61</b> |
| <i>Description</i> .....                          | 61        |
| <i>Message sent</i> .....                         | 61        |
| <b>User modification completed</b> .....          | <b>62</b> |
| <i>Description</i> .....                          | 62        |
| <i>Message sent</i> .....                         | 62        |
| <b>Contactless card encoded</b> .....             | <b>63</b> |
| <i>Description</i> .....                          | 63        |
| <i>Message sent</i> .....                         | 63        |
| <b>Contactless card reset</b> .....               | <b>64</b> |
| <i>Description</i> .....                          | 64        |
| <i>Message sent</i> .....                         | 64        |
| <b>Settings changed</b> .....                     | <b>65</b> |
| <i>Description</i> .....                          | 65        |
| <i>Message sent</i> .....                         | 65        |
| <b>Contactless card security keys reset</b> ..... | <b>66</b> |
| <i>Description</i> .....                          | 66        |
| <i>Message sent</i> .....                         | 66        |
| <b>Firmware upgrade</b> .....                     | <b>67</b> |
| <i>Description</i> .....                          | 67        |

|  |           |
|--|-----------|
| <i>Message sent</i> .....  | 67        |
| <b>Job code check failure</b> .....                                    | <b>68</b> |
| <i>Description</i> .....   | 68        |
| <i>Message sent</i> .....  | 68        |
| <b>Terminal boot completed</b> .....                                   | <b>69</b> |
| <i>Description</i> .....   | 69        |
| <i>Message sent</i> .....  | 69        |
| <b>Add user</b> .....  | <b>70</b> |
| <i>Description</i> .....   | 70        |
| <i>Message sent</i> .....  | 70        |
| <b>Reboot initiated</b> .....  | <b>71</b> |
| <i>Description</i> .....   | 71        |
| <i>Message sent</i> .....  | 71        |
| <b>Duress finger detected</b> .....                                    | <b>72</b> |
| <i>Description</i> .....   | 72        |
| <i>Message sent</i> .....  | 72        |
| <b>Security policy changed</b> .....                                   | <b>73</b> |
| <i>Description</i> .....   | 73        |
| <i>Message sent</i> .....  | 73        |
| <b>Basic MMI Answer (Returned by the Controller)</b> .....             | <b>74</b> |
| <i>Description</i> .....   | 74        |
| <i>Command: frame returned by the controller to the terminal</i> ..... | 74        |
| <b>Enhanced MMI Answer (Returned by the Controller)</b> .....          | <b>75</b> |
| <i>Description</i> .....   | 75        |
| <i>Command: frame returned by the controller to the terminal</i> ..... | 75        |
| <i>Command without any action to perform</i> .....                     | 77        |
| <i>Example</i> .....   | 77        |
| <br>   |           |
| <b>ANNEX 1 : WIEGAND DATA FORMAT</b> .....                             | <b>78</b> |
| <b>ANNEX 2 : ISO 7811/2 - 1995 - TRACK 2 DATACLOCK FORMAT</b> .....    | <b>80</b> |
| <b>ANNEX 3 : RS485 PROTOCOL</b> .....                                  | <b>83</b> |
| <b>ANNEX 4 : RS422 PROTOCOL</b> .....                                  | <b>86</b> |
| <b>ANNEX 5 : BIBLIOGRAPHY</b> .....                                    | <b>92</b> |
| <b>ANNEX 6 : GLOSSARY, ACRONYMS AND ABBREVIATION</b> .....             | <b>95</b> |
| <b>ANNEX 7 : SUPPORT</b> .....   | <b>99</b> |

# Table of Figures

---

|  |    |
|--|----|
| Figure 1: Remote Message Overview .....  | 14 |
| Figure 2: Ethernet or Wi-Fi™, UDP, TCP or SSL protocols.....                     | 18 |
| Figure 3: Remote Message Flow when response from controller is not required..... | 23 |
| Figure 4: Remote Message Flow when response from controller is required .....    | 24 |
| Figure 5: Remote Message Flow for stored events in Real Time mode .....          | 25 |
| Figure 6: Serial port, RS485 protocol .....                                      | 27 |
| Figure 7 : Serial port, RS422 protocol .....                                     | 27 |
| Figure 8: Wiegand frame format .....   | 79 |
| Figure 9: Data Clock signals.....  | 82 |
| Figure 10: Other Data Clock signals .....  | 82 |
| Figure 11 : RS485 frame processing.....  | 85 |
| Figure 12: RS422 frame processing.....   | 88 |
| Figure 13: RS422 typical frame workflow .....                                    | 90 |
| Figure 14: RS422 transmission error.....   | 90 |
| Figure 15: RS422 transmission timeout error.....                                 | 91 |



## MorphoAccess® SIGMA Family Terminal

Congratulations for choosing a MorphoAccess® SIGMA Family Automatic Fingerprint Recognition Terminal.

MorphoAccess® SIGMA Family provides an innovative and effective solution for access control applications using Fingerprint Verification or/and Identification.

Among a range of alternative biometric technologies, the use of finger imaging has significant advantages: each finger constitutes an unalterable physical signature, developed before birth and preserved until death. Unlike DNA, a finger image is unique for each individual - even identical twins.

The MorphoAccess® SIGMA Family terminals integrate Morpho image processing and feature matching algorithms. This technology is based on lessons learned during 25 years of experience in the field of biometric identification and the creation of literally millions of individual fingerprint identification records.

Designed for physical access control applications, MorphoAccess® SIGMA Family terminals feature a compact, attractive design, coupled with high reliability and security. These 5<sup>th</sup> generation terminals are both robust and easy to use for a variety of applications, including office, headquarters and administrative building security, as well as protection of external access points.

## Scope of the document

This guide relates to the remote message configuration of MorphoAccess® SIGMA Family Terminal in MA5G mode.

| Terminal Series             | Terminal Name                             | Biometrics | Contactless smartcard reader |                             |       | Outdoor |
|-----------------------------|---|------------|------------------------------|-----------------------------|-------|---------|
|                             |   |            | iCLASS®                      | MIFARE®<br>DESFire®<br>NFC® | Prox® |         |
| MorphoAccess®<br>SIGMA      | MorphoAccess®<br>Sigma                    | ✓          |                              |                             |       |         |
|                             | MorphoAccess®<br>Sigma WR                 | ✓          |                              |                             |       | ✓       |
|                             | MorphoAccess®<br>Sigma iCLASS®            | ✓          | ✓                            |                             |       |         |
|                             | MorphoAccess®<br>Sigma iCLASS®<br>WR      | ✓          | ✓                            |                             |       | ✓       |
|                             | MorphoAccess®<br>Sigma Multi              | ✓          |                              | ✓                           |       |         |
|                             | MorphoAccess®<br>Sigma Multi WR           | ✓          |                              | ✓                           |       | ✓       |
|                             | MorphoAccess®<br>Sigma Prox               | ✓          |                              |                             | ✓     |         |
|                             | MorphoAccess®<br>Sigma Prox WR            | ✓          |                              |                             | ✓     | ✓       |
| MorphoAccess®<br>SIGMA Lite | MorphoAccess®<br>SIGMA Lite WR            | ✓          |                              |                             |       | ✓       |
|                             | MorphoAccess®<br>SIGMA Lite<br>iCLASS® WR | ✓          | ✓                            |                             |       | ✓       |

|  |   |   |   |   |   |   |
|--|---|---|---|---|---|---|
|  | MorphoAccess®<br>SIGMA Lite Multi<br>WR       | ✓ |   | ✓ |   | ✓ |
|  | MorphoAccess®<br>SIGMA Lite Prox®<br>WR       | ✓ |   |   | ✓ | ✓ |
| MorphoAccess®<br>SIGMA Lite+             | MorphoAccess®<br>SIGMA Lite+ WR               | ✓ |   |   |   | ✓ |
|  | MorphoAccess®<br>SIGMA Lite+<br>iCLASS® WR    | ✓ | ✓ |   |   | ✓ |
|  | MorphoAccess®<br>SIGMA Lite+ Multi<br>WR      | ✓ |   | ✓ |   | ✓ |
|  | MorphoAccess®<br>SIGMA Lite+<br>Prox® WR      | ✓ |   |   | ✓ | ✓ |
| MorphoAccess®<br>SIGMA Extreme<br>Series | MorphoAccess®<br>SIGMA Extreme<br>iCLASS®     | ✓ | ✓ |   |   | ✓ |
|  | MorphoAccess®<br>SIGMA Extreme<br>FFD iCLASS® | ✓ | ✓ |   |   | ✓ |
|  | MorphoAccess®<br>SIGMA Extreme<br>Multi       | ✓ |   | ✓ |   | ✓ |
|  | MorphoAccess®<br>SIGMA Extreme<br>FFD Multi   | ✓ |   | ✓ |   | ✓ |
|  | MorphoAccess®<br>SIGMA Extreme<br>Prox®       | ✓ |   |   | ✓ | ✓ |
|  | MorphoAccess®<br>SIGMA Extreme<br>FFD Prox®   | ✓ |   |   | ✓ | ✓ |

**NOTE:** *\*Here, WR indicates terminal is Weather Resistant.*

**NOTE:** *\*Here, FFD indicates terminal supports hardware fake finger detection feature*

## Remote Message Overview

The MorphoAccess® SIGMA Family terminal can send status messages in real time to a controller by different ways and through different protocols. This information, termed as **Remote Messages** in this document, can be used, for instance, to display on an external screen the result of a biometric operation, the name or the ID of the person identified, etc. depending on the role of the controller in the system.

This document describes various solutions offered by the MorphoAccess® SIGMA Family terminal to dialog with a controller, and how to make use of them.

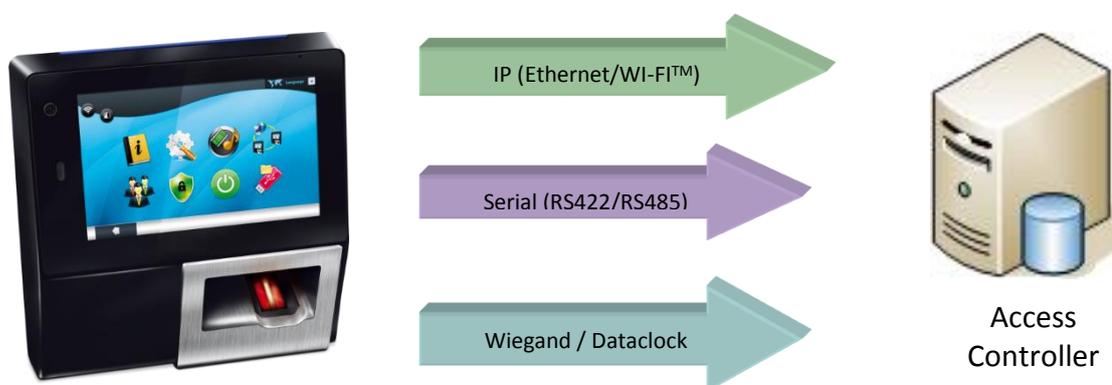


Figure 1: Remote Message Overview

## Supported Protocols

The MorphoAccess® SIGMA Family terminal can send messages about the biometric operations performed by the terminal to a controller through the following protocols:

- Wiegand
- Dataclock
- Serial (RS485/422)
- IP

The format of the messages frames differs according to the protocol chosen. Note that Wiegand/Dataclock messages can be also enriched with extended error ID. This feature is described in section [User control failure event frame](#) and [Dataclock: failure messages](#).

| Protocol Used  | Verification OK  | Verification KO  |
|--|--|--|
| <b>Wiegand</b><br><i>The terminal acts as a magnetic badge reader.</i>   | The ID of the identified user is sent. The frame format can be configured. | Nothing is sent if frame format for failure event is not configured.<br><br>Or The ID of the identified user or zero ID is sent if the frame format for failure event is configured. |
| <b>Dataclock</b><br><i>The terminal acts as a magnetic badge reader.</i> | The ID (ISO2) of the identified user is sent.                              | Nothing is sent.<br>Or numerical ID describing the cause of the failure.   |
| <b>RS485/422</b>   | Identification result is sent.   | The biometric check result (failure) is sent.  |
| <b>UDP</b>   | Complete identification result is sent.                                    | The biometric check result (failure) is sent.  |
| <b>TCP</b>   |  |  |
| <b>SSL</b>   |  |  |

The subsequent chapters explain how to activate the sending of the remote messages for each channel available: Wiegand/Dataclock, Serial, and IP.

### *In case of Wiegand – Dataclock*

These outputs are multiplexed. It means that only one of them can be enabled: (TR- / D1) and (TR+ / D0).

The configuration of each protocol requires modifying some parameters. If you do not know how to perform such operation, please refer to the MorphoAccess® SIGMA Family **Parameters Guide**.

Parameters can be changed using remote management commands.



## IP Remote Messages

### Presentation

In remote mode the terminal acts as a **client** and sends TCP, UDP or SSL information to the PC that is a **server**.

It is possible to choose an alternative controller in case of default controller failure.

The messages can be sent both in UDP and TCP/SSL protocols. Standard UDP protocol configuration is kept for compatibility with previous firmware versions and as simple way for logging events. The TCP/SSL protocol configuration offers enhanced features (wait for response from controllers).

The sent messages formats are the same in UDP, TCP and SSL: only the protocol is different.

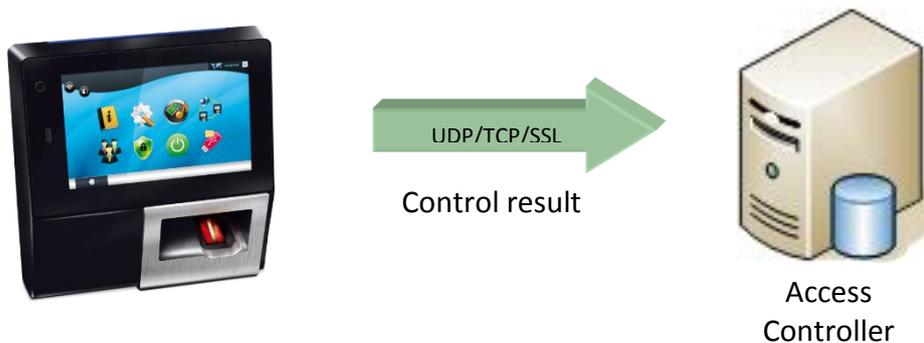


Figure 2: Ethernet or Wi-Fi™, UDP, TCP or SSL protocols

In this chapter “Controller” is used for “host TCP, UDP or SSL server”. It could be a standard computer or an integrated system.

## Activation of Remote Message

Remote messages over IP channel can be configured using key `'remote_msg_conf.send_ethernet_state'`, set value 1 to enable.

The message format over IP channel can be configured using key `'remote_msg_conf.format'`, to either Basic or Extended format (refer to [Message format](#) for format description). When using Extended format, it is also possible to enable real-time mode:

- When controller is not reachable, the event is stored in terminal memory.
- When a new event occurs, the previous memorized events are sent prior to the new event.
- When the memorized events memory is full (5000 events), the oldest event is lost when storing the new event.

Feedback waiting over IP channel is disabled by default, to enable controller feedback use key `'remote_msg_conf.feedback_interface'` and set the value to 1.

Terminal also allows alternate controller so the configuration key is available to configure the use of alternate controller, use the key `'remote_msg_ip_conf.mode'` to configure which controller the remote message shall be sent.

|  |   |
|--|---|
| <code>remote_msg_conf.send_ethernet_state</code> |   |
| 0 or 1   | Configuration of remote message sending. (default is 0, 1 to enable)  |
| <code>remote_msg_conf.format</code>              |   |
| 0 to 2   | Configuration of remote message format. (since firmware 4.5)<br>0: Basic format (default)<br>1: Extended format<br>2: Extended format with real time mode |
| <code>remote_msg_conf.feedback_interface</code>  |   |
| 0 to 3   | Configuration for feedback waiting. (default is 0, 1 to enable feedback waiting over IP channel)  |
| <code>remote_msg_ip_conf.mode</code>             |   |

|        |   |
|--------|---|
| 0 to 2 | Which controller shall be used to send remote message                           |
|        | 0 – Send remote message to controller 1 only                                    |
|        | 1 – Send remote message to both controllers                                     |
|        | 2 – Send remote message to controller 1 if fails then only send to controller 2 |

Particular event which user wants to send through remote message interface should be enabled by calling thrift API of MA5G named as “**events\_set\_config**”. Reference for particular thrift API can be found in documentation of Distant Command named as “**MA5G\_Distant\_Commands**”.

## Controller Capabilities

The MorphoAccess® SIGMA Family terminal is able to act as a controller (default working state: it can allow or deny access depending on authentication or identification process result).

In case of using enhanced TCP/SSL features, the MorphoAccess® SIGMA Family terminal behaviour can be selected by changing the value of key `'remote_msg_ip_conf.host_on_no_response'`.

- **0:** if connection to controller failed, or an error response is received from controller, then the MorphoAccess® SIGMA Family terminal cannot act as controller. Even in case the biometric control result was “Access Allowed”, the connection error makes the terminal deny the access.
- **1 (default value):** if connection to controller failed, or an error response is received from controller, then the MorphoAccess® SIGMA Family terminal can act as controller. If the biometric control result was “Access Allowed”, the access is granted.

| remote_msg_ip_conf.host_on_no_response |  |
|--|--|
| 0 or 1                                 | Let the MorphoAccess® SIGMA Family terminal act as controller or not (default is 1, for true). |

## Default Controller Definition

| remote_msg_ip_conf.host_1_ip       |   |
|------------------------------------|---|
| IP Address                         | It defines the controller’s IP on the network.  |
| remote_msg_ip_conf.host_1_port     |   |
| 11020                              | It defines the controller’s port on the network<br><b>Note:</b> Check that your firewall is correctly configured.                                   |
| remote_msg_ip_conf.host_1_timeout  |   |
| 2000                               | Timeout used for connection, reading and writing data (at TCP/UDP level) to/from the remote controller. In multiple of 10ms (2000 means 20 seconds) |
| remote_msg_ip_conf.host_1_protocol |   |
| 0                                  | It defines the controller’s protocol for communication<br>(0-TCP, 1-UDP, 2-SSL)   |

## Alternative Controller Definition

It is possible to define an alternative controller on the network. Message sending over alternate controller can be defined by key '*remote\_msg\_ip\_conf.mode*'.

Alternative controller will be disabled by default in case the value of IP and port in both controllers are same.

|   |   |
|---|---|
| <i>remote_msg_ip_conf.host_2_ip</i>       |   |
| IP Address                                | It defines the controller's IP on the network.  |
| <i>remote_msg_ip_conf.host_2_port</i>     |   |
| 11021                                     | It defines the controller's port on the network<br><b>Note:</b> Check that your firewall is correctly configured.                                   |
| <i>remote_msg_ip_conf.host_2_timeout</i>  |   |
| 2000                                      | Timeout used for connection, reading and writing data (at TCP/UDP level) to/from the remote controller. In multiple of 10ms (2000 means 20 seconds) |
| <i>remote_msg_ip_conf.host_2_protocol</i> |   |
| 0   | It defines the controller's protocol for communication<br>(0-TCP, 1-UDP, 2-SSL)   |

## Remote Message Sending Flow

If the response from the controller is not required (all the cases except Control successful event), then below workflow is established:

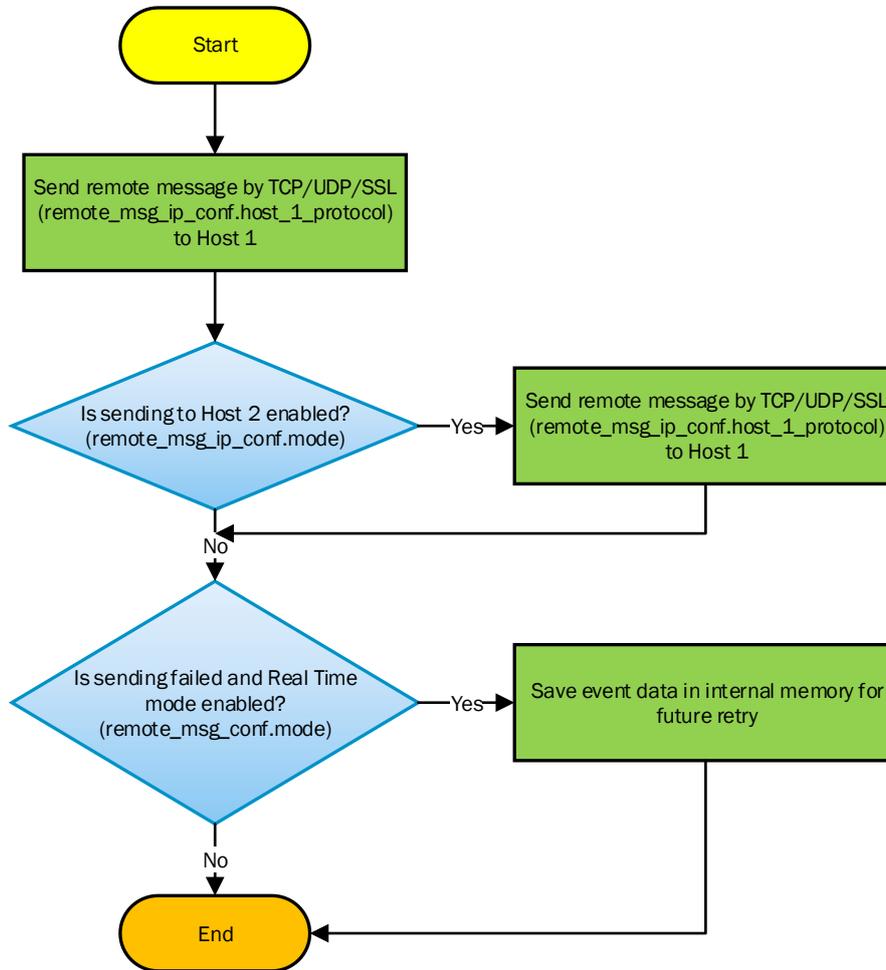


Figure 3: Remote Message Flow when response from controller is not required

If the response from the controller is required (Control successful event) , then below workflow is established:

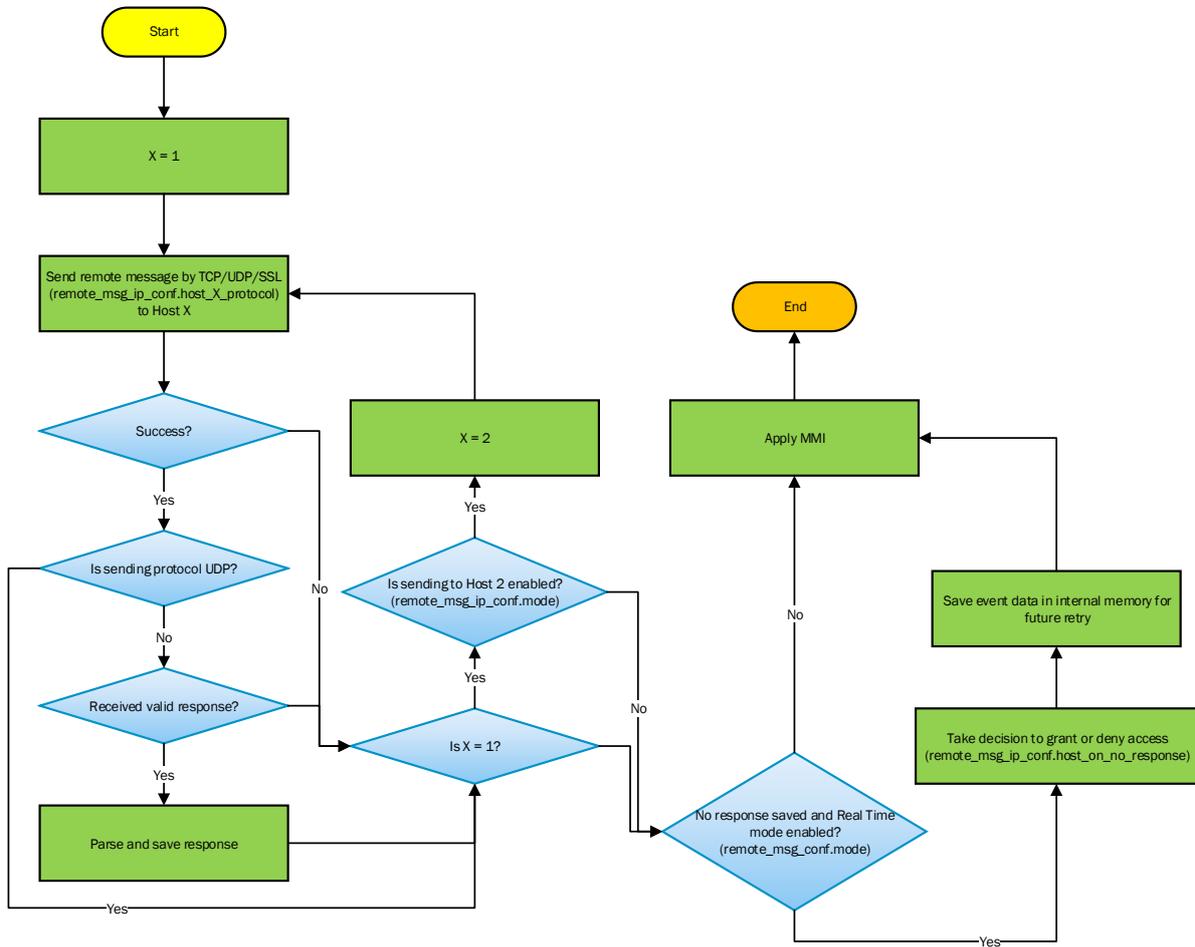


Figure 4: Remote Message Flow when response from controller is required

When Remote message over IP is configured in Real Time mode, events stored in internal memory (events that were not sent successfully at the time they occurred) are sent to the controller(s) prior to sending current event. The following workflow is established for all stored events that are [sent without response from controller](#):

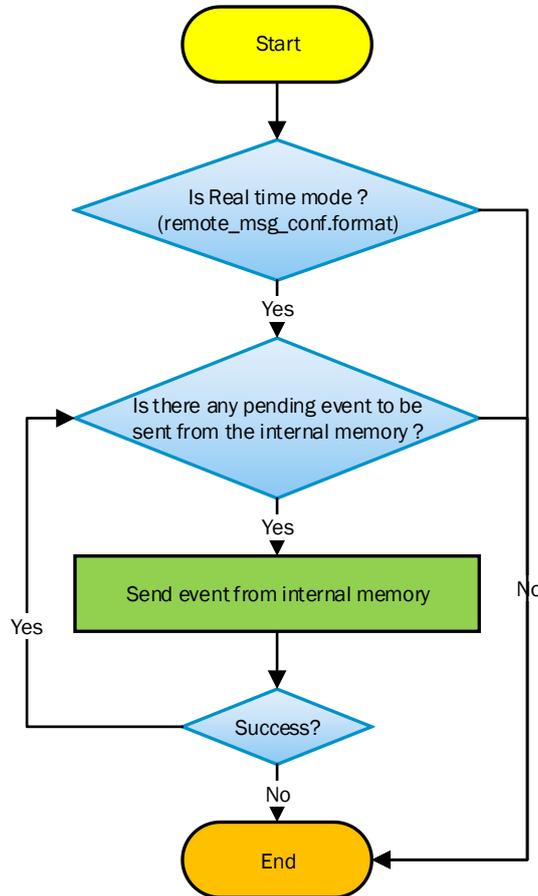


Figure 5: Remote Message Flow for stored events in Real Time mode



## Serial Remote Messages

This feature is available since Firmware MA3.1.x.

### RS485 Presentation

The MorphoAccess® SIGMA Family terminal can also send information through the serial link using RS485 protocol. Several terminals can be connected on the same serial link.

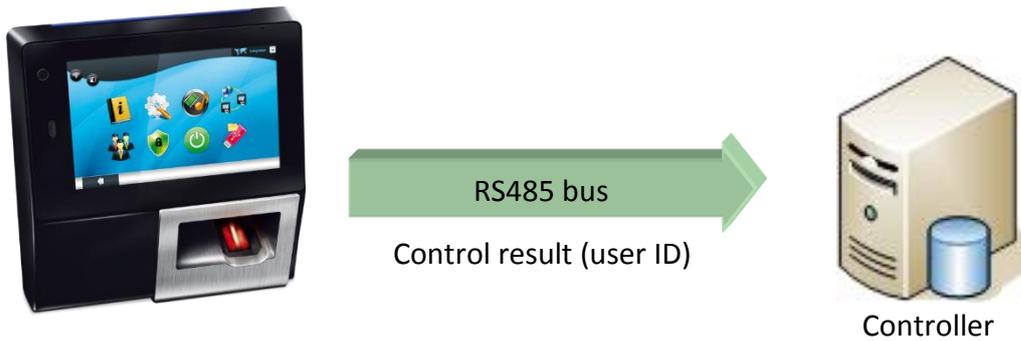


Figure 6: Serial port, RS485 protocol

### RS422 Presentation

The MorphoAccess® SIGMA series terminal can send information through the serial link using RS422 protocol.

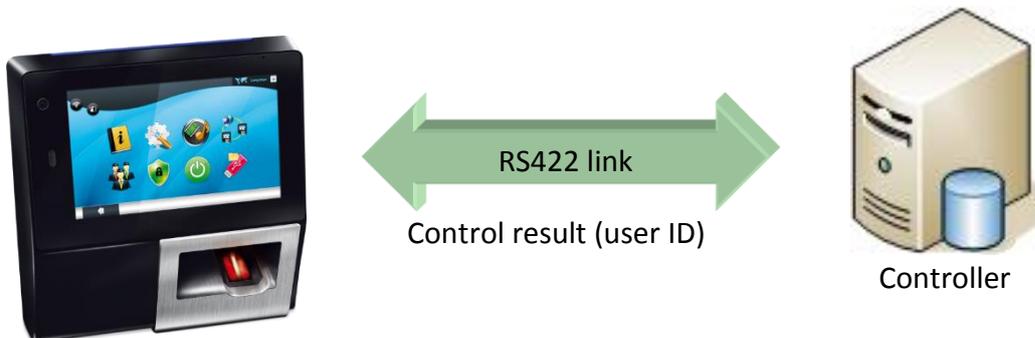


Figure 7 : Serial port, RS422 protocol

## Activation

The following key activates or deactivates sending messages on serial link.

### *remote\_msg\_conf.send\_serial\_state*

|   |                        |
|---|------------------------|
| 0 | Messages are not sent. |
| 1 | Messages are sent.     |

Serial channel protocol format can be configured by using the following key.

### *remote\_msg\_serial\_conf.format*

|     |   |
|-----|---|
| 422 | RS422 protocol (For MorphoAccess® SIGMA and SIGMA Extreme terminal) |
| 485 | RS485 protocol (default)  |

## Terminal Identifier

Terminal identifier parameter defines the address of MorphoAccess® terminal over the RS485 bus. It can be configured using the following key.

### *Serial\_communication.net\_id*

|       |  |
|-------|--|
| 0-255 | Net ID is required to identify a terminal in a network (serially connected) of several terminals |
|-------|--|

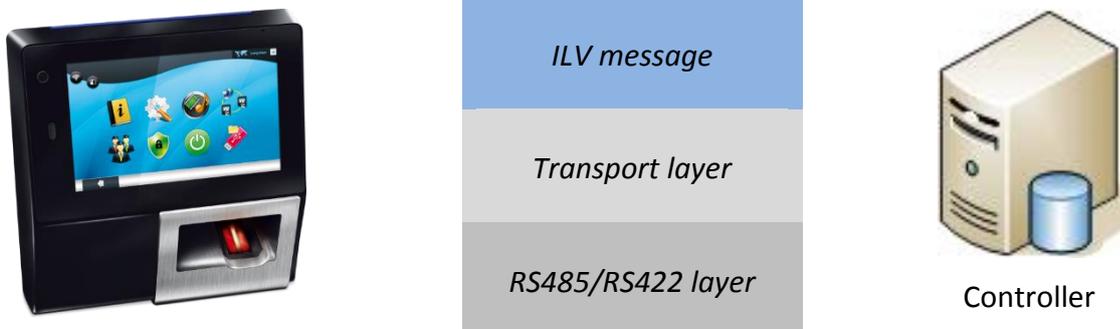
- Terminal identifier values such as DLE [0x1B], XON [0x11] or XOFF [0x13], are forbidden.

The RS485 protocol is described in [Annex 3: RS485 Protocol](#).

The RS422 protocol is described in [Annex 4: RS422 Protocol](#).

## Data Format

Remote information is sent on the RS485 or RS422 (MorphoAccess® 500 Series only) serial channel. The transport layer ensures the transmission and ensures frame acknowledgement.



It is possible to re-send the frame, in case of a failure. When using a RS485 serial layer, the terminal uses an anti-collision protocol to prevent collisions on the link.

- The RS485 protocol is described in [Annex 3 : RS485 Protocol](#).
- The RS422 protocol is described in [Annex 4: RS422 Protocol](#).

## RS485/RS422 - Message format

Messages sent though RS485 or RS422 have the following format termed as the **ILV** format.

| <i>ILV messages</i> |   |              |
|---------------------|---|--------------|
| Identifier          | Length  | Value        |
| 1 byte              | 2 bytes                                       | Length bytes |
| Message identifier  | Message data length<br>(little endian format) | Message data |

The application data has three fields:

- **Identifier** called **I**: this is the identifier of the command,
- **Length** called **L**: this is the length of the Value field in bytes,
- **Value** called **V**: this is the data or parameters.

The length of data structure is variable. Message data will depend on the biometric control result.

The transport layers are described in [Annex 3: RS485 Protocol](#) and in [Annex 4: RS422 Protocol](#).

## Serial Link Settings

It is possible to set the baud rate, data bits, stop bits size and parity type. To configure these serial parameters use thrift command *'terminal\_set\_configuration'*.

Refer document “**MorphoAccess® SIGMA Family Thrift Generic Commands Guide**”.



## Wiegand Remote Messages

### Presentation

The payload data encapsulated in a Wiegand frame is either the ID of the person identified, in case of successful user control operation. In case of unsuccessful user control operation, Wiegand frame with available ID or zero ID is sent if Wiegand frame is configured for failure event. If no Wiegand frame is configured for failure event then nothing is sent for failure.

### External Port Wiegand Protocol selection

A configuration entry allows selecting the Wiegand as external port protocol.

| wiegand.external_port_output_type |  |
|-----------------------------------|--|
| 0                                 | Wiegand, Port type is Wiegand (Default). |
| 1                                 | Dataclock, Port type is DataClock.       |

### Activation

A configuration entry allows enabling the external port output state.

| wiegand.external_port_output_status |   |
|-------------------------------------|---|
| 0                                   | Disabled. Wiegand/Dataclock frame not sent (Default). |
| 1                                   | Enabled. Wiegand/Dataclock frame is sent.             |
| -1                                  | Reserved (but internally works as Enabled)            |

## Setting up Wiegand Interface

When set up to communicate with Wiegand protocol, the MorphoAccess® SIGMA Family terminal can handle multiple data format for various output events (e.g. user control success, user control failure, duress, tamper).

MorphoAccess® SIGMA Family terminal supports preconfigured standard Wiegand formats along with user defined custom Wiegand format. Various standard Wiegand and custom Wiegand format configuration are described in **MorphoAccess Sigma - Wiegand AN.docx**.

Different Wiegand frame format can be defined for both inputs as well as various output events.

Wiegand frame timings (pulse width and pulse interval) are customizable.

### Duress finger event Wiegand frame definition

A configuration entry defines duress finger event Wiegand frame

| wiegand.event_duress_finger |   |
|-----------------------------|---|
| 0                           | No format, Wiegand frame not send.  |
| 1                           | Reverse Wiegand format. Verify or Identify success event Wiegand frame is reversed.                             |
| 10                          | Custom Wiegand format slot 0, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 11                          | Custom Wiegand format slot 1, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 12                          | Custom Wiegand format slot 2, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 13                          | Custom Wiegand format slot 3, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 14                          | Custom Wiegand format slot 4, this value can only be set if format exists in corresponding custom Wiegand slot. |

|    |   |
|----|---|
| 15 | Custom Wiegand format slot 5, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 16 | Custom Wiegand format slot 6, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 17 | Custom Wiegand format slot 7, this value can only be set if format exists in corresponding custom Wiegand slot. |

### User control failure event frame definition

A configuration entry (wiegand.event\_identify\_fail) defines Wiegand frame for finger biometry triggered user control operation failure event

A configuration entry (wiegand.event\_verify\_fail) defines Wiegand frame for non finger biometry triggered user control operation failure event

| wiegand.event_identify_fail |   |
|-----------------------------|---|
| wiegand.event_verify_fail   |   |
| -1                          | No format, Wiegand frame not send.  |
| 0                           | standard 26 bit format  |
| 1                           | Apollo 44 bit format.   |
| 2                           | Northen 34 bit format   |
| 3                           | Northen 34 bit format (no parity)   |
| 4                           | Ademco 34 bit format.   |
| 5                           | HID corporate 1000 format.  |
| 6                           | HID 37 bit format.  |
| 10                          | Custom Wiegand format slot 0, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 11                          | Custom Wiegand format slot 1, this value can only be set if format exists in corresponding custom Wiegand slot. |

|    |   |
|----|---|
| 12 | Custom Wiegand format slot 2, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 13 | Custom Wiegand format slot 3, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 14 | Custom Wiegand format slot 4, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 15 | Custom Wiegand format slot 5, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 16 | Custom Wiegand format slot 6, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 17 | Custom Wiegand format slot 7, this value can only be set if format exists in corresponding custom Wiegand slot. |

## User control success event frame definition

A configuration entry (wiegand.event\_identify\_pass) defines Wiegand frame for finger biometry triggered user control operation successful event

A configuration entry (wiegand.event\_verify\_pass) defines Wiegand frame for non finger biometry triggered user control operation successful event

| wiegand.event_identify_pass |   |
|-----------------------------|---|
| wiegand.event_verify_pass   |   |
| -1                          | No specific format<br>i.e. use input format defined by <code>wiegand.external_port_input_format</code> .        |
| 0                           | standard 26 bit format  |
| 1                           | Apollo 44 bit format.   |
| 2                           | Northen 34 bit format   |
| 3                           | Northen 34 bit format (no parity)   |
| 4                           | Ademco 34 bit format.   |
| 5                           | HID corporate 1000 format.  |
| 6                           | HID 37 bit format.  |
| 10                          | Custom Wiegand format slot 0, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 11                          | Custom Wiegand format slot 1, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 12                          | Custom Wiegand format slot 2, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 13                          | Custom Wiegand format slot 3, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 14                          | Custom Wiegand format slot 4, this value can only be set if format exists in corresponding custom Wiegand slot. |

|    |   |
|----|---|
| 15 | Custom Wiegand format slot 5, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 16 | Custom Wiegand format slot 6, this value can only be set if format exists in corresponding custom Wiegand slot. |
| 17 | Custom Wiegand format slot 7, this value can only be set if format exists in corresponding custom Wiegand slot. |

## Tamper event Wiegand frame definition

A configuration entry defines Tamper event Wiegand frame

| wiegand.event_tamper |  |
|----------------------|--|
| 0                    | No format, Wiegand frame not send.   |
| 1                    | Generate 130 bit Wiegand string.<br>E <128 bit serial number> O<br>Contains 128 bit device serial number and two parity bits.<br>Even parity (bit 0) will be calculated from bits 1-64<br>Odd parity (bit 129) will be calculated from bits 65-128 |
| 10                   | Custom Wiegand format slot 0, this value can only be set if format exists in corresponding custom Wiegand slot.  |
| 11                   | Custom Wiegand format slot 1, this value can only be set if format exists in corresponding custom Wiegand slot.  |
| 12                   | Custom Wiegand format slot 2, this value can only be set if format exists in corresponding custom Wiegand slot.  |
| 13                   | Custom Wiegand format slot 3, this value can only be set if format exists in corresponding custom Wiegand slot.  |
| 14                   | Custom Wiegand format slot 4, this value can only be set if format exists in corresponding custom Wiegand slot.  |
| 15                   | Custom Wiegand format slot 5, this value can only be set if format exists in corresponding custom Wiegand slot.  |
| 16                   | Custom Wiegand format slot 6, this value can only be set if format exists in corresponding custom Wiegand slot.  |
| 17                   | Custom Wiegand format slot 7, this value can only be set if format exists in corresponding custom Wiegand slot.  |

## Wiegand Frame Timing

Wiegand data format is described in section [Appendix 1 – Wiegand Data Format](#).

A configuration entry allows pulse width configuration.

| wiegand.pulse_width |   |
|---------------------|---|
| 60                  | Width of single Wiegand pulse<br>Range is from 20 to 100 microseconds |

A configuration entry allows complete pulse internal configuration (includes pulse width).

| wiegand.pulse_interval |  |
|------------------------|--|
| 3000                   | Pulse Interval of single Wiegand pulse (Idle time = interval - width)<br>Range is from 200 to 20000 microseconds |

## DataClock Remote Messages

### Presentation

The payload data encapsulated in a Dataclock frame is either the ID of the person identified, in case of successful identification, or an ID describing the reason of the identification failure (if the Failure ID are activated, see chapter [Dataclock: failure messages](#)).

Dataclock frame content is described in section [Appendix 3 - ISO 7811/2 -1995 - Track 2 Dataclock Format](#).

### External Port Dataclock Protocol selection

A configuration entry allows selecting the Dataclock as external port protocol.

| wiegand.external_port_output_type |  |
|-----------------------------------|--|
| 0                                 | Wiegand, Port type is Wiegand (Default). |
| 1                                 | Dataclock, Port type is DataClock.       |

### Activation

A configuration entry allows enabling the external port output state.

| wiegand.external_port_output_status |   |
|-------------------------------------|---|
| 0                                   | Disabled. Wiegand/Dataclock frame not sent (Default). |
| 1                                   | Enabled. Wiegand/Dataclock frame is sent.             |

## DataClock: failure messages

### Presentation

*Failure ID* option allows sending extended error codes through the Dataclock layer. You can activate this option and associate any numeric value for each existing failure case.

**NOTE:** *This feature has no impact on the IP and Serial (RS422/RS485) remote messages.*

**NOTE:** *The administrator has to check that the identifier is not already stored in the database.*

### Software Configuration

To configure dataclock ID, use thrift command `'events_set_config'`.

Refer document 'TODO' for thrift command help.



## Section 5 : Remote Message Frames

## Message Format

All the exchanged messages over IP between the MorphoAccess® SIGMA Family terminal and the remote controller have the same structure. These messages are called **ILV** (Identifier, Length, and Value).

| <i>ILV messages</i>       |   |                     |
|---------------------------|---|---------------------|
| <b>Identifier</b>         | <b>Length</b>   | <b>Value</b>        |
| 1 byte                    | 2 bytes   | Length bytes        |
| <i>Message identifier</i> | <i>Message data length<br/>(little endian format)</i> | <i>Message data</i> |

The application data has three fields:

- **Identifier** called **I**: this is the identifier of the command,
- **Length** called **L**: this is the length of the Value field in bytes,
- **Value** called **V**: this is the data or parameters.

This data structure is variable. Its length is variable. Message data depends on the type of message and the format configured (refer to [Activation of Remote Message](#)).

### Basic format

Messages in Basic format contain basic information about the event.

|                    |                         |         |
|--------------------|-------------------------|---------|
| Identifier value   | Event identifier        | 1 byte  |
| Length value       | 0x0000 + n              | 2 bytes |
| Value (Parameters) | Event basic information | n bytes |

### Extended format

Messages in Extended format contain additional information for all events.

|                    |  |          |
|--------------------|--|----------|
| Identifier value   | Event identifier                           | 1 byte   |
| Length value       | 0x0000 + 32 + n                            | 2 bytes  |
| Value (Parameters) | Extended data                              | 32 bytes |
|                    | - <i>Terminal Serial Number (14 bytes)</i> |          |

|   |                                |
|---|--------------------------------|
|  | - Event date & time (17 bytes) |
|   | - Event status (1 byte)        |
| <i>Event basic information</i>  |                                |
|   | <i>n bytes</i>                 |

### *Terminal Serial Number*

---

14 bytes terminal Serial Number in ASCII. For example “1800ABC0123456”.

### *Event date & time*

---

17 bytes ASCII buffer formatted as follows: **DD/MM/YY hh:mm:ss**.

### *Event Status*

---

**[0x00]** : Real-time event (event just occurred)

**[0x01]** : Offline event, user control success. Controller was not reachable when the event initially occurred and user was granted access based on controls and terminal configuration.

**[0x02]** : Offline event, user control failed. Controller was not reachable when the event initially occurred and user was denied access based on controls and terminal configuration.

**[0xFF]** : Offline event, for events other than “Control OK (0x00)”

## Control is OK

### Description

This frame is sent to the controller when the user is recognized by terminal.

### Frame sent when control is OK

|                    |  |          |
|--------------------|--|----------|
| Identifier value   | <b>0x00</b> : User ID is sent in ASCII format and control succeeded. | 1 byte   |
| Length value       | $0x0000 + L + 1 + 17$  | 2 bytes  |
| Value (Parameters) | User ID  | L bytes  |
|                    | <i>Attendance Status</i>   | 1 byte   |
|                    | <i>MA date and hour</i>  | 17 bytes |

#### *User ID*

User identifier in ASCII. "94066" for example.

#### *Attendance Status*

This status is always sent in Extended format, and sent in Basic format only when T&A is enabled.

This is an ASCII character defining the transaction performed :

0x49 = ' I ' : IN                      0x69 = ' i ' : IN DUTY

0x4F = ' O ' : OUT                    0x6F = ' o ' : OUT DUTY

If Extended Time & Attendance is activated and the pressed key is associated to the "user defined" function, below HEX code of the key is sent:

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| F1   | F2   | F3   | F4   | F5   | F6   | F7   | F8   |
| 0x01 | 0x02 | 0x03 | 0x04 | 0x05 | 0x06 | 0x07 | 0x08 |
| F9   | F10  | F11  | F12  | F13  | F14  | F15  | F16  |
| 0x09 | 0x0A | 0x0B | 0x0C | 0x0D | 0x0E | 0x0F | 0x10 |

*MA date and hour*

This data is not sent in Extended format as it is already part of ‘Extended data’, and is sent in Basic format only when T&A is enabled.

A 17 bytes ASCII buffer formatted as follows: “DD/MM/YY hh:mm:ss”

**Example**

This frame means user “528610” has been recognized.

In Basic format:

| Byte 1 | Byte 2      | Byte 3 | Byte 4                    | Byte 5 | Byte 6 | Byte 7 | Byte 8 | Byte 9 |
|--------|-------------|--------|---------------------------|--------|--------|--------|--------|--------|
| 0x00   | 0x06        | 0x00   | 0x35                      | 0x32   | 0x38   | 0x36   | 0x31   | 0x30   |
| OK     | L = 6 bytes |        | User identifier: “528610” |        |        |        |        |        |

In Extended format:

| Byte 1                            | Byte 2       | Byte 3  | Byte 4                                   | Byte 5  | Byte 6                              | Byte 7  | Byte 8    | Byte 9  |
|-----------------------------------|--------------|---------|--|---------|-------------------------------------|---------|-----------|---------|
| 0x00                              | 0x27         | 0x00    | 0x31                                     | 0x38    | 0x30                                | 0x30    | 0x41      | 0x42    |
| OK                                | L = 39 bytes |         | Terminal Serial Number: “1800ABC0123456” |         |                                     |         |           |         |
| Byte 10                           | Byte 11      | Byte 12 | Byte 13                                  | Byte 14 | Byte 15                             | Byte 16 | Byte 17   | Byte 18 |
| 0x43                              | 0x30         | 0x31    | 0x32                                     | 0x33    | 0x34                                | 0x35    | 0x36      | 0x32    |
|                                   |              |         |  |         |                                     |         |           |         |
| Byte 19                           | Byte 20      | Byte 21 | Byte 22                                  | Byte 23 | Byte 24                             | Byte 25 | Byte 26   | Byte 27 |
| 0x30                              | 0x2F         | 0x31    | 0x30                                     | 0x2F    | 0x31                                | 0x37    | 0x20      | 0x30    |
| Date & Time : “20/10/17 07:23:00” |              |         |  |         |                                     |         |           |         |
| Byte 28                           | Byte 29      | Byte 30 | Byte 31                                  | Byte 32 | Byte 33                             | Byte 34 | Byte 35   | Byte 36 |
| 0x37                              | 0x3A         | 0x32    | 0x33                                     | 0x3A    | 0x30                                | 0x30    | 0x00      | 0x35    |
|                                   |              |         |  |         |                                     |         | Real time |         |
| Byte 37                           | Byte 38      | Byte 39 | Byte 40                                  | Byte 41 | Byte 42                             |         |           |         |
| 0x32                              | 0x38         | 0x36    | 0x31                                     | 0x30    | 0xFF                                |         |           |         |
| User identifier: “528610”         |              |         |  |         | Attendance status : ‘no key’ (0xFF) |         |           |         |

## Control Failed

### Description

This frame is sent to the controller when the control failed.

### Command: sent frame

|                    |   |                 |
|--------------------|---|-----------------|
| Identifier value   | <b>0x10</b> : User ID is sent in ASCII format and control failed. | 1 byte          |
| Length value       | 1+L   | 2 bytes         |
| Value (Parameters) | Biometric Error Code  | 1 byte          |
|                    | User ID (according to the configuration)                          | L bytes         |
|                    | <i>Attendance Status</i>  | <i>1 byte</i>   |
|                    | <i>MA date and hour</i>   | <i>17 bytes</i> |

#### *Biometric Error Code*

---

Failure: err\_bio\_control\_failed [**0x01**],

Timeout: err\_bio\_control\_timeout [**0x19**],

User not in base: err\_bio\_not\_in\_base [**0x12**],

User not in time: err\_bio\_not\_on\_time [**0x02**],

Invalid contactless smartcard: err\_bio\_inval\_card [**0x03**],

Fake finger detected: err\_bio\_fake\_finger\_detected [**0x30**],

Pin mismatch: err\_bio\_pin\_mismatch [**0x31**],

Temporal validity expired: err\_bio\_temporal\_val\_expired [**0x32**],

User not in white list: err\_bio\_user\_not\_in\_white\_lst [**0x33**],

Black listed card: err\_bio\_blk\_lst\_card [**0x34**],

Face not detected: err\_bio\_face\_not\_detected [**0x35**],

User rule check failure: err\_bio\_usr\_rule\_check\_failure [**0x36**],

Generic error: err\_bio\_ident\_error [**0xFF**],

### User ID

---

The user ID is sent if the MorphoAccess® SIGMA Family terminal works in authentication mode only.

### Attendance Status

---

This status is always sent in Extended format, and sent in Basic format only when T&A is enabled.

This is an ASCII character defining the transaction performed :

0x49 = ' I ' : IN                      0x69 = ' i ' : IN DUTY

0x4F = ' O ' : OUT                    0x6F = ' o ' : OUT DUTY

If Extended Time & Attendance is activated and the pressed key is associated to the “user defined” function, below HEX code of the key is sent:

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| F1   | F2   | F3   | F4   | F5   | F6   | F7   | F8   |
| 0x01 | 0x02 | 0x03 | 0x04 | 0x05 | 0x06 | 0x07 | 0x08 |
| F9   | F10  | F11  | F12  | F13  | F14  | F15  | F16  |
| 0x09 | 0x0A | 0x0B | 0x0C | 0x0D | 0x0E | 0x0F | 0x10 |

### MA date and hour

---

This data is not sent in Extended format as it is already part of ‘Extended data’, and is sent in Basic format only when T&A is enabled.

A 17 bytes ASCII buffer formatted as follows: “DD/MM/YY hh:mm:ss”

## Examples

**Identification Mode:** this frame means that identification failed.

In Basic format:

| Byte 1 | Byte 2     | Byte 3 | Byte 4                |
|--------|------------|--------|-----------------------|
| 0x10   | 0x01       | 0x00   | 0x01                  |
| NOK    | L = 1 byte |        | Identification failed |

**Verification Mode:** this frame means the user “528610” presented its badge, but biometric verification failed.

In Basic format:

| Byte 1 | Byte 2      | Byte 3 | Byte 4              | Byte 5                    | Byte 6 | Byte 7 | Byte 8 | Byte 9 | Byte 10 |
|--------|-------------|--------|---------------------|---------------------------|--------|--------|--------|--------|---------|
| 0x10   | 0x07        | 0x00   | 0x01                | 0x35                      | 0x32   | 0x38   | 0x36   | 0x31   | 0x30    |
| NOK    | L = 7 bytes |        | Verification failed | User identifier: “528610” |        |        |        |        |         |

# Tamper Alarm sent by MorphoAccess® SIGMA Family terminal

## Description

This frame is sent to the controller when the tamper switch is activated. It can also play a sound alarm while sending the Alarm ID.

In case of intrusion, the alarm is emitted.

## Sending

The key "*tamper.alarm\_interval*" must be set to the duration between IP frames sending (for example, value 1500 (\*10ms) means that the alarm will be sent every 15 seconds.

The terminal does not wait for response.

## Command: sent frame

|                    |                             |         |
|--------------------|-----------------------------|---------|
| Identifier value   | <b>0xC1</b> : ILV_ALARM_ID. | 1 byte  |
| Length value       | 0x04                        | 2 bytes |
| Value (Parameters) | Alarm state                 | 4 bytes |

### *Alarm state*

---

0x00000000: an intrusion has been detected.

0x000000FF: end of intrusion.

## Internal log file full Message

### Description

Once the internal log (transaction log) file is full, the terminal can send a message each time a line is being written in the file.

### Message sent

|                    |  |         |
|--------------------|--|---------|
| Identifier value   | 0x02: LOGFULL_MESS_ID  | 1 byte  |
| Length value       | 0x01   | 2 bytes |
| Value (Parameters) | <i>Response needed [0x00 for MorphoAccess® SIGMA Family]</i> | 1 bytes |

#### *Response needed*

If set to 0x01, the terminal will wait for a response after having sent the message.

If set to 0x00, the terminal will end the communication after having sent the message.

**NOTE:** *Even if the response needed is set to one, no specific treatment is made on the response. And hence MorphoAccess® SIGMA Family terminal will never send value 0x01.*

## Door opened for too long

### Description

This frame is sent to the controller if SDAC mode is enabled and door opened time exceeds.

### Message sent

|                    |   |         |
|--------------------|---|---------|
| Identifier value   | <b>0x70</b> : <i>door_opened_for_too_long</i> | 1 byte  |
| Length value       | 0x00  | 2 bytes |
| Value (Parameters) | <i>None</i>                                   | 0 bytes |

## Forced door open

### Description

This frame is sent to the controller if SDAC mode is enabled and door is opened forcefully.

### Message sent

|                    |                                       |         |
|--------------------|---------------------------------------|---------|
| Identifier value   | <b>0x71</b> : <i>forced_door_open</i> | 1 byte  |
| Length value       | 0x00                                  | 2 bytes |
| Value (Parameters) | <i>None</i>                           | 0 bytes |

## Door closed after alarm

### Description

This frame is sent to the controller if SDAC mode is enabled and door is closed after it was opened forcefully.

### Message sent

|                    |  |         |
|--------------------|--|---------|
| Identifier value   | <b>0x72</b> : <i>door_closed_after_alarm</i> | 1 byte  |
| Length value       | 0x00   | 2 bytes |
| Value (Parameters) | <i>None</i>                                  | 0 bytes |

## Door unlocked

### Description

This frame is sent to the controller if SDAC mode is enabled and door is unlocked for timed override mode / the scheduled time.

### Message sent

|                    |                                    |         |
|--------------------|------------------------------------|---------|
| Identifier value   | <b>0x73</b> : <i>door_unlocked</i> | 1 byte  |
| Length value       | 0x00                               | 2 bytes |
| Value (Parameters) | <i>None</i>                        | 0 bytes |

## Door locked back

### Description

This frame is sent to the controller if SDAC mode is enabled and door is locked back after time override mode / the scheduled time is expired.

### Message sent

|                    |                                       |         |
|--------------------|---------------------------------------|---------|
| Identifier value   | <b>0x74</b> : <i>door_locked_back</i> | 1 byte  |
| Length value       | 0x00                                  | 2 bytes |
| Value (Parameters) | <i>None</i>                           | 0 bytes |

## Management menu login

### Description

This frame is sent to the controller if user logs in to the management (administration) menu.

### Message sent

|                    |  |         |
|--------------------|--|---------|
| Identifier value   | <b>0x75</b> : <i>management_menu_login</i> | 1 byte  |
| Length value       | 0x00                                       | 2 bytes |
| Value (Parameters) | <i>None</i>                                | 0 bytes |

## Management menu logout

### Description

This frame is sent to the controller if user logs out from the management (administration) menu or user automatically logs out due to timeout.

### Message sent

|                    |   |         |
|--------------------|---|---------|
| Identifier value   | <b>0x76</b> : <i>management_menu_logout</i> | 1 byte  |
| Length value       | 0x00  | 2 bytes |
| Value (Parameters) | <i>None</i>                                 | 0 bytes |

## Database deleted

### Description

This frame is sent to the controller when user database is deleted.

### Message sent

|                    |                                       |         |
|--------------------|---------------------------------------|---------|
| Identifier value   | <b>0x77</b> : <i>database_deleted</i> | 1 byte  |
| Length value       | 0x00                                  | 2 bytes |
| Value (Parameters) | <i>None</i>                           | 0 bytes |

## Enrolment completed

### Description

This frame is sent to the controller when enrolment process is completed.

### Message sent

|                    |  |         |
|--------------------|--|---------|
| Identifier value   | <b>0x78</b> : <i>enrolment_completed</i> | 1 byte  |
| Length value       | 0x00                                     | 2 bytes |
| Value (Parameters) | <i>None</i>                              | 0 bytes |

## Deletion completed

### Description

This frame is sent to the controller if user record is deleted from the user database.

### Message sent

|                    |   |         |
|--------------------|---|---------|
| Identifier value   | <b>0x79</b> : <i>deletion_completed</i> | 1 byte  |
| Length value       | 0x00                                    | 2 bytes |
| Value (Parameters) | <i>None</i>                             | 0 bytes |

## User modification completed

### Description

This frame is sent to the controller if user record is modified in the user database.

### Message sent

|                    |  |         |
|--------------------|--|---------|
| Identifier value   | <b>0x7A</b> : <i>user_modification_completed</i> | 1 byte  |
| Length value       | 0x00   | 2 bytes |
| Value (Parameters) | <i>None</i>                                      | 0 bytes |

## Contactless card encoded

### Description

This frame is sent to the controller when the contactless card encode process is completed.

### Message sent

|                    |   |         |
|--------------------|---|---------|
| Identifier value   | <b>0x7B</b> : <i>contactless_card_encoded</i> | 1 byte  |
| Length value       | 0x00  | 2 bytes |
| Value (Parameters) | <i>None</i>                                   | 0 bytes |

## Contactless card reset

### Description

This frame is sent to the controller when the contactless card reset process is completed.

### Message sent

|                    |   |         |
|--------------------|---|---------|
| Identifier value   | <b>0x7C</b> : <i>contactless_card_reset</i> | 1 byte  |
| Length value       | 0x00  | 2 bytes |
| Value (Parameters) | <i>None</i>                                 | 0 bytes |

## Settings changed

### Description

This frame is sent to the controller when any parameter key configuration is changed.

### Message sent

|                    |                                       |         |
|--------------------|---------------------------------------|---------|
| Identifier value   | <b>0x7D</b> : <i>settings_changed</i> | 1 byte  |
| Length value       | 0x00                                  | 2 bytes |
| Value (Parameters) | <i>None</i>                           | 0 bytes |

## Contactless card security keys reset

### Description

This frame is sent to the controller when contactless card security keys reset completed.

### Message sent

|                    |   |         |
|--------------------|---|---------|
| Identifier value   | <b>0x7E</b> : <i>contactless_card_security_keys_reset</i> | 1 byte  |
| Length value       | 0x00  | 2 bytes |
| Value (Parameters) | <i>None</i>   | 0 bytes |

## Firmware upgrade

### Description

This frame is sent to the controller when the firmware upgrade is started.

### Message sent

|                    |                                      |         |
|--------------------|--------------------------------------|---------|
| Identifier value   | <b>0x80:</b> <i>firmware_upgrade</i> | 1 byte  |
| Length value       | 0x00                                 | 2 bytes |
| Value (Parameters) | <i>None</i>                          | 0 bytes |

## Job code check failure

### Description

This frame is sent to the controller if the job code check is failed.

### Message sent

|                    |   |                 |
|--------------------|---|-----------------|
| Identifier value   | <b>0x81</b> : <i>job_code_check_failure</i> | 1 byte          |
| Length value       | 1+L   | 2 bytes         |
| Value (Parameters) | User ID (according to the configuration)    | L bytes         |
|                    | <i>Attendance Status (T&amp;A only)</i>     | <i>1 byte</i>   |
|                    | <i>MA date and hour (T&amp;A only)</i>      | <i>17 bytes</i> |

**Note:** Value part of this remote message is same as in case of "[control failed](#)"

## Terminal boot completed

### Description

This frame is sent to the controller when the terminal boot up is completed.

### Message sent

|                    |   |         |
|--------------------|---|---------|
| Identifier value   | <b>0x82:</b> <i>terminal_boot_completed</i> | 1 byte  |
| Length value       | 0x00  | 2 bytes |
| Value (Parameters) | <i>None</i>                                 | 0 bytes |

## Add user

### Description

This frame is sent to the controller when the user record is added to the user database.

### Message sent

|                    |                               |         |
|--------------------|-------------------------------|---------|
| Identifier value   | <b>0x83</b> : <i>add_user</i> | 1 byte  |
| Length value       | 0x00                          | 2 bytes |
| Value (Parameters) | <i>None</i>                   | 0 bytes |

## Reboot initiated

### Description

This frame is sent to the controller when terminal reboot is initiated.

### Message sent

|                    |                                       |         |
|--------------------|---------------------------------------|---------|
| Identifier value   | <b>0x84</b> : <i>reboot_initiated</i> | 1 byte  |
| Length value       | 0x00                                  | 2 bytes |
| Value (Parameters) | <i>None</i>                           | 0 bytes |

## Duress finger detected

### Description

This frame is sent to the controller when duress finger is detected while authentication / identification.

### Message sent

|                    |  |                 |
|--------------------|--|-----------------|
| Identifier value   | <b>0x85:</b> <i>duress_finger_detected</i> | 1 byte          |
| Length value       | 0x00                                       | 2 bytes         |
| Value (Parameters) | User ID (according to the configuration)   | L bytes         |
|                    | <i>Attendance Status (T&amp;A only)</i>    | <i>1 byte</i>   |
|                    | <i>MA date and hour (T&amp;A only)</i>     | <i>17 bytes</i> |

**Note:** Value part of this remote message is same as in case of "[control failed](#)"

## Security policy changed

### Description

This frame is sent to the controller when security related parameters configuration updated (i.e. Tamper enabled/disabled, secure management started/stopped, etc.).

### Message sent

|                    |  |         |
|--------------------|--|---------|
| Identifier value   | <b>0x86</b> : <i>security_policy_changed</i> | 1 byte  |
| Length value       | 0x00   | 2 bytes |
| Value (Parameters) | <i>None</i>                                  | 0 bytes |

## Basic MMI Answer (Returned by the Controller)

### Description

On reception of the frame “Control OK” and after checking the user is authorized. The controller returns this frame on the established socket (same connection).

### Command: frame returned by the controller to the terminal

|                    |                              |         |
|--------------------|------------------------------|---------|
| Identifier value   | <b>0x50</b> : Access status. | 1 byte  |
| Length value       | 1                            | 2 bytes |
| Value (Parameters) | Access Granted or Denied     | 1 byte  |

#### *Access Granted or Denied*

---

**0x00**: Access is granted: the MorphoAccess® SIGMA Family terminal will apply MMI as per the configuration.

**0xFF**: Access is denied: the MorphoAccess® SIGMA Family terminal will apply MMI as per the configuration.

**0x01 or any other value**: nothing happens. MorphoAccess® SIGMA Family terminal goes back in control mode.

## Enhanced MMI Answer (Returned by the Controller)

### Description

This feature is available since Firmware MA3.2.x.

On reception of the “Control OK” message, the controller is expected to return a frame as an answer. This frame describes the actions to be taken by the MorphoAccess® SIGMA Family terminals on the established socket (same connection):

- The duration of the sound signal to play
- The duration of the relay activation
- The light signal to emit with the status LED (only for MorphoAccess® SIGMA Lite)
- The text to display in the screen (for all MorphoAccess® SIGMA Family except MorphoAccess® SIGMA Lite)

### Command: frame returned by the controller to the terminal

|                    |                           |          |
|--------------------|---------------------------|----------|
| Identifier value   | 0x51: MMI Order.          | 1 byte   |
| Length value       | 0x0000 + 0x60             | 2 bytes  |
| Value (Parameters) | Sound activation          | 1 byte   |
|                    | Sound activation duration | 1 byte   |
|                    | Relay activation          | 1 byte   |
|                    | Relay activation duration | 1 byte   |
|                    | Display activation        | 1 byte   |
|                    | Text line 1               | 30 bytes |
|                    | Text line 2               | 30 bytes |
|                    | Text line 3               | 30 bytes |
|                    | Display duration          | 1 byte   |

**Sound Activation:** Configure the sound to play.

- 0 = No sound for MorphoAccess® SIGMA Family
- 1 = Access Denied sound for MorphoAccess® SIGMA Family
- 2 = Access Granted sound for MorphoAccess® SIGMA Family

**Sound Activation Duration:** Define the duration of the sound signal emission, in 10ms units (a value of 100 means 1 second). Range = [0, 100].

This is applicable for MorphoAccess® SIGMA Lite+ and MorphoAccess® SIGMA Lite terminals only. For MorphoAccess® SIGMA and SIGMA Extreme terminals, the entire sound file is played.

**Relay state:** The state applied to the Relay:

- 0 = no action
- 1 = trigger the relay

**Relay Duration:** Define the duration of the relay activation in 1s units. Range = [0, 10].

**Display Activation :** Defines the state of the Status LED and the display screen:

| Value | MorphoAccess® SIGMA LITE                     | MorphoAccess® SIGMA                          | MorphoAccess® SIGMA Lite+ | MorphoAccess® SIGMA Extreme |
|-------|--|--|---------------------------|-----------------------------|
| 0     | LED OFF (no specific light signal)           | display text on screen with GREY background  |                           |                             |
| 1     | RED LED (emission of a RED light signal)     | display text on screen with RED background   |                           |                             |
| 2     | GREEN LED (emission of a GREEN light signal) | display text on screen with GREEN background |                           |                             |

**Text Line 1, 2, 3:** NULL terminated strings containing the text to display on screen. Only the first 22 characters are used. The 23rd character (or a previous one) must be NULL.

**Display Duration:** Signifies the LED ON time for MorphoAccess® SIGMA Lite terminals. Whereas it signifies the duration for which the screen is ON for other terminals. This is in 100 ms units (value of 10 is equal to 1 second).

**Note.**

- In order to correctly display the text, the recommended text size is :
  - 22 characters for MorphoAccess® SIGMA
  - 14 characters for MorphoAccess® SIGMA Lite+
  - 15 characters for MorphoAccess® SIGMA Extreme
- SDAC mode must be enabled for Relay operation based on the feedback received. Existing SDAC configuration regarding relay duration (Door Unlock Duration) of the terminal, will not be considered if relay state and duration are provided in feedback.
- Whatever the display duration value, message disappears when user touch the screen

## Command without any action to perform

The following frame is returned by the controller to the MorphoAccess® terminal when no action is required:

|                  |                  |         |
|------------------|------------------|---------|
| Identifier value | 0x51: MMI Order. | 1 byte  |
| Length value     | 0x0000           | 2 bytes |

## Example

This frame means user cannot enter in the zone protected by the MorphoAccess® SIGMA Family terminal.

| Byte 1 | Byte 2     | Byte 3 | Byte 4           |
|--------|------------|--------|------------------|
| 0x50   | 0x01       | 0x00   | 0xFF             |
| AS     | L = 1 byte |        | Access is denied |

This frame means user can enter in the zone protected by the MorphoAccess® SIGMA Family terminal.

| Byte 1 | Byte 2     | Byte 3 | Byte 4            |
|--------|------------|--------|-------------------|
| 0x50   | 0x01       | 0x00   | 0x00              |
| AS     | L = 1 byte |        | Access is granted |

On reception of this frame no action will be taken on the MorphoAccess® SIGMA Family terminal.

| Byte 1 | Byte 2     | Byte 3 | Byte 4                                    |
|--------|------------|--------|---|
| 0x50   | 0x01       | 0x00   | 0xFF (Any value other than 0x00 and 0xFF) |
| AS     | L = 1 byte |        | No Action                                 |



The 26 bits of transmission consists of two parity bits and 24 code bits.

The 8 first code bits are encoding the facility code. This code identifies each MorphoAccess® in a network.

The 16 other bits are data bits.

The first bit transmitted is the first parity bit. It is even parity calculated over the first 12 bits.

The last bit transmitted is the second parity bit. It is odd parity bit calculated over the last 12 code bits.

|                     |                        |                |                    |
|---------------------|------------------------|----------------|--------------------|
| Even parity (1 bit) | Facility code (8 bits) | Data (16 bits) | Odd parity (1 bit) |
|---------------------|------------------------|----------------|--------------------|

Compliant with access control 26-Bit - Wiegand reader interface standard 03/1995

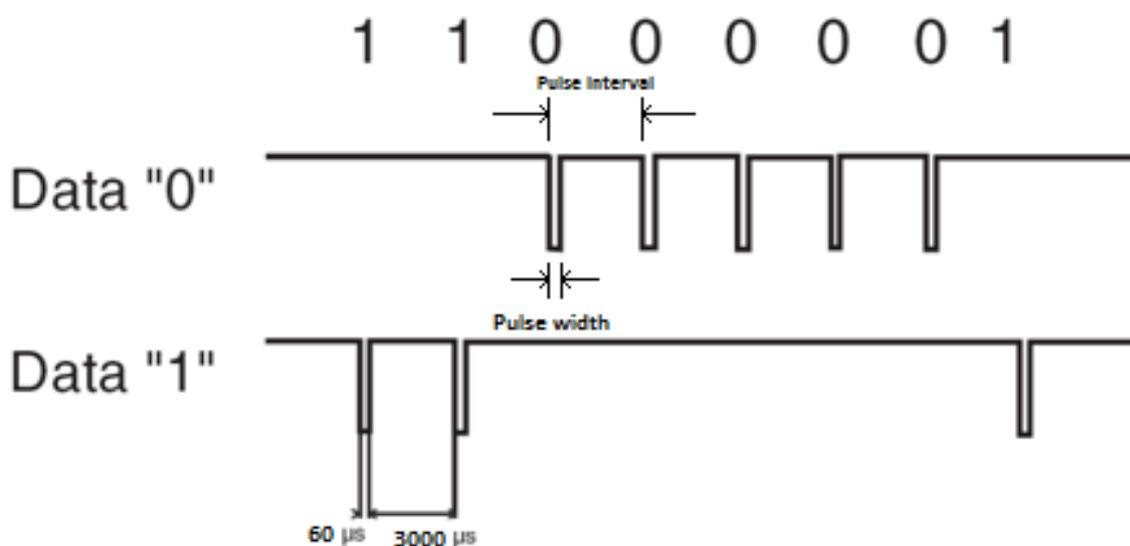


Figure 8: Wiegand frame format



## Data Encoding Table

| Value     | Bit pattern | Meaning                          |
|-----------|-------------|----------------------------------|
| 0         | 0 0 0 0-1   | "0"                              |
| 1         | 1 0 0 0-0   | "1"                              |
| 2         | 0 1 0 0-0   | "2"                              |
| 3         | 1 1 0 0-1   | "3"                              |
| 4         | 0 0 1 0-0   | "4"                              |
| 5         | 1 0 1 0-1   | "5"                              |
| 6         | 0 1 1 0-1   | "6"                              |
| 7         | 1 1 1 0-0   | "7"                              |
| 8         | 0 0 0 1-0   | "8"                              |
| 9         | 1 0 0 1-1   | "9"                              |
| 10 (Ahex) | 0 1 0 1-1   | Unused character                 |
| 11 (Bhex) | 1 1 0 1-0   | Start sentinel (start character) |
| 12 (Chex) | 0 0 1 1-1   | Unused character                 |
| 13 (Dhex) | 1 0 1 1-0   | Field separator                  |
| 14 (Ehex) | 0 1 1 1-0   | Unused character                 |
| 15 (Fhex) | 1 1 1 1-1   | End sentinel (stop character)    |

The least significant bit of every digit is sent first; the fifth bit is an odd parity bit for each group of 4 data bits.

The complete message always looks as follows:

|           |       |                 |     |     |            |
|-----------|-------|-----------------|-----|-----|------------|
| Left edge | Start | Data characters | End | LRC | Right edge |
|-----------|-------|-----------------|-----|-----|------------|

The LRC is calculated by the following procedure: each of the 4 bits in the LRC character is an even parity bit of the equivalent bits in the telegram including start and stop sentinel.

The fifth bit is the odd parity of the 4 LRC bits (it is not calculated over all the parity bits).

Input data should be preceded and followed by a clock synchronization pattern (NULL data).

## Dataclock levels

In normal operation mode (default) input and output signals are defined:

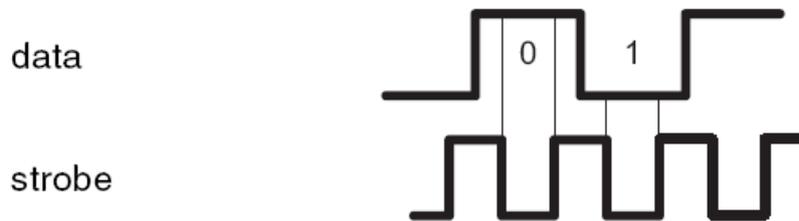


Figure 9: Data Clock signals

Other modes are (only for output):

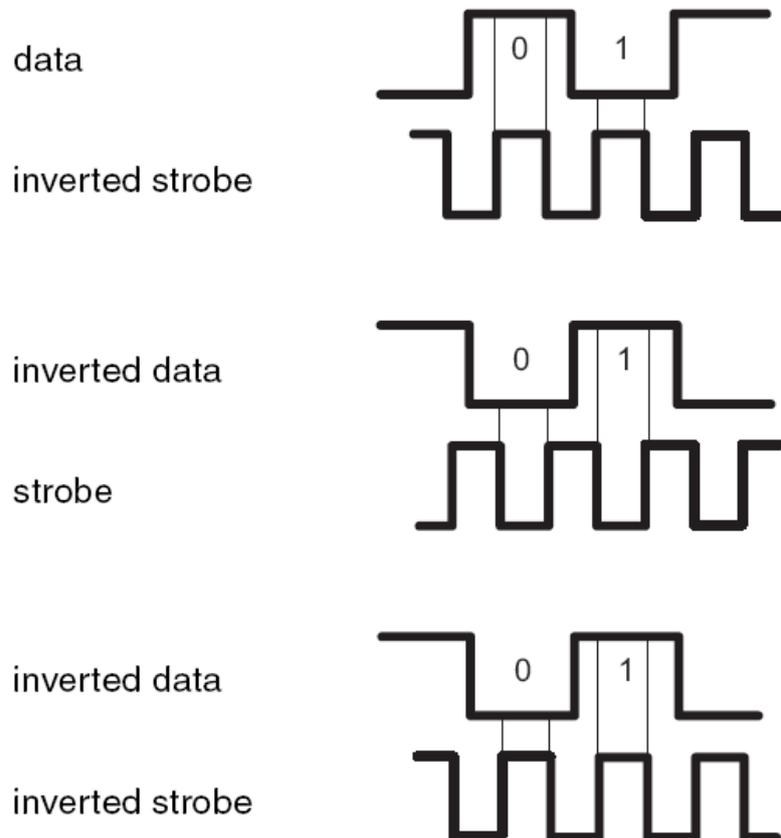


Figure 10: Other Data Clock signals



## RS485 Protocol

### Data Packet Structure:

- The packet format is:

|                 |    |     |      |               |     |     |
|-----------------|----|-----|------|---------------|-----|-----|
| STX             | ID | TID | DATA | CRC           | DLE | ETX |
| Start Of Packet |    |     |      | End Of Packet |     |     |

### Abbreviation:

| Fields name | Definition                 | Size (Bytes) | Value       |
|-------------|----------------------------|--------------|-------------|
| <STX>       | Start Text                 | 1            | <b>0x02</b> |
| <ID>        | Packet Identifier          | 1            | <b>0xE1</b> |
| <TID>       | Terminal Identifier        | 1            | --          |
| <DATA>      | Data value                 | Up to 1024   | --          |
| <CRC>       | Transmission error control | 2            | --          |
| <DLE>       | Data Link Escape           | 1            | <b>0x1B</b> |
| <ETX>       | End Text                   | 1            | <b>0x03</b> |

- The maximum size allowed for a packet is 2058 bytes:  
(STX+ID+DLE+ETX+ (TID+DATA+CRC)\*2 [if stuffed])

### Byte Order:

- The packet byte order is in Little Endian format: multi bytes data are sent with Least Significant Byte (LSB) first.

### Data:

- Data are formatted as ILV packets.

### Stuffing:

- Software handshake capabilities (XON-XOFF) are preserved by replacing, in the <TID + Data + CRC>, all XON (0x11) / XOFF (0x13) characters by the couple <DLE> <XON+1> (0x12) or <DLE> <XOFF+1> (0x14).
- To prevent confusion with the frames sequences <STX><ID> and <DLE><ETX>, every <DLE> byte in the <TID+ Data + CRC> is preceded by an extra <DLE> byte ('stuffing').
- Stuffing must be processed before sending a packet and removed ('unstuffed') after receiving the packet.
  - NOTE:** A simple <DLE> <ETX> sequence does not necessarily signify the end of the packet, as these can be bytes in the middle of a data string.
- The end of a packet is <ETX> preceded by an odd number of <DLE> bytes.

### CRC Calculation:

- The type of the CRC is CRC16 V41.
- The CRC is computed as a function of the Data part before Stuffing.

- The initial value is 0x0000.

**Packet Identifier:**

- The packet identifier byte is 0x61.

**Terminal Identifier:**

- The terminal identifier byte defines the MorphoAccess® address on the RS485 network.

**Frames sequence:**

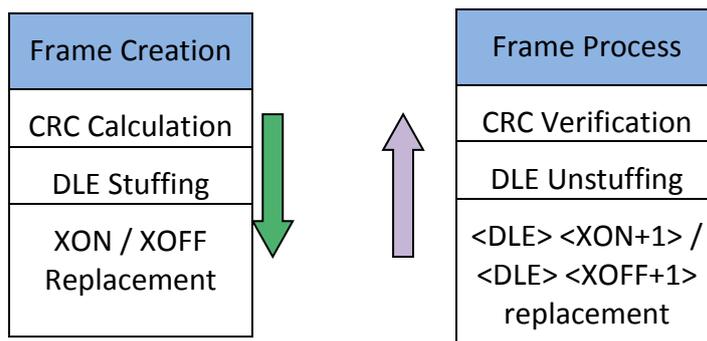


Figure 11 : RS485 frame processing

**Frames Emission:**

- 3 attempts are made in case of failure.

**Examples:**

- User “094066” has been recognized by a MorphoAccess® number 89 (TID).

| STX | ID | TID | Data: ILV |    |    |    |    |    |    |    |    | CRC | CRC | DLE | ETX |
|-----|----|-----|-----------|----|----|----|----|----|----|----|----|-----|-----|-----|-----|
| 02  | E1 | 59  | 00        | 06 | 00 | 30 | 39 | 34 | 30 | 36 | 36 | CE  | D1  | 1B  | 03  |

- User “62487” has been recognized by a MorphoAccess® number 89 (TID).

| STX | ID | TID | Data: ILV |    |    |    |    |    |    |    |    | CRC | CRC | DLE | ETX |
|-----|----|-----|-----------|----|----|----|----|----|----|----|----|-----|-----|-----|-----|
| 02  | E1 | 59  | 00        | 05 | 00 | 36 | 32 | 34 | 38 | 37 | A0 | AA  | 1B  | 03  |     |

- Identification failed on MorphoAccess® number 89 (TID).

| STX | ID | TID | Data: ILV |    |    |    | CRC | CRC | DLE | ETX |
|-----|----|-----|-----------|----|----|----|-----|-----|-----|-----|
| 02  | E1 | 59  | 10        | 01 | 00 | 01 | B6  | 3C  | 1B  | 03  |



## RS422 Protocol

### Data Packet Structure:

- The packet format is:

|                 |    |    |      |     |               |     |
|-----------------|----|----|------|-----|---------------|-----|
| STX             | ID | RC | DATA | CRC | DLE           | ETX |
| Start Of Packet |    |    |      |     | End Of Packet |     |

### Abbreviation

| Fields name | Definition                 | Size (Bytes) | Value       |
|-------------|----------------------------|--------------|-------------|
| <STX>       | Start Text                 | 1            | <b>0x02</b> |
| <ID>        | Packet Identifier          | 1            | --          |
| <RC>        | Request Counter            | 1            | --          |
| <DATA>      | Data value                 | Up to 1024   | --          |
| <CRC>       | Transmission error control | 2            | --          |
| <DLE>       | Data Link Escape           | 1            | <b>0x1B</b> |
| <ETX>       | End Text                   | 1            | <b>0x03</b> |

- The maximum size allowed for a packet is 2 058 bytes:  
(STX+ID+DLE+ETX+ (RC+DATA+CRC)\*2 [if stuffed])

### Byte Order:

- The packet byte order is in Little Endian format: multi bytes data with Least Significant Byte (LSB) first.

### Data:

- Data are formatted as ILV packets.

### Stuffing:

- Software handshake capabilities (XON-XOFF) are preserved by replacing, in the <RC + Data + CRC>, all XON (0x11) / XOFF (0x13) characters by the couple <DLE> <XON+1> (0x12) or <DLE> <XOFF+1> (0x14).
- To prevent confusion with the frames sequences <STX><ID> and <DLE><ETX>, every <DLE> byte in the <RC+ Data + CRC> is preceded by an extra <DLE> byte ('stuffing').
- Stuffing must be processed before sending a packet and removed ('unstuffing') after receiving the packet.
- Notice that a simple <DLE> <ETX> sequence does not necessarily signify the end of the packet, as these can be bytes in the middle of a data string.
- The end of a packet is <ETX> preceded by an odd number of <DLE> bytes.

### CRC Calculation:

- The type of the CRC is CRC16 V41.
- The CRC is computed as a function of the Data part before Stuffing.
- The initial value is 0x0000.

#### Packet Identifier:

The identifier is formatted as follows:

| Bit 7 (MSB) | Bit 6 | Bit 5 | Bit 4      | Bit 3       | Bit 2 | Bit 1 | Bit 0 (LSB) |
|-------------|-------|-------|------------|-------------|-------|-------|-------------|
| IN/OUT      | First | Last  | Reserved 0 | Packet Type |       |       |             |

- The MSB (Bit 7) is reserved for packet direction. Setting this bit sets the direction to IN. Clearing this bit will set the direction to OUT.
- An OUT packet is a packet sent **by the Host system to the MorphoAccess®** terminal.
- An IN packet is a packet sent **by the MorphoAccess®** terminal **to the Host system**.
- **Bit 6** is reserved for Packet Order information. Set this bit when it is the first packet while transmitting a set of packets.
- **Bit 5** is reserved for Packet Order information. Set this bit when it is the last packet while transmitting a set of packets.
- **Bit 4** is a reserved bit and must be cleared.
- **Bits 3 to 0** are used for Packet Identification:

The following packet types are implemented:

| ID Value | Description |
|----------|-------------|
| 0x1      | Data Packet |
| 0x2      | ACK Packet  |
| 0x4      | NACK Packet |

#### Frames sequence:

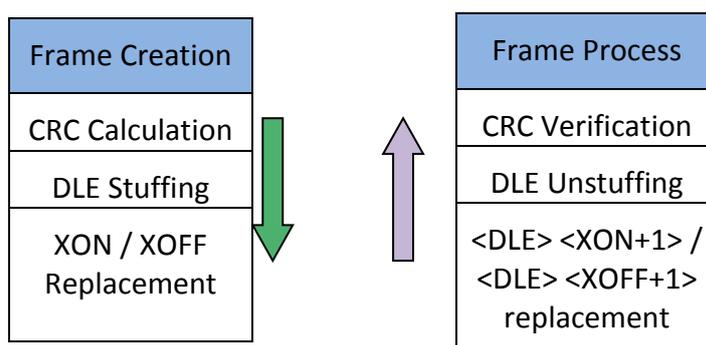


Figure 12: RS422 frame processing

#### Timing Characteristics:

- The maximum time elapsed between the transmissions of two bytes of a frame is 100ms.

- The maximum time elapsed between the emission of a Packet Data and the reception of the ACK is 500ms.

### Communication Error Case:

The following error cases must be detected:

- Timeout between the reception of two bytes (the timeout starts after the reception of STX),
- Bad CRC check,
- Unstuffing error (<DLE> is followed by an unexpected character).

### Request Counter (RC) management:

The following rules have to be implemented:

- the RC of a data is filled with the current Counter value,
- the RC of an ACK/NACK packet is filled with the RC of the data packet to ACK/NACK,
- on the reception of an ACK/NACK packet, the RC is compare to the latest data packet sent. If it is an ACK, the counter is increased on a hit. If several packets are received with the same RC, only one ACK is sent.

### Retransmission:

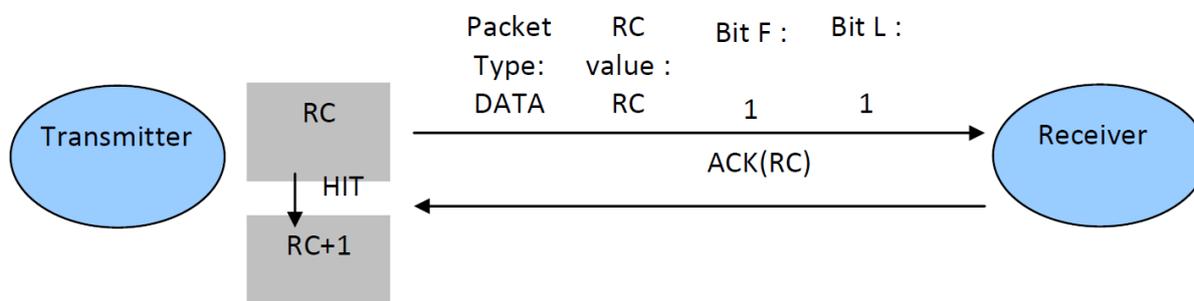
- In case of reception of a NACK or Timeout, the transmitter tries to send the same packet again.
- A packet can be retransmitted 3 times. After that, another NACK reception or Timeout leads to ERROR of the transmission.

### Error cases:

- When a frame is not valid (Bad CRC, Unstuffing error, Rx timeout), the receiver must send a NACK packet.
- When the transmitter is waiting for an ACK/NACK packet, all the other packet that are received, must be ignored.

### Typical Transactions workflow:

- Emission of a data packet that contains less than 1024 bytes of effective data:



- Emission of a data packet that contains more than 1024 bytes of effective data:

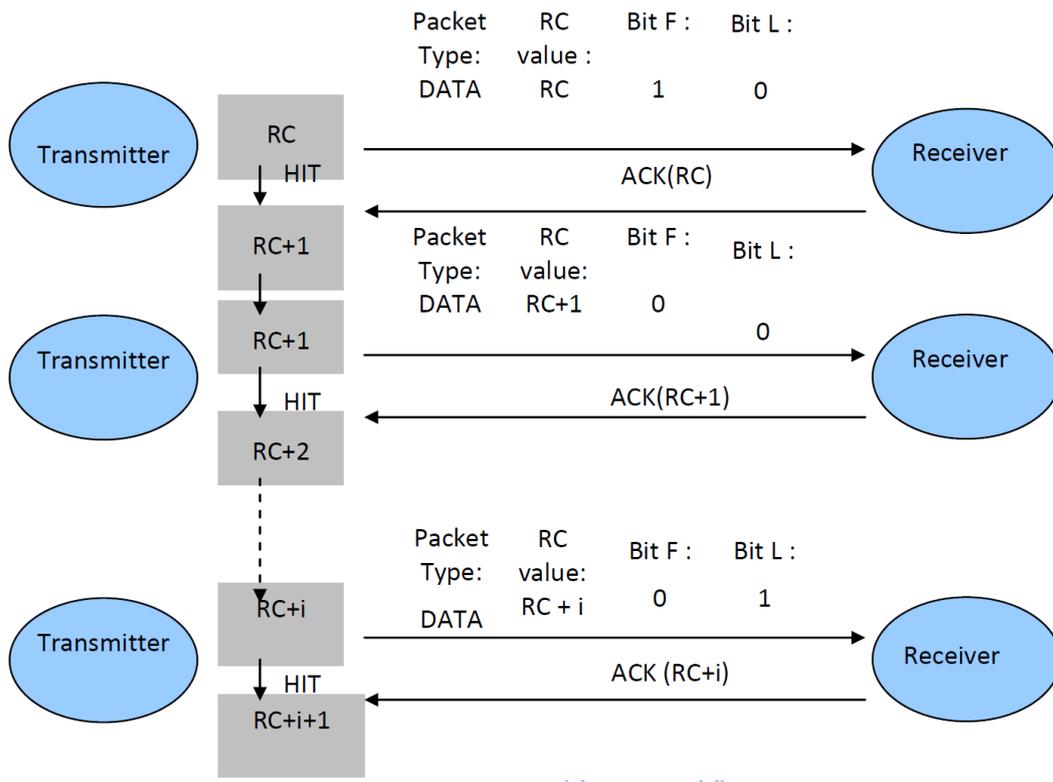


Figure 13: RS422 typical frame workflow

- An Error occurred while transmitting the data packet:

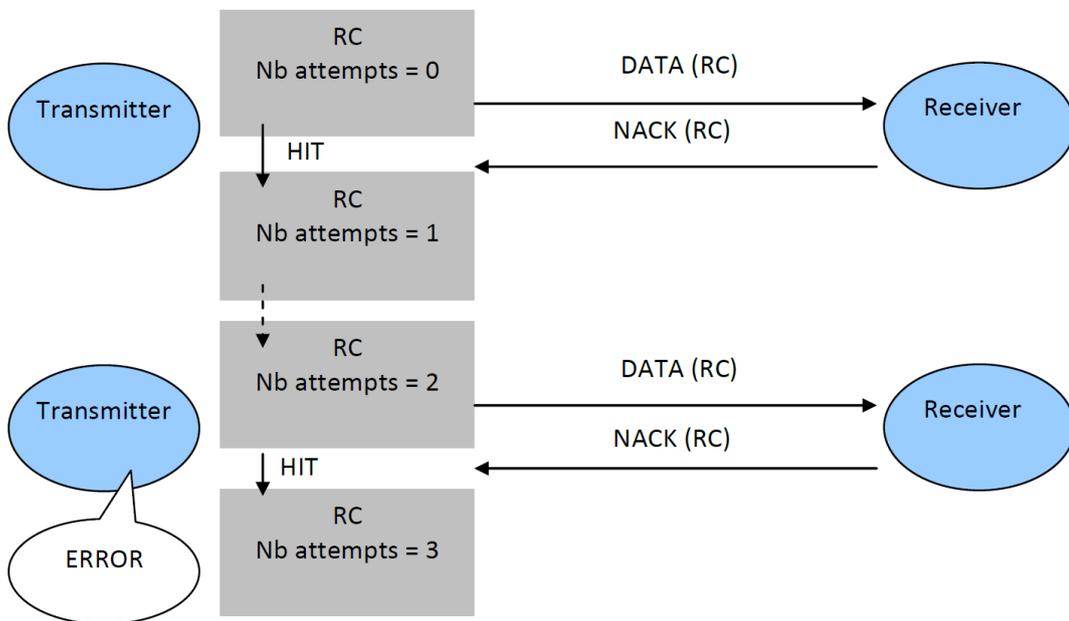


Figure 14: RS422 transmission error

- The data packet is transmitted but the receiver does not transmit ACK or NACK:

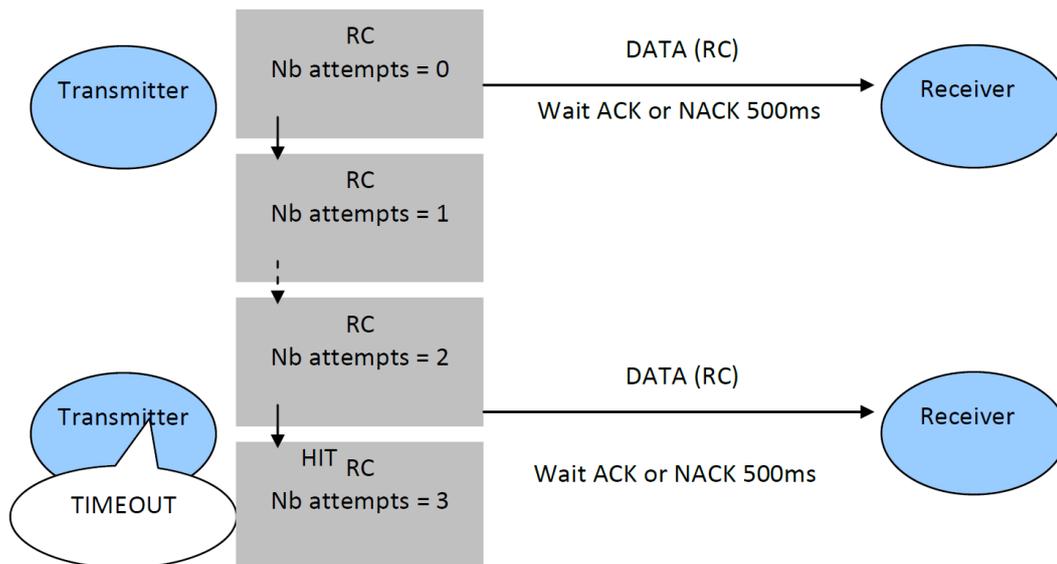


Figure 15: RS422 transmission timeout error

**Examples:**

- User “094066” has been recognized by a MorphoAccess® terminal.

| STX | ID | RC | Data: ILV |    |    |    |    |    |    |    | CRC | CRC | DLE | ETX |    |
|-----|----|----|-----------|----|----|----|----|----|----|----|-----|-----|-----|-----|----|
| 02  | E1 | XX | 00        | 06 | 00 | 30 | 39 | 34 | 30 | 36 | 36  | CE  | D1  | 1B  | 03 |

- User “62487” has been recognized by a MorphoAccess® terminal.

| STX | ID | RC | Data: ILV |    |    |    |    |    |    |    | CRC | CRC | DLE | ETX |
|-----|----|----|-----------|----|----|----|----|----|----|----|-----|-----|-----|-----|
| 02  | E1 | XX | 00        | 05 | 00 | 36 | 32 | 34 | 38 | 37 | A0  | AA  | 1B  | 03  |

- Identification failed on MorphoAccess® terminal.

| STX | ID | RC | Data: ILV |    |    |    | CRC | CRC | DLE | ETX |
|-----|----|----|-----------|----|----|----|-----|-----|-----|-----|
| 02  | E1 | XX | 10        | 01 | 00 | 01 | B6  | 3C  | 1B  | 03  |

**Note:** CRC values calculated assuming RC value 0x59 in all above examples.



## How to get latest version of the documents

The last version of the documents is available on a CD-ROM package from our factory, or can be downloaded from our web site at the address below:

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

(Login and password required).

To request a login, please send us an email to the address below:

[support.bioterminals@idemia.com](mailto:support.bioterminals@idemia.com)

## Documents concerning the MorphoAccess® terminal

### *Bibliography*

#### **MorphoAccess® SIGMA Family Bibliography,**

Ref. 2016\_2000022505 - [MorphoAccess® SIGMA Family Bibliography](#)

This document gives document's references for MorphoAccess® SIGMA Family terminals. This document is in English.



## GLOSSARY

- **Access Controller/Controller:** This term is used for centralized access controller. Terminal communicates with controller for granting or denying access to the user.
- **Host:** This term is used for server of Access Controller.
- **Terminal:** This term is used for MorphoAccess® SIGMA Family terminal
- **Device:** This term is used for an external device attached to MorphoAccess® SIGMA Family terminal, such as USB Mass Storage device.
- **User Enrolment:** creation of a record in a database with personal data of a unique user, or creation of a card with personal data of a user
- **Firmware:** The set of programs contained permanently in a hardware device (as read-only memory) that controls the unit.
- **Host Mode:** The normal mode of operation when the device is waiting for a card to be presented to the terminal.
- **Single Door Access Control (SDAC):** The capability of controlling/monitoring all functions related to a single entry/exit point.
- **Software** The set of programs associated with a computer system.
- **Template:** A term used to describe the data that is stored during the enrolment process. The data is a mathematical representation of the ridge pattern of the enrolled finger scan.
- **Primary Template:** This is the template that resides in the first template slot on the smart card. When verification is initiated, this primary template is the first template that is used in that verification process.
- **Secondary Template:** This is an optional second template stored on the smart card that is also used in the verification process if the primary template verification fails.
- **1:1 Mode:** In 1:1 mode, a user enters his or her User ID first. Then the user is requested to provide a personal data such as place a finger on a sensor or enter a PIN. Then the acquired data is matched against the reference data linked to user ID (example: fingerprint found on users' card which provides the User ID at beginning of the process).
- **1: N Mode:** In 1: N mode, a user places his or her finger on the device without entering an ID. The terminal compares the user's scanned finger with the many enrolled fingers in its internal database.
- **Identification (Searching or 1:N):** The operation of Identifying a user by comparing a live finger scan against all stored finger-scan records in a database to determine a match. Identification uses the finger scan only - no cards or PINs. Identification is only available on devices that are in 1:N mode.

- **Authentication (1:1):** The operation of confirming a user is who he claims to be by comparing a live finger scan image against a stored fingerprint template. The result (pass or fail) that is returned is based on whether the score is above a pre-defined threshold value. Some type of credential (PIN, Prox card, smart card, etc.) is necessary to initiate the biometric verification.
- **Webserver:** Webserver is a web-based application embedded in the MorphoAccess® SIGMA Family terminal. Webserver enables the management of the settings of the terminal from any computer (desktop, laptop, tablet ...) equipped with a compatible Internet browser and connected to the same network as the terminal.
- **ILV:** All the exchanged messages over IP between the MorphoAccess® SIGMA Family terminal and the remote controller have the same structure. These messages are called **ILV** (Identifier, Length, and Value).

## Acronyms and Abbreviations

- **MA:** MorphoAccess®, a generic name of the physical access control terminals by IDEMIA.
- **LCD:** Liquid Crystal Display
- **LED:** Light Emitting Diode
- **MAC (address):** Media Access Control, a unique identifier assigned to network interfaces for communications on the physical network segment
- **IPv4:** Internet Protocol version 4
- **IPv6:** Internet Protocol version 6 - IPv6 is intended to replace IPv4, which still carries the large majority of Internet traffic (2013).
- **DNS:** Domain Name Server. It provides naming for all systems, computers, terminals in a network
- **DHCP:** Dynamic Host Configuration Protocol
- **TCP:** Transmission Control Protocol
- **UDP:** User Datagram Protocol
- **SSL:** Secure Sockets Layer
- **PIN:** Personal Identification Number
- **T&A:** Time and Attendance Mode
- **MMI:** Man Machine Interface
- **SDAC:** Single Door Access Control
- **GPIO:** General Purpose Input Output



## 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](mailto: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.



Head office:

**IDEMIA**

11, boulevard Gallieni

92130 Issy-les-Moulineaux – France

[www.idemia.com](http://www.idemia.com)