

Access and Time Biometric Terminals

Host System and Remote Message Interfaces



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
01	2016-11-02	New document's reference based on 2015_2000014298_v2 Features added or updated: <ul style="list-style-type: none">- Enhanced MMI answer- RS485/RS422 - Message format- RS485: Annex 3- RS422: Annex 4
02	2016-11-24	Features updated: <ul style="list-style-type: none">- Enhanced MMI answer: Relay activation duration unit change
03	2017-04-12	Replace 5G Series with SIGMA Family Add the MorphoAccess® SIGMA Extreme Series product Change the contact information
04	2017-06-21	Add note about text display duration in Enhanced MMI
05	2017-12-27	Update company name (IDEMIA)
06	2018-04-17	Features updated: <ul style="list-style-type: none">- Section 2: IP Remote messages- Section 5: Extended remote message format and real time mode
07	2018-10-03	Add <i>MorphoWave</i> Compact
08	2020-01-14	Add limitation, feedback not supported with UDP protocol
09	2020-05-26	Merge with Host System Interface Add VisionPass

Table of Contents

REVISION HISTORY	3
SECTION 1 : INTRODUCTION	10
<i>Scope of the document.....</i>	<i>11</i>
SECTION 2 : HOST SYSTEM INTERFACE	13
<i>Host System Interface Overview</i>	<i>14</i>
<i>Remote Management Protocol</i>	<i>15</i>
TCP Remote Management	15
SSL Securing.....	16
RS422 Remote Management	19
<i>About Thrift Commands</i>	<i>20</i>
Implementation tips.....	20
<i>Serial Communication</i>	<i>21</i>
Serial Packet Structure	21
Data chunking	22
SECTION 3 : TCP/SSL/UDP REMOTE MESSAGES	23
<i>Remote Message Overview</i>	<i>24</i>
Supported Protocols.....	25
<i>IP Remote Messages</i>	<i>27</i>
Presentation.....	27
Activation of Remote Message	28
Controller Capabilities.....	30
Default Controller Definition	30
Alternative Controller Definition	31
Remote Message Sending Flow	32
SECTION 4 : SERIAL REMOTE MESSAGES	35
<i>Serial Remote Messages</i>	<i>36</i>
RS485 Presentation.....	36
RS422 Presentation.....	36
Activation	37
Terminal Identifier.....	37
Data Format.....	38
RS485/RS422 - Message format	38
Serial Link Settings	39

SECTION 5 : WIEGAND / DATACLOCK REMOTE MESSAGES	40
Wiegand Remote Messages	41
Presentation.....	41
External Port Wiegand Protocol selection	41
Activation	41
Setting up Wiegand Interface.....	42
Duress finger event Wiegand frame definition	42
User control failure event frame definition.....	43
User control success event frame definition	45
Tamper event Wiegand frame definition.....	47
Wiegand Frame Timing.....	48
DataClock Remote Messages	49
Presentation.....	49
External Port Dataclock Protocol selection	49
Activation	49
DataClock: failure messages.....	50
Presentation.....	50
Software Configuration.....	50
SECTION 6 : REMOTE MESSAGE FRAMES.....	51
Message Format	52
Basic format	52
Extended format	52
Control is OK.....	54
Description	54
Frame sent when control is OK	54
Example.....	55
Control Failed.....	56
Description	56
Command: sent frame.....	56
Examples	57
Tamper Alarm sent by Access and Time Biometric terminal	59
Description	59
Sending.....	59
Command: sent frame.....	59
Internal log file full Message	60
Description	60
Message sent	60

Door opened for too long	61
Description	61
Message sent	61
Forced door open	62
Description	62
Message sent	62
Door closed after alarm.....	63
Description	63
Message sent	63
Door unlocked.....	64
Description	64
Message sent	64
Door locked back.....	65
Description	65
Message sent	65
Management menu login	66
Description	66
Message sent	66
Management menu logout.....	67
Description	67
Message sent	67
Database deleted.....	68
Description	68
Message sent	68
Enrolment completed	69
Description	69
Message sent	69
Deletion completed.....	70
Description	70
Message sent	70
User modification completed.....	71
Description	71
Message sent	71
Contactless card encoded	72
Description	72
Message sent	72
Contactless card reset	73

Description	73
Message sent	73
Settings changed.....	74
Description	74
Message sent	74
Contactless card security keys reset.....	75
Description	75
Message sent	75
Firmware upgrade	76
Description	76
Message sent	76
Job code check failure	77
Description	77
Message sent	77
Terminal boot completed	78
Description	78
Message sent	78
Add user	79
Description	79
Message sent	79
Reboot initiated	80
Description	80
Message sent	80
Duress finger detected	81
Description	81
Message sent	81
Security policy changed.....	82
Description	82
Message sent	82
Basic MMI Answer (Returned by the Controller)	83
Description	83
Command: frame returned by the controller to the terminal.....	83
Enhanced MMI Answer (Returned by the Controller)	84
Description	84
Command: frame returned by the controller to the terminal.....	84
Command without any action to perform	86
Example.....	86

<i>Technical Support and Hotline</i>	<i>108</i>
<i>North America</i>	<i>108</i>
<i>South America</i>	<i>108</i>
<i>Asia, Pacific:</i>	<i>108</i>
<i>Europe, Middle-East, Africa</i>	<i>108</i>
<i>Web site</i>	<i>108</i>
ANNEX 1 : WIEGAND DATA FORMAT	87
ANNEX 2 : ISO 7811/2 - 1995 - TRACK 2 DATACLOCK FORMAT.....	89
ANNEX 3 : RS485 PROTOCOL.....	92
ANNEX 4 : RS422 PROTOCOL.....	95
ANNEX 5 : BIBLIOGRAPHY	101
ANNEX 6 : GLOSSARY, ACRONYMS AND ABBREVIATION	103
ANNEX 7 : SUPPORT	107

Table of Figures

Figure 1: Operating terminal through Host system	14
Figure 1: Remote Message Overview	24
Figure 2: Ethernet or Wi-Fi™, UDP, TCP or SSL protocols.....	27
Figure 3: Remote Message Flow when response from controller is not required.....	32
Figure 4: Remote Message Flow when response from controller is required	33
Figure 5: Remote Message Flow for stored events in Real Time mode	34
Figure 6: Serial port, RS485 protocol	36
Figure 7 : Serial port, RS422 protocol	36
Figure 8: Wiegand frame format	88
Figure 9: Data Clock signals.....	91
Figure 10: Other Data Clock signals	91
Figure 11 : RS485 frame processing.....	94
Figure 12: RS422 frame processing.....	97
Figure 13: RS422 typical frame workflow	99
Figure 14: RS422 transmission error.....	99
Figure 15: RS422 transmission timeout error.....	100

Section 1 : Introduction

Scope of the document

This guide relates to the host system and remote message interfaces of Access and Time Biometric Terminal.

Terminal Series	Terminal Name	Biometrics	Contactless smartcard reader		
			iCLASS® iCLASS® SE	MIFARE® DESFire® NFC®	Prox®
MorphoAccess® SIGMA Series	MorphoAccess® SIGMA	✓			
	MorphoAccess® SIGMA iCLASS®	✓	✓		
	MorphoAccess® SIGMA Multi	✓		✓	
	MorphoAccess® SIGMA Prox	✓			✓
MorphoAccess® SIGMA Lite Series	MorphoAccess® SIGMA Lite MorphoAccess® SIGMA Lite+	✓			
	MorphoAccess® SIGMA Lite iCLASS® MorphoAccess® SIGMA Lite + iCLASS®	✓	✓		
	MorphoAccess® SIGMA Lite Multi MorphoAccess® SIGMA Lite + Multi	✓		✓	
	MorphoAccess® SIGMA Lite Prox MorphoAccess® SIGMA Lite + Prox	✓			✓
MorphoAccess® SIGMA Extreme Series	MorphoAccess® SIGMA Extreme iCLASS®	✓	✓		
	MorphoAccess® SIGMA Extreme Multi	✓		✓	

Terminal Series	Terminal Name	Biometrics	Contactless smartcard reader		
			iCLASS® iCLASS® SE	MIFARE® DESFire® NFC®	Prox®
	MorphoAccess® SIGMA Extreme Prox	✓			✓
	MorphoAccess® SIGMA Extreme FFD iCLASS®	✓	✓		
	MorphoAccess® SIGMA Extreme FFD Multi	✓		✓	
	MorphoAccess® SIGMA Extreme FFD Prox	✓			✓
MorphoWave® Compact	MorphoWave® Compact MDPI	✓	✓	✓	✓
	MorphoWave® Compact MD	✓		✓	
VisionPass	VisionPass MDPI	✓	✓	✓	✓
	VisionPass MD	✓		✓	

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

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

Section 2 : Host System Interface

Host System Interface Overview

The Access and Time Biometric Terminals provides remote management facilities, using which it is possible to change the terminal settings through Ethernet using a TCP-IP connection, or through serial link using a RS422 connection.

The Access and Time Biometric Terminal acts as a server and provides a unique socket, in case of TCP-IP connection.

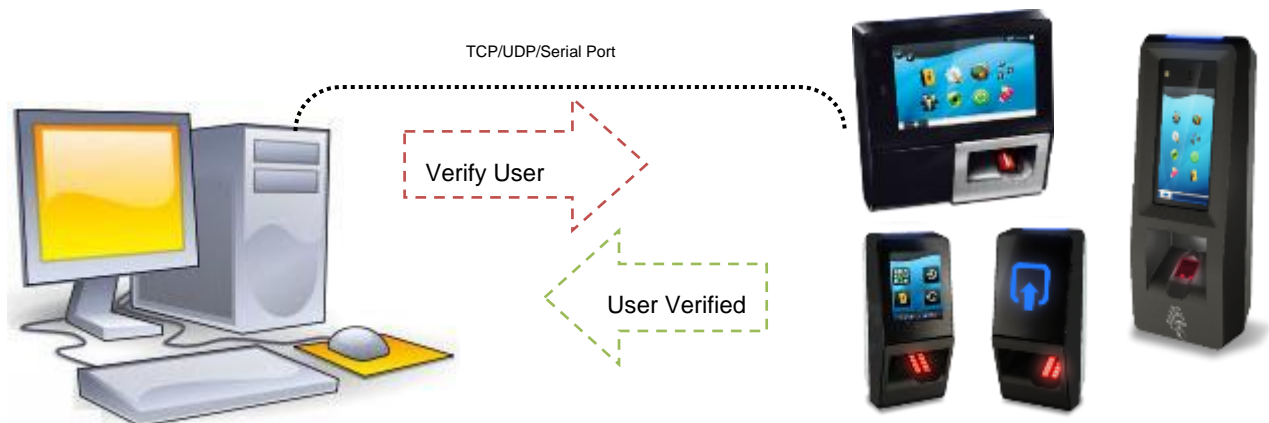


Figure 1: Operating terminal through Host system

The following operations are allowed:

- Changing system settings such as control type, control timeout...,
- Sending biometric requests,
- Firmware upgrade

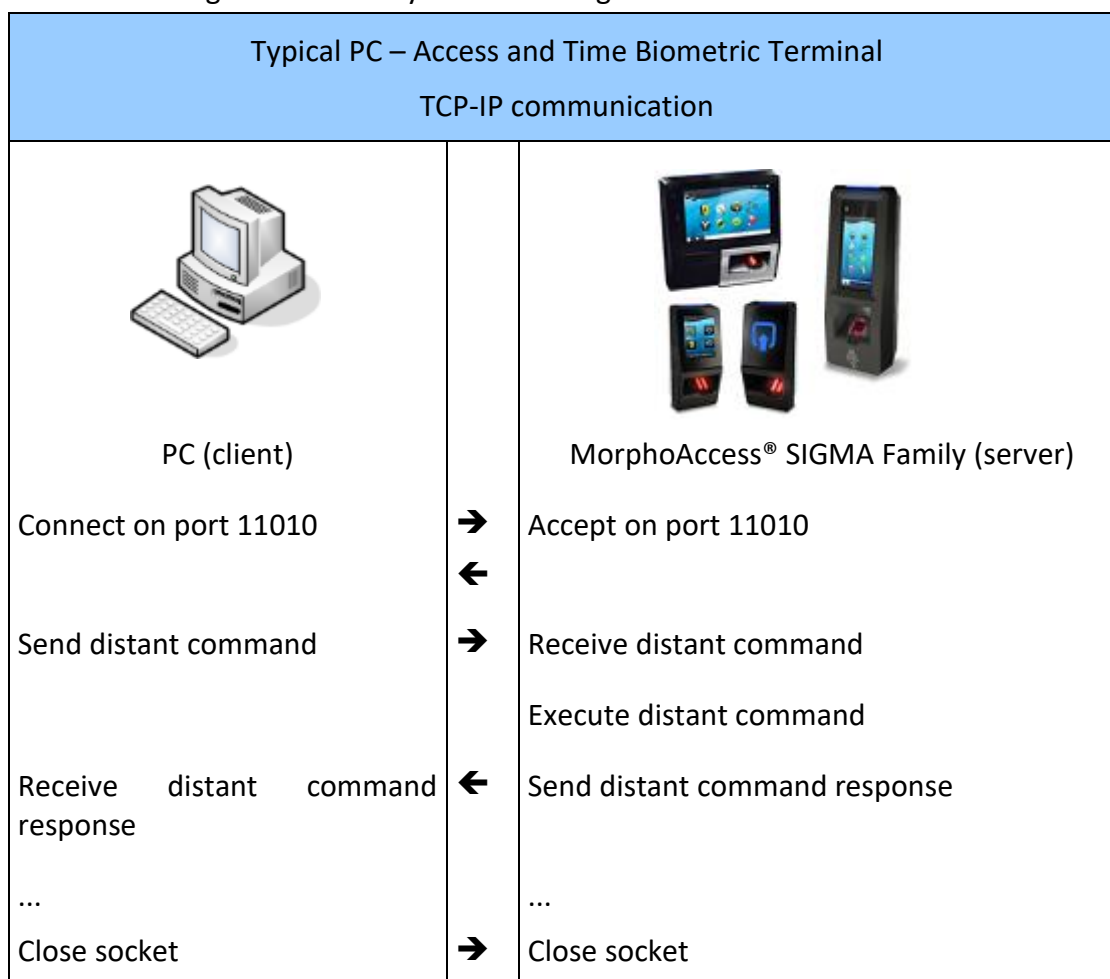
Recommendation: If network intensive or database intensive operations are performed on terminal from host system using distant commands, then the terminal response time would be affected until this background operation is completed. Hence it is advisable to do such network or database intensive operation when terminal is in idle state.

Remote Management Protocol

TCP Remote Management

It is possible to administrate a Access and Time Biometric Terminal from a distant computer. In this case the Access and Time Biometric Terminal works as a standard TCP server waiting for requests coming from a remote client.

A standard exchange follows always the following schema:



Note: Maximum 10 input connections can be opened at the same time on Access and Time Biometric Terminal.

SSL Securing

Secure Sockets Layer (SSL) and its successor Transport Layer Security (TLS) are cryptographic protocols designed to provide communication security over Ethernet or Wi-Fi™ channels.

These protocols are used to protect the communication between Access and Time Biometric Terminal and a distant system, such as a central access controller or a terminal configuration station.

The cryptographic protocols supported by the terminal are listed below:

- SSLv3 only (Only for compatibility purposes)
- SSLv23 (Accept TLS 1.0, 1.1, 1.2)
- TLS 1.0 only
- TLS 1.1 only
- TLS 1.2 only

The terminal supports the algorithms listed below for communication security:

- AES128-SHA Openssl ciphersuite
- AES256-SHA Openssl ciphersuite
- AES128-SHA256 Openssl ciphersuite
- AES256-SHA256 Openssl ciphersuite
- AES128-GCM-SHA256 Openssl ciphersuite
- ECDHE-ECDSA-AES256-SHA Openssl ciphersuite
- ECDHE-ECDSA-AES128-GCM-SHA256 Openssl ciphersuite
- ECDHE-ECDSA-AES128-SHA256 Openssl ciphersuite
- ECDHE-ECDSA-AES128-SHA Openssl ciphersuite

Note: The communication security is automatically configured during negotiation between the client and the server. The client specifies the security level requested, and the server accepts or proposes a lower level. The client accepts it or cancels its request. The final configuration corresponds to the higher security level common with the client and the server.

Compatibility of cipher algorithms with SSL protocol versions

Cipher Algorithm List	Protocol Version				
	ssl23	ssl3	tlsv1	tlsv1.1	tlsv1.2
AES128-SHA	Y	Y	Y	Y	Y
AES256-SHA	Y	Y	Y	Y	Y
AES128-SHA256	N	N	N	N	Y
AES256-SHA256	N	N	N	N	Y
AES128-GCM-SHA256	N	N	N	N	Y
ECDHE-ECDSA-AES256-SHA:ECDH-ECDSA-AES256-SHA	Y	Y	Y	Y	Y
ECDHE-ECDSA-AES128-GCM-SHA256:ECDH-ECDSA-AES128-GCM-SHA256	N	N	N	N	Y
ECDHE-ECDSA-AES128-SHA256:ECDH-ECDSA-AES128-SHA256	N	N	N	N	Y
ECDHE-ECDSA-AES128-SHA:ECDH-ECDSA-AES128-SHA	Y	Y	Y	Y	Y

Note: Cipher algorithm that ends with 'SHA256' supports only SSL protocol version tls1.2.

SSL Protocol Versions support for remote communication

		Client side (from PC application)				
		sslsv23	sslsv3	tlsv1	tlsv11	tlsv12
On Terminal	sslsv23	Y	N	Y	Y	Y
	sslsv3	Y	Y	N	N	N
	tlsv1	Y	N	Y	N	N
	tlsv11	Y	N	N	Y	N
	tlsv12	N	N	N	N	Y

The above table describes the protocol versions supported by client side application, when communication is started by terminal using specific protocol. For e.g. If terminal starts communication using sslsv23 protocol, then client side application will be able to communicate using all the protocol versions. While if communication is initiated using sslsv3 protocol, then client application will only support sslsv23 and sslsv3 protocol versions for communication.

References

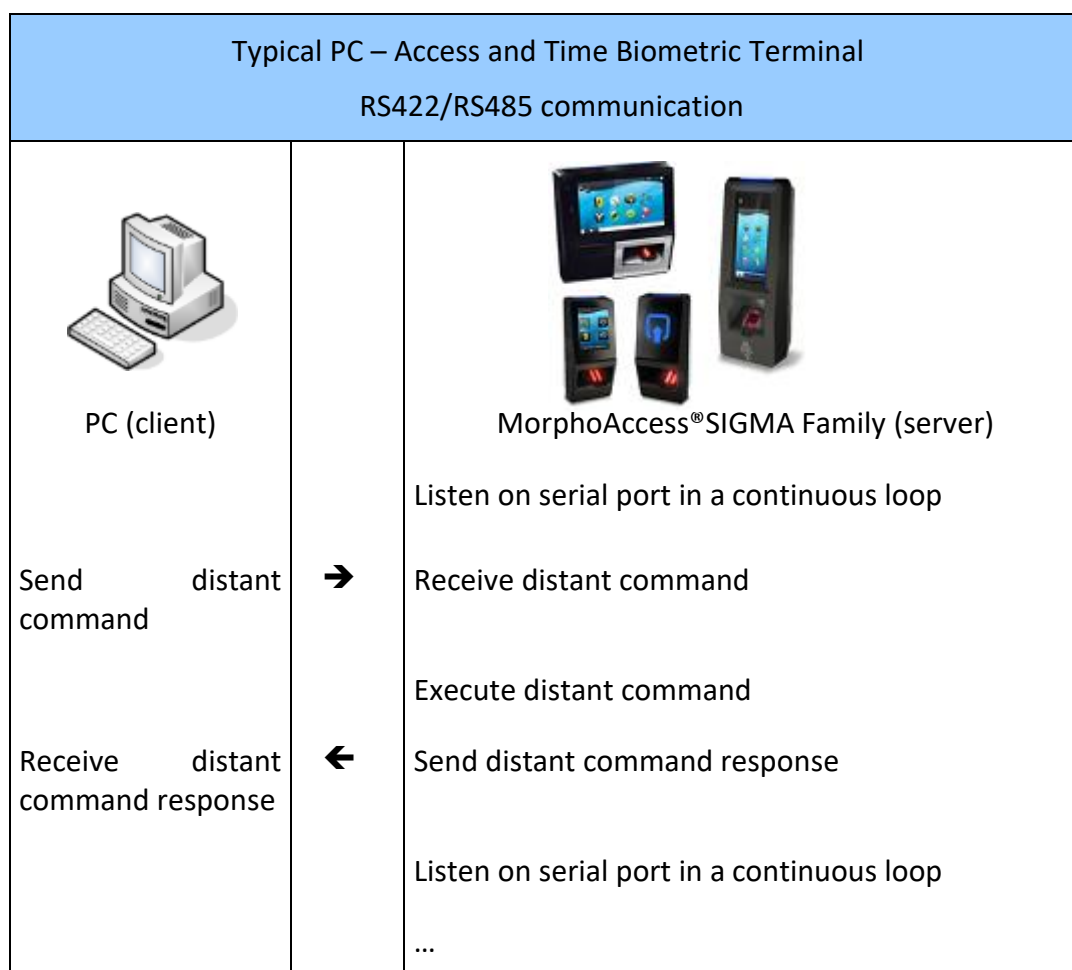
- See “SSL Configuration” section under Security Menu, in Access and Time Biometric Terminal **Admin User Guide**, for details on how to configure SSL communication port
- See SSL Solution for MorphoAccess® documentation for details on SSL securing

RS422 Remote Management

It is also possible to administrate the Access and Time Biometric Terminal from a remote computer using a RS422 connection. In this case the Access and Time Biometric Terminal works as a server waiting for requests coming from a remote client.

The terminal listens to the serial port. If there is data, then the terminal executes the requests, else it listens again to the serial port until there is something or the user asks to close the connection.

A standard exchange follows always the following schema:



About Thrift Commands

Access and Time Biometric Terminals can communicate using Apache Thrift framework. Know more about Apache Thrift framework on <http://thrift.apache.org/>.

Distant commands used by host system for interacting with terminal in MA5G mode are available in document, “**MA5G_distant_commands**”.

Implementation tips

- In order to optimize the network bandwidth and transfer timing and to avoid communication errors, we recommend to use buffered transport layer. This can be achieved differently depending on the programming language that is used:
 - In C# or C++, you can use the TBufferedTransport class from Thrift library.
 - In Java, you can use the TSocket class from Thrift library (it internally uses buffered streams).
 - For any language you could find an implementation in the Thrift library or implement your own system of buffering in memory before writing the complete command to the socket.
- We recommend to use Thrift 0.9.0, or Thrift 0.9.2 especially for C# developers who want to benefit from native TLS support in Thrift library.

Serial Communication

The serial channel communication implements a particular protocol and applies it over the data to send or receive. The data transfer is accompanied by the appropriate data encapsulation and data chunking.

Serial Packet Structure

- **MA5G communication mode:** Following is the packet structure that the serial channel shall implement:

Byte#	1	2	3 – 4	5 - 8	9 – 10	11 - <11 + Data size>
Field	0x4D	0x41	<Net ID>	<Checksum>	<Data size>	<Data>
Length	1	1	2	4	2	<Data size> (maximum 1024 bytes)
Endianness	-	-	Big	Big	Big	-

Checksum calculation

The checksum shall be calculated on the data part only.

Steps:

1. Take a sum all the bytes in the data.
2. Take 2's complement of the sum.

Example:

If the data is 0x0102, then checksum is $((\sim(0x01 + 0x02) + 1) \Rightarrow 0xFFFFFFFF)$ (For C/C++, it's decimal (-3)).

Data chunking

The maximum chunk size of the data that shall be sent on the serial channel is predefined as 1024 bytes (1 KB).

Before actually sending the data, based on the data size the chunk is produced. If the data size is more than the predefined maximum chunk size, then more than one chunk will be produced and packet encapsulation is applied to each chunk, if applicable. Then each chunk is sent to the physical channel sequentially (not writing all chunks in one go).

Similarly, after receiving the data, the process should be reversed, i.e., the encapsulated packet structure will be removed from each chunk received sequentially, and the data from each chunk will be combined into a single final data, which is then passed on to the above layers.

Section 3 : TCP/SSL/UDP Remote Messages

Remote Message Overview

The Access and Time Biometric 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 Access and Time Biometric Terminal to dialog with a controller, and how to make use of them.

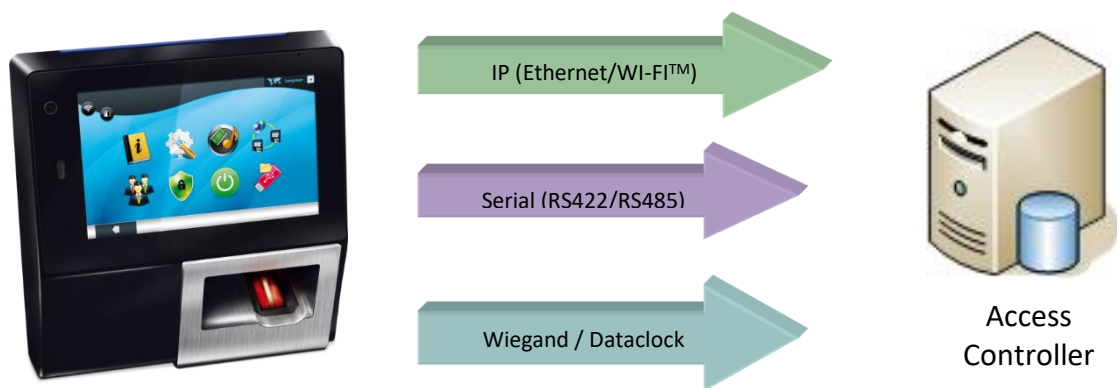


Figure 2: Remote Message Overview

Supported Protocols

The Access and Time Biometric 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
Wiegand <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. Or The ID of the identified user or zero ID is sent if the frame format for failure event is configured.
Dataclock <i>The terminal acts as a magnetic badge reader.</i>	The ID (ISO2) of the identified user is sent.	Nothing is sent. Or numerical ID describing the cause of the failure.
RS485/422	Identification result is sent.	The biometric check result (failure) is sent.
UDP	Complete identification result is sent.	The biometric check result (failure) is sent.
TCP		
SSL		

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 Series or MorphoWave® Compact **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.

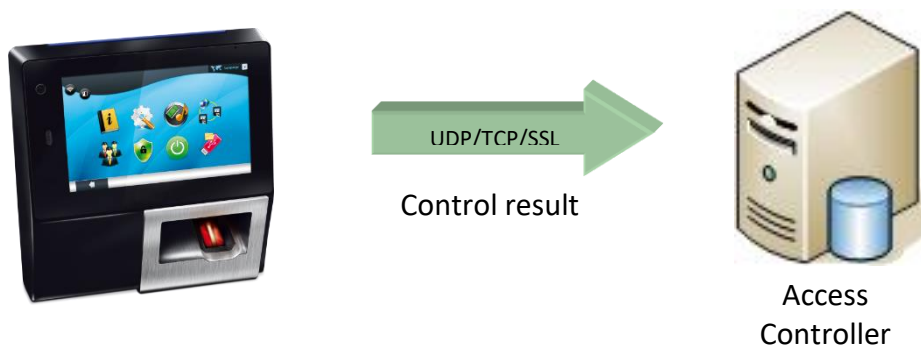


Figure 3: Ethernet or Wi-Fi™, UDP, TCP or SSL protocols

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

When using TCP/SSL protocol, Controller could provide feedback using specific messages format (described in the dedicated section of this document).



Feedback messages are not available when using UDP protocol.

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.

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

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 Access and Time Biometric 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 Access and Time Biometric 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 Access and Time Biometric 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 Access and Time Biometric 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 Access and Time Biometric 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 Note: 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 (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 Note: 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 (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:

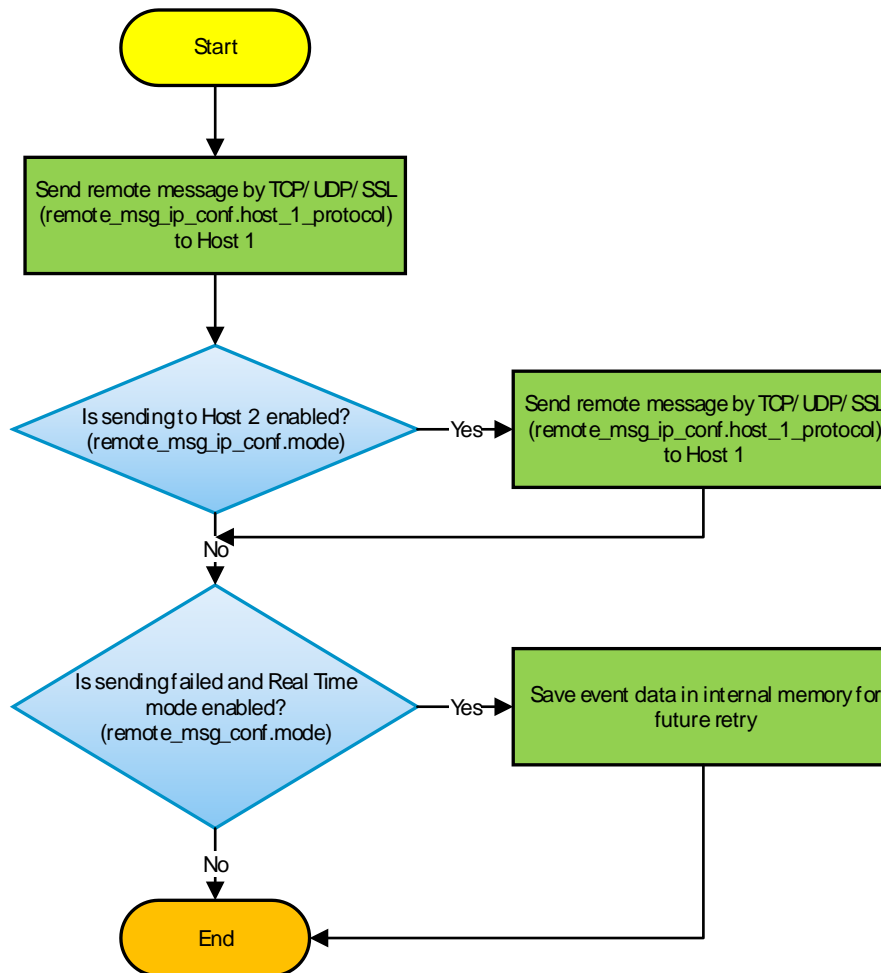


Figure 4: Remote Message Flow when response from controller is not required

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

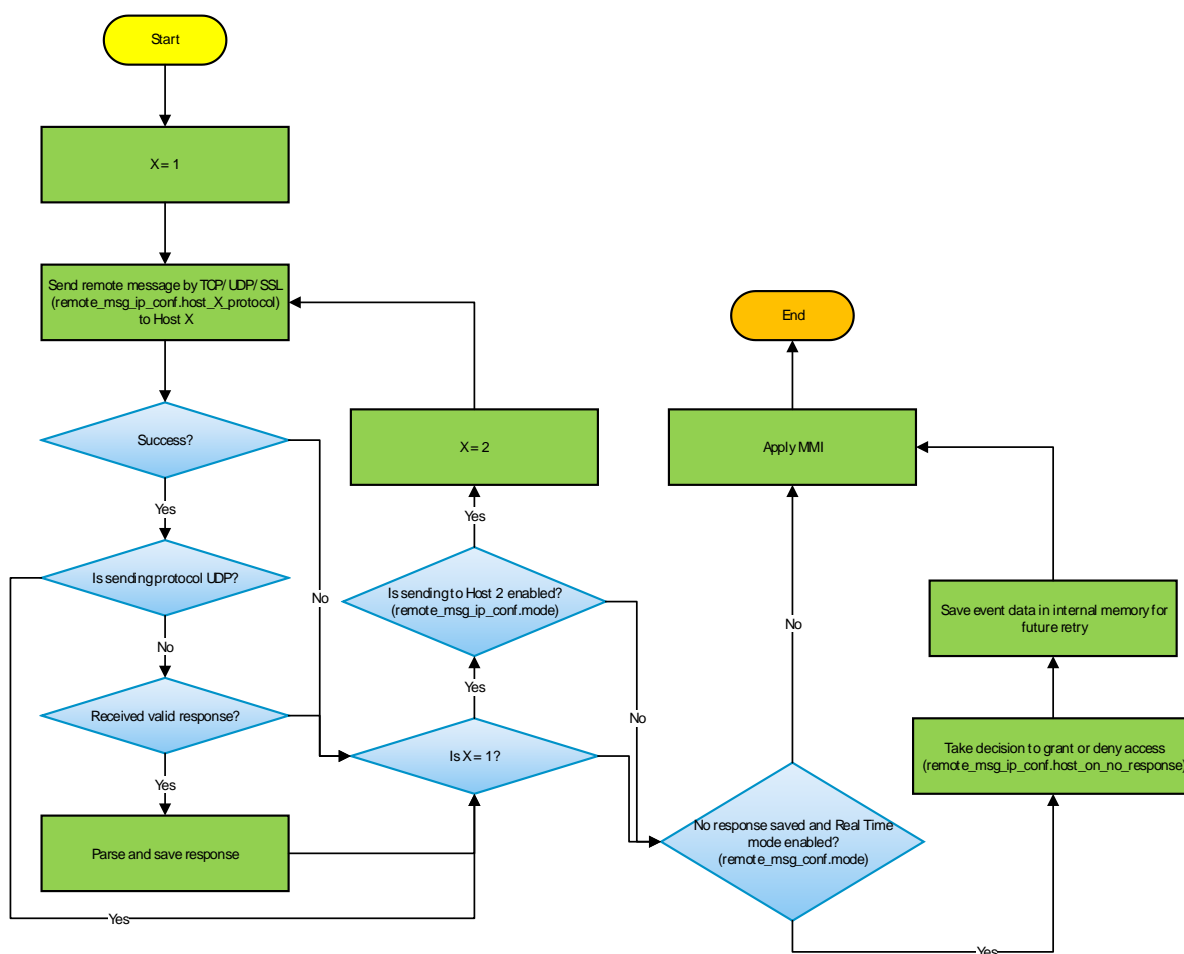


Figure 5: Remote Message Flow when response from controller is required

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](#):

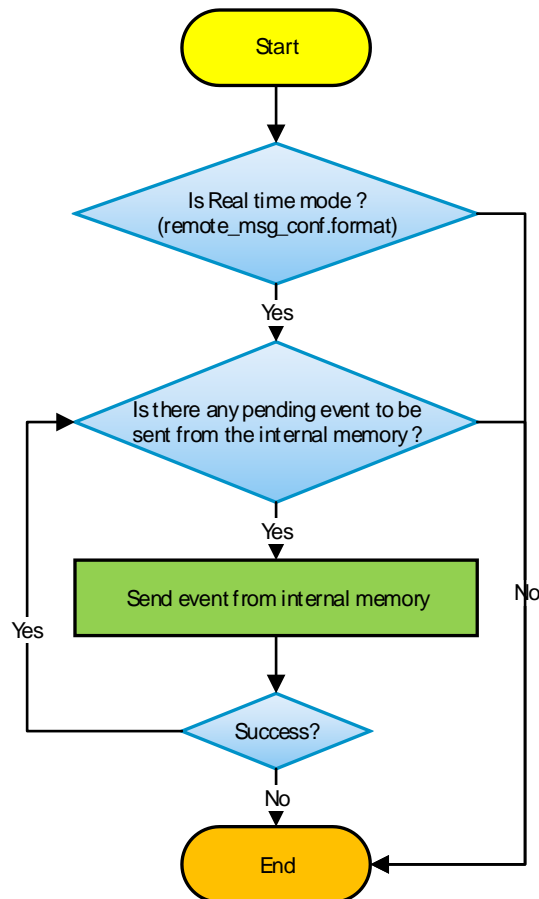


Figure 6: Remote Message Flow for stored events in Real Time mode

Section 4 : Serial Remote Messages

Serial Remote Messages

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

RS485 Presentation

The Access and Time Biometric Terminal can also send information through the serial link using RS485 protocol. Several terminals can be connected on the same serial link.

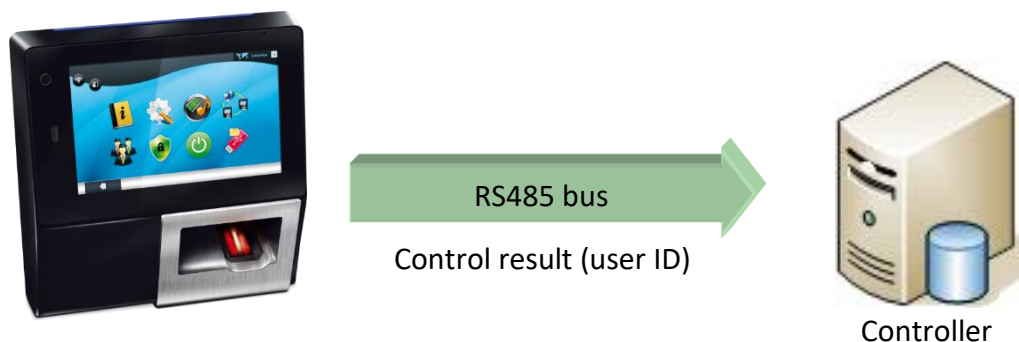


Figure 7: Serial port, RS485 protocol

RS422 Presentation

The Access and Time Biometric Terminal can send information through the serial link using RS422 protocol.

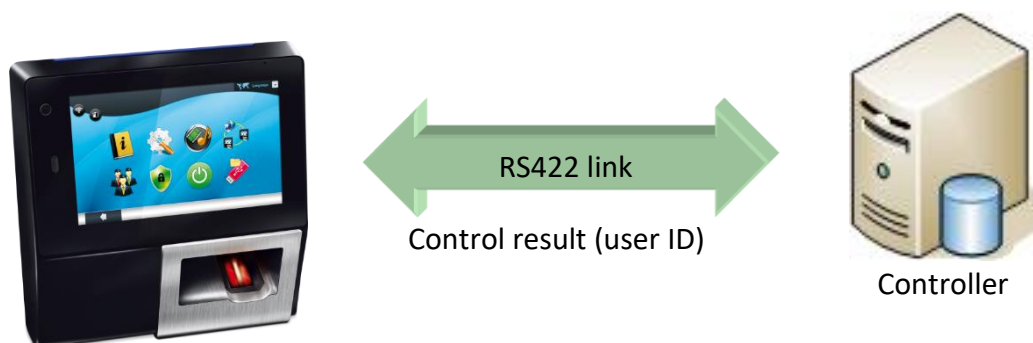


Figure 8 : Serial port, RS422 protocol

Activation

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

<i>remote_msg_conf.send_serial_state</i>	
0	Messages are not sent.
1	Messages are sent.

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

<i>remote_msg_serial_conf.format</i>	
422	RS422 protocol (For all Biometric terminals except SIGMA Lite & SIGMA Lite+)
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.

<i>Serial_communication.net_id</i>	
0-255	Net ID is required to identify a terminal in a network (serially connected) of several terminals

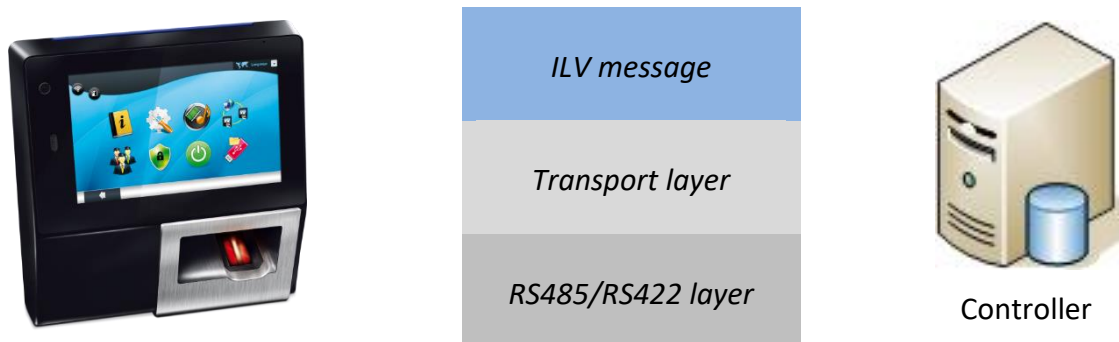
- Terminal identifier values such as DLE [0x1B], XON [0x11] or XOFF [0x13], are forbidden.

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® SIGMA Series, MorphoAccess® Extreme Series, *MorphoWave*® Compact and VisionPass 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 (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 **“Access and Time Biometric Terminals Distant Commands Guide”**.

Section 5 : Wiegand / Dataclock Remote Messages

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 Access and Time Biometric Terminal can handle multiple data format for various output events (e.g. user control success, user control failure, duress, tamper).

Access and Time Biometric 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

<code>wiegand.event_identify_pass</code>	
<code>wiegand.event_verify_pass</code>	
-1	No specific format 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. E <128 bit serial number> O Contains 128 bit device serial number and two parity bits. Even parity (bit 0) will be calculated from bits 1-64 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	
	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)	
	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 6 : Remote Message Frames

Message Format

All the exchanged messages over IP between the Access and Time Biometric Terminal and the remote controller have the same structure. These messages are called **ILV** (Identifier, Length, and Value).

<i>ILV messages</i>		
Identifier	Length	Value
1 byte	2 bytes	Length bytes
<i>Message identifier</i>	<i>Message data length (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

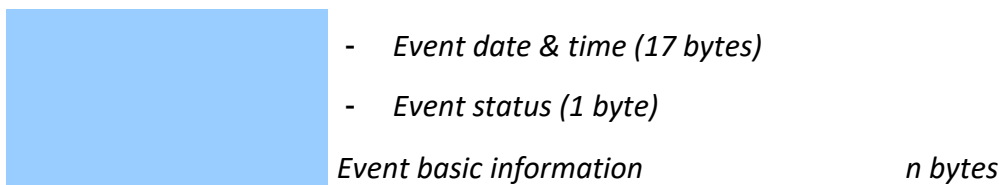
Messages in Basic format contain basic information about the event.

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
	- Terminal Serial Number (14 bytes)	



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

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	0x10: User ID is sent in ASCII format and 1 byte control failed.	
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 Access and Time Biometric 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 Access and Time Biometric 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	0xC1 : 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® 1 bytes SIGMA Family]</i>	

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 Series or MorphoWave® Compact terminal or VisionPass 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	0x70: <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	0x71 : <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	0x72 : <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	0x73 : <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	0x74 : <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	0x75 : <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	0x76 : <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	0x77 : <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	0x78: <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	0x79: <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	0x7A: <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	0x7B : <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	0x7C: <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	0x7D : <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	0x7E : <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	0x80: <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	0x81: <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&A only)</i>	<i>1 byte</i>
	<i>MA date and hour (T&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	0x82: <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	0x83: <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	0x84 : <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	0x85: <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&A only)</i>	1 byte
	<i>MA date and hour (T&A only)</i>	17 bytes

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	0x86 : <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	0x50 : 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 Access and Time Biometric Terminal will apply MMI as per the configuration.

0xFF: Access is denied: the Access and Time Biometric Terminal will apply MMI as per the configuration.

0x01 or any other value: nothing happens. Access and Time Biometric 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 Access and Time Biometric 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 Access and Time Biometric Terminals 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 Access and Time Biometric Terminal
- 1 = Access Denied sound for Access and Time Biometric Terminal
- 2 = Access Granted sound for Access and Time Biometric Terminal

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 other 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, MorphoWave Compact and VisionPass
- 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 Access and Time Biometric 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 Access and Time Biometric Terminal 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 Access and Time Biometric 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 Access and Time Biometric 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	

Annex 1 : Wiegand Data Format

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

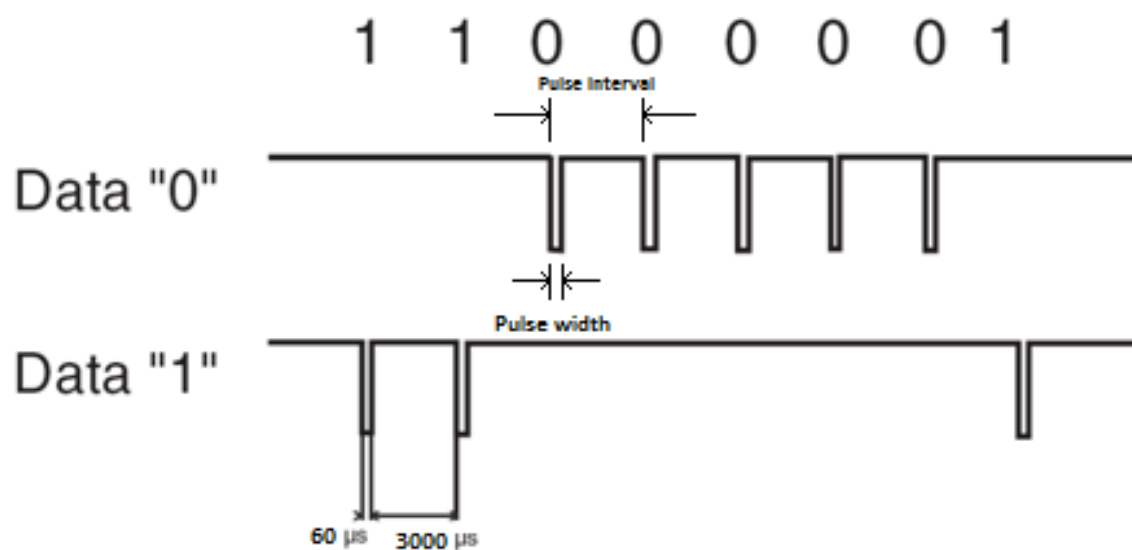


Figure 9: Wiegand frame format

Annex 2 : ISO 7811/2 - 1995 - Track 2 DataClock 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:

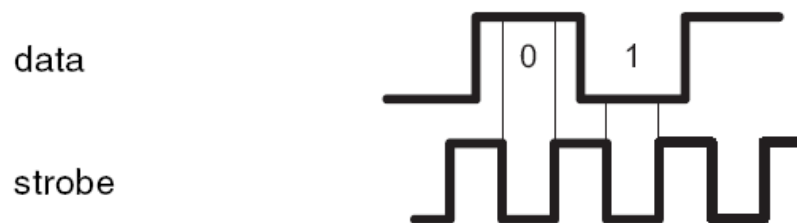


Figure 10: Data Clock signals

Other modes are (only for output):

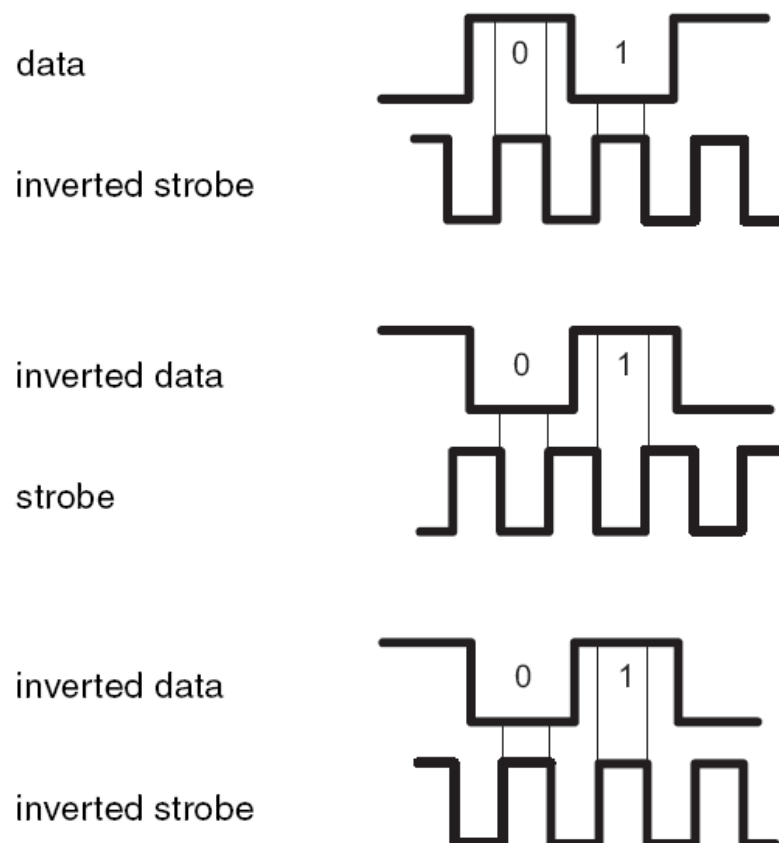


Figure 11: Other Data Clock signals

Annex 3 : RS485 Protocol

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	0x02
<ID>	Packet Identifier	1	0xE1
<TID>	Terminal Identifier	1	--
<DATA>	Data value	Up to 1024	--
<CRC>	Transmission error control	2	--
<DLE>	Data Link Escape	1	0x1B
<ETX>	End Text	1	0x03

- 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 ('unstuffing') 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:

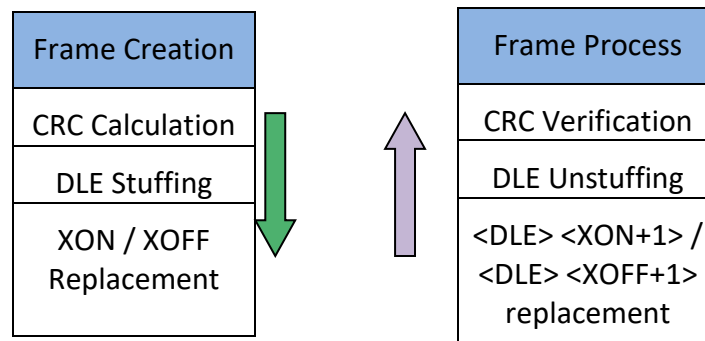


Figure 12 : RS485 frame processing

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

Annex 4 : RS422 Protocol

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	0x02
<ID>	Packet Identifier	1	--
<RC>	Request Counter	1	--
<DATA>	Data value	Up to 1024	--
<CRC>	Transmission error control	2	--
<DLE>	Data Link Escape	1	0x1B
<ETX>	End Text	1	0x03

- 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 Access and Time Biometric terminal.**
- An IN packet is a packet sent **by the Access and Time Biometric 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:

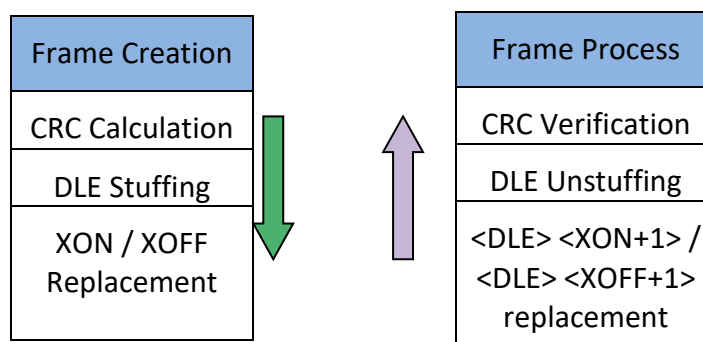


Figure 13: RS422 frame processing

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:

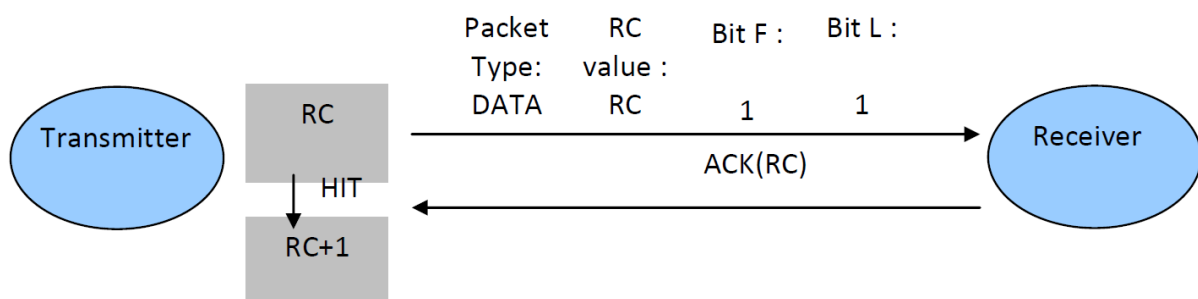
- In case of reception of a NACK or Timeout, the transmitter tries to send the same packet again.
- A packet can be retransmitted 3 times. After that, another NACK reception or Timeout leads to ERROR of the transmission.

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:

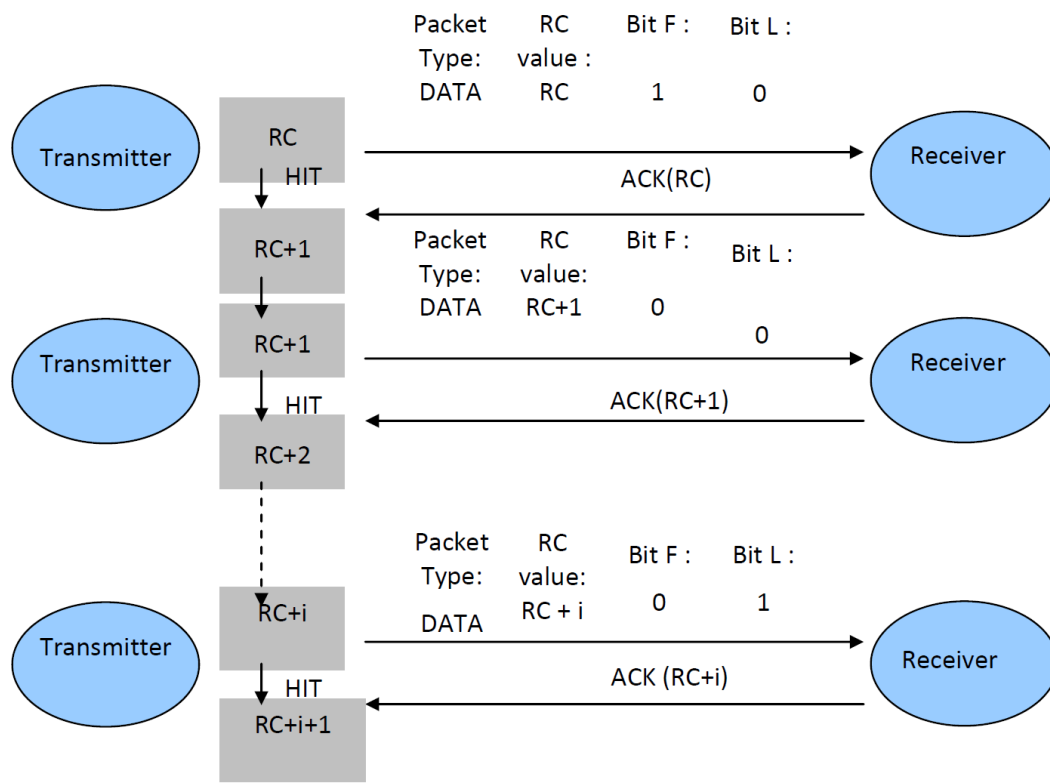


Figure 14: RS422 typical frame workflow

- An Error occurred while transmitting the data packet:

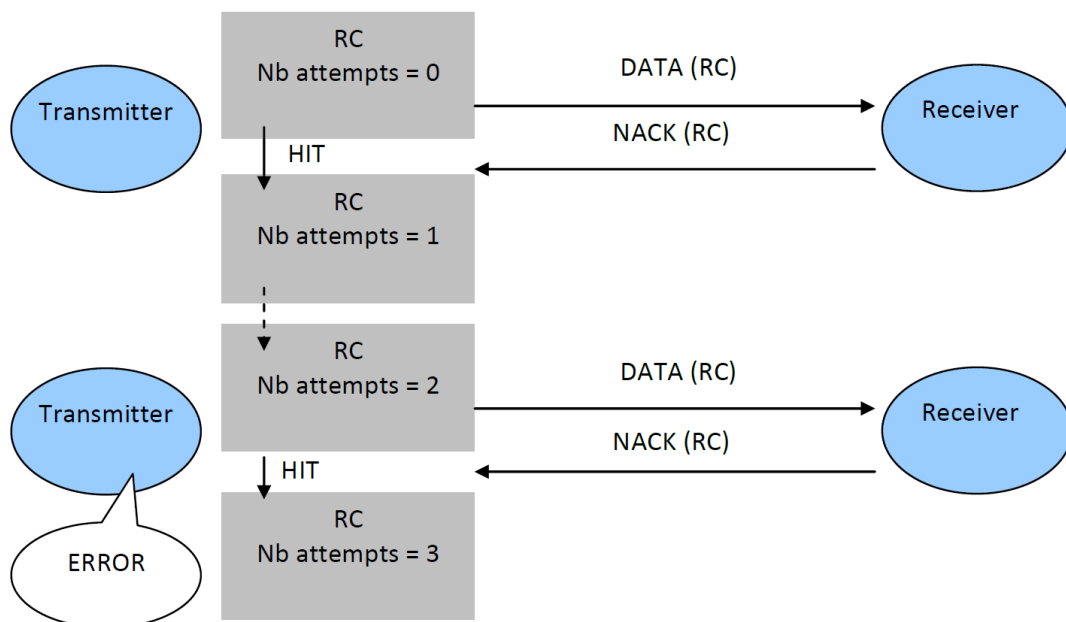


Figure 15: RS422 transmission error

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

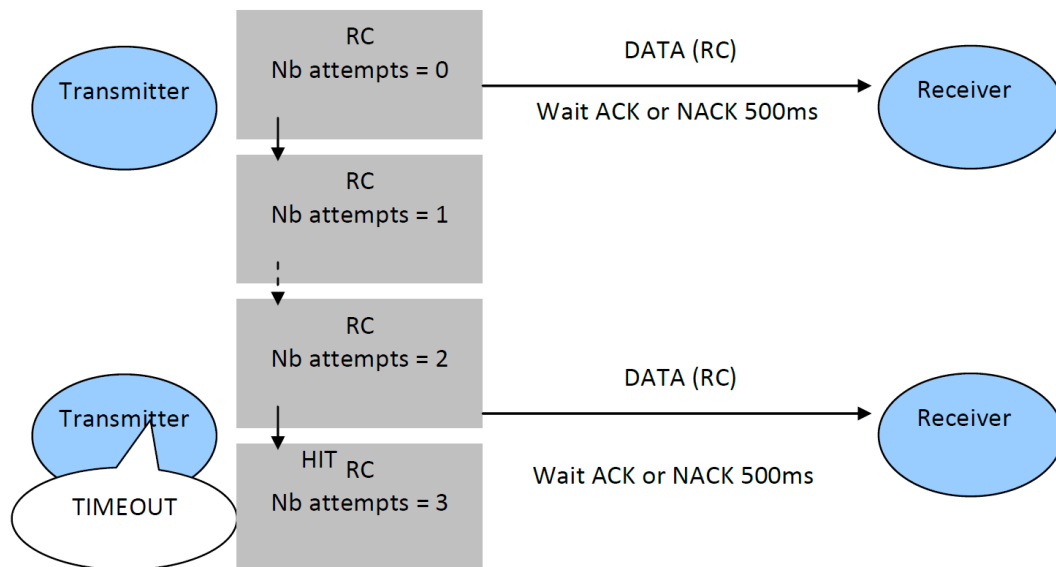


Figure 16: RS422 transmission timeout error

Examples:

- User "094066" has been recognized by an Access and Time Biometric 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 an Access and Time Biometric 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 Access and Time Biometric 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.

Annex 5 : Bibliography

How to get latest version of the documents

The last version of the documents 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

Annex 6 : Glossary, Acronyms and Abbreviation

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 Access and Time Biometric Terminal
- **Device:** This term is used for an external device attached to Access and Time Biometric 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 Access and Time Biometric 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 Access and Time Biometric 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

Annex 7 : Support

Technical Support and Hotline

North America

Mail: support.bioterminals.us@idemia.com

Tel: +1 888 940 7477

South America

Mail: support.bioterminals.us@idemia.com

Tel: +1 714 575 2973

Asia, Pacific:

Mail: support.bioterminals.in@idemia.com

Tel: +91 1800 120 203 020

Europe, Middle-East, Africa

Mail: support.bioterminals@idemia.com

Tel: +33 1 30 20 30 40

Web site

For the latest firmware, software, document releases, and news, please check our websites :

www.biometric-terminals.com

(To get your log in and password please contact your sales representative).



Head office :

IDEMIA

2, place Samuel de Champlain

92400 Courbevoie France

www.idemia.com