Sagem Défense Sécurité
SAFRAN Group

MorphoAccess™ 100 Series

Remote Messages Specification

Produced by Sagem Défense Sécurité

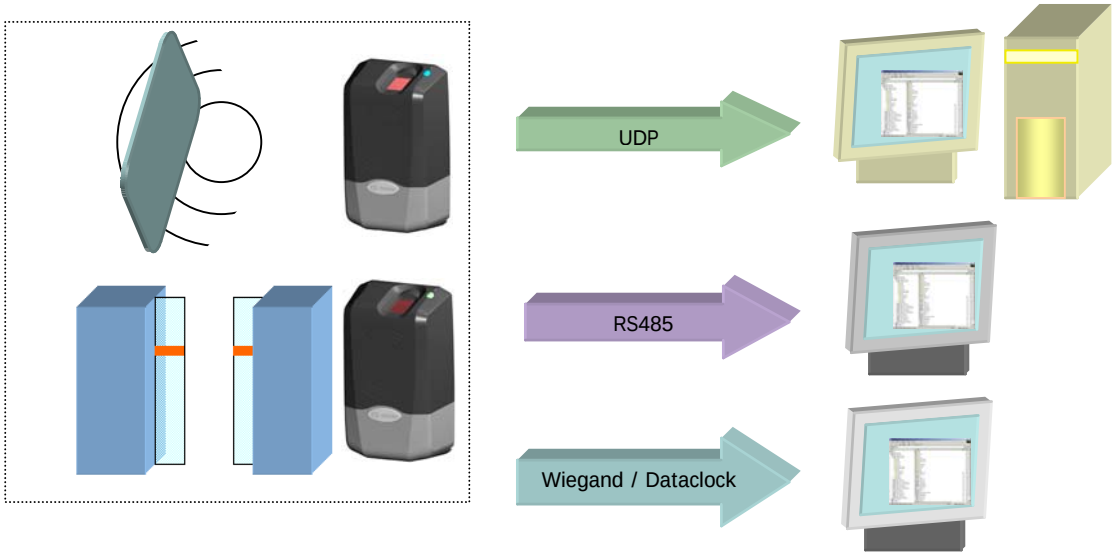
www.sagem.com

Table of Content

OVERVIEW

The MorphoAccess™ terminal can send status messages in real time to a controller by different means and through different protocols. This information, called ***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…depending on the role of the controller in the system.

This document describes the different solutions offered by the MorphoAccess™ to dialog with a controller, and how to make use of them.



REFERENCES

Reading the following manuals may be useful to understand the functionalities presented in this document:

- SAGEM MorphoAccess™ Installation Guide.
- SAGEM MorphoAccess™ Host System Interface Specification.

SUPPORTED PROTOCOLS

Messages about the biometric operations performed by the MorphoAccess™ can be sent by the terminal to a controller through the following protocols:

- Wiegand
- Dataclock
- RS485
- Ethernet (UDP)

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 Wiegand or Dataclock: failure messages.

	Verification OK	Verification KO
Wiegand *The terminal acts as a magnetic badge reader.*	The ID of the identified user is sent. The frame format can be configured.	Nothing is sent. Or numerical ID describing the cause of the failure.
Dataclock *The terminal acts as a magnetic badge reader.*	The ID (ISO2) of the identified user is sent.	Nothing is sent. Or numerical ID describing the cause of the failure.
RS485	Complete identification result is sent.	The biometric check result (failure) is sent.
UDP	Complete identification result is sent.	The biometric check result (failure) is sent.

REMOTE MESSAGES ACTIVATION

CONFIGURATION OF THE PROTOCOLS

Remote messages can be sent by the MorphoAccess™ terminal on several layers at the same time (for example: Wiegand and Ethernet). This chapter explains how to activate the sending of the remote messages for each layer available: Wiegand/Dataclock, Serial, Ethernet.

The configuration of each protocol requires modifying some parameters. It means if you don't know how to perform such operation, please refer to the *User Guide*.

Parameters can be changed using remote management commands.

RS485 – WIEGAND – DATACLOCK

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

Priority is given to *Wiegand* then *Dataclock*, then *RS485*.


It means activating RS485 with Wiegand enabled will have **no effect** on the RS485 layer.

WIEGAND REMOTE MESSAGES

PRESENTATION

The payload data encapsulated in a Wiegand 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 Wiegand or Dataclock: failure messages.

The MorphoAccess™ led can also be driven by two *Led In* signals, one for “red” another for “green”: this feature improves integration in an access control system and is described in *MA1XX User Guide*.

ACTIVATION

A configuration entry allows enabling the Wiegand output.

app/send ID wiegand/enabled	
0	ID is not sent (default)
1	ID sent. ID format is defined using specific parameters

SETTING UP WIEGAND INTERFACE

When set up to communicate with Wiegand protocol, the MorphoAccess™ can handle multiple data format.

Default format (26-bit) is described in Appendix 1: Wiegand Data Format.

The Wiegand frame format is defined using six configuration keys. Different protocol can be defined for input and output.

Wiegand frame timings are not customizable. Additional security (ciphering) is not handled. All Wiegand protocols are reverse.

Frame definition

Here after are listed the customizable parameters of a Wiegand frame.

Length

A Wiegand frame can contain up to 128 bits.

Control bits

In a Wiegand frame, start and stop bits are used as control bits. They can be fixed to 0 or 1 or be used as parity (odd or even) bits calculated over bits of the frame.

Data

In the Wiegand protocol, three data are handled: the *Site Code* (also called *Facility Code* or *Comparison Number*), the *ID* (also called *Badge Number* or *Sequence Number*) and a *Custom Data*. Data can have a variable bit size and can be located anywhere in the frame. Data are inserted in the frame MSB first.

app/send ID wiegand/frame length	
1-128	Defines the number of bits of the frame.
app/send ID wiegand/start	
	Defines the start control bit.
0.0	Reset to 0.
1.0	Set to 1.
2.n	Even parity calculated over the n first bits.
3.n	Odd parity calculated over the n first bits.
4.0	No start bit.
app/send ID wiegand/stop	
	Defines the stop control bit.
0.0	Reset to 0.
1.0	Set to 1.
2.n	Even parity calculated over the n last bits.
3.n	Odd parity calculated over the n last bits.
4.0	No stop bit.
app/send ID wiegand/site	
n.m	Insert m bits of site value at offset n.
app/send ID wiegand/ID	
n.m	Insert m bits of ID value at offset n.
app/send ID wiegand/custom	
0.0	Reserved for SAGEM custom protocols.

WIEGAND « ANY BIT » EXAMPLES

26-bit: default format

```
/Length:     26

/Start: 2.12 //even parity calculated over the first 12 bits

/Site: 1.8         //1-byte facility code inserted in first

/ID: 9.16          //2-byte ID inserted in second

/Custom: 0.0

/Stop: 3.12        //odd parity calculated over the last 12
bits
```

34-bit

```
/Length:     34

/Start: 0.0  //reset to 0

/Site: 1.16  //2-byte comparison number inserted in first

/ID: 17.16   //2-byte sequence number inserted in second

/Custom:     0.0

/Stop: 1.0         //set to 1
```

37-bit: HID format

```
/Length:     37

/Start: 2.18 //even parity calculated over the first 18 bits

/Site: 0.0

/ID: 1.35          //35-bit ID (max value is 4294967295)

/Custom:     0.0

/Stop: 3.18        //odd parity calculated over the last 18
bits
```

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 Wiegand or Dataclock: failure messages.

DataClock frame content is described in section Appendix 3 -Iso 7811/2-1995 - Track 2 Dataclock Format.

ACTIVATION

A configuration entry allows enabling the Dataclock output.

app/send ID dataclock/enabled	
0	ID is not sent (default)
1	ID sent

WIEGAND OR DATACLOCK: FAILURE MESSAGES

PRESENTATION

Failure ID option allows sending extended error codes through the Wiegand or Dataclock layer. You can activate this option and associate any value between 0 to 65535 for each existing failure case.

This feature has no impact on the Ethernet and Serial remote messages.

Note that the administrator has to check that the Identifier is not already stored in the database.

SOFTWARE CONFIGURATION

app/failure ID/failure ID enabled	
0	Nothing is sent through Wiegand or Dataclock when biometric control fails.
1	A specific identifier is sent through Wiegand or Dataclock when biometric control fails.

Specific identifier can be associated with a given failure case.

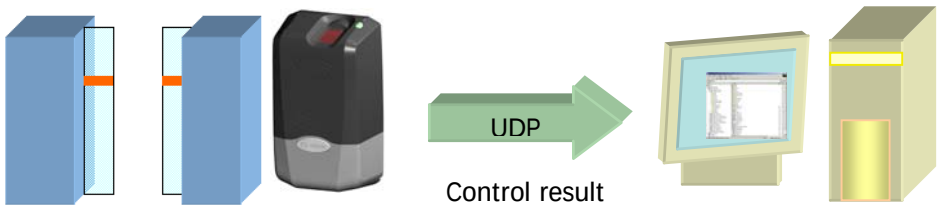
app/failure ID/not recognized ID	
0 - 65535	This value is sent when a user is not identified (i.e. a biometric operation has failed).
app/failure ID/timeout ID	
0 - 65535	This value is sent when the identification/verification operation aborts due to a timeout error.
app/failure ID/not in DB ID	
0 - 65535	This value is sent when no record can be found in the database for the specified user id (i.e. no biometric operation can be performed).
app/failure ID/generic error ID	
0 - 65535	This value is sent when any other biometric error occurs.
app/failure ID/alarm ID	
0 - 65535	This value is sent when the back cover is removed (tamper switch detection).

ETHERNET (UDP) REMOTE MESSAGES

*In this mode the terminal acts as a **client** and sends UDP information to the PC that is a **server**.*

PRESENTATION

The MA^{1XX} can send information using UDP.



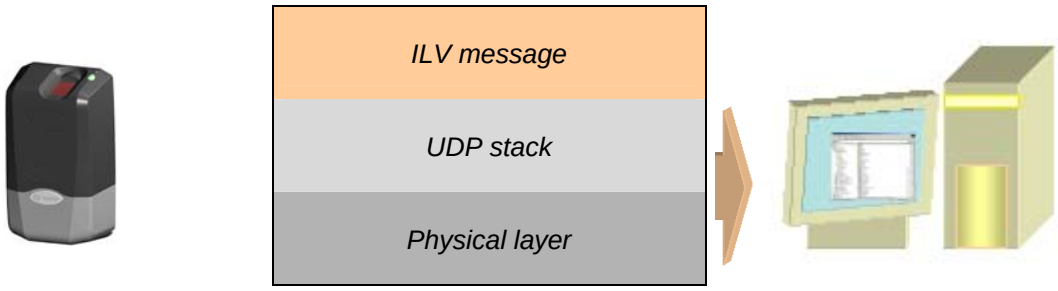
ACTIVATION AND SETTINGS

A configuration entry allows enabling the UDP remote messages.

app/send ID UDP/enabled	
0	Messages sending deactivated on the Ethernet link.
1	Messages sending activated on the Ethernet link.
app/send ID UDP/host address or name	
”10.10.161.39“ or ” Hostname”	It defines the destination computer on the networks. It can be defined using IP or hostname.
app/send ID UDP/host port	
11020	Ethernet messages are sent from the port 11020 through UDP protocol, but the port can be modified.

MESSAGE FORMAT

Remote information are sent on the UDP layer.



Messages sent though UDP have the following format named ILV.

ILV messages		
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 byte,
- **Value** called V; this is the data or parameters.

This data structure is variable. Its length is variable. Message data will depend on the biometric control result.

UDP - MESSAGE SENT WHEN CONTROL IS OK

Description

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

Command

Frame sent

Identifier value	**0x00**: User ID is sent in ASCII format and control succeeded.	1 byte
Length value	L	2 bytes
Value (Parameters)	User ID	L bytes

User ID

User identifier in ASCII. “94066” for example.

Example

This frame means user “528610” has been recognized.

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”					

UDP - MESSAGE SENT WHEN CONTROL FAILED

Description

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

Command

Frame sent

Identifier value	**0x10**: 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

Biometric Error Code

Failure: CONTROL_FAILED [0x01]

“Timeout”: CONTROL_TIMEOUT [0x19]

User not in base: LOG_NOT_IN_BASE [0x12]

Generic error: LOG_ IDENT_ERROR [0xFF]

User ID

The user ID is sent if the MorphoAccess™ works in authentication mode (with contactless card).

Examples

Identification mode: this frame means that identification failed.

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.

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”					

UDP – TAMPER SWITCH ALARM

Description

This frame is sent to the controller when the tamper switch is activated.

The device can send an alarm trough UDP communication port. It can also play a sound alarm whilst sending the alarm.

To send an alarm in UDP, the key *app/send ID UDP/enabled* must be set to 1, otherwise no alarm will be sent, even if the key *app/tamper alarm/level* is set to send an alarm.

After enabling the UDP message, the key *app/tamper alarm/level* must be set to 1 (Send Alarm) or 2 (Send Alarm and Buzzer).

After setting these two keys the terminal is ready to send an alarm through UDP in case of intrusion detection.

Command

Frame sent

Identifier value	**0xC1**: ILV_ALARM_ID.	1 byte
Length value	0x04	2 bytes
Value (Parameters)	RFU	4 bytes

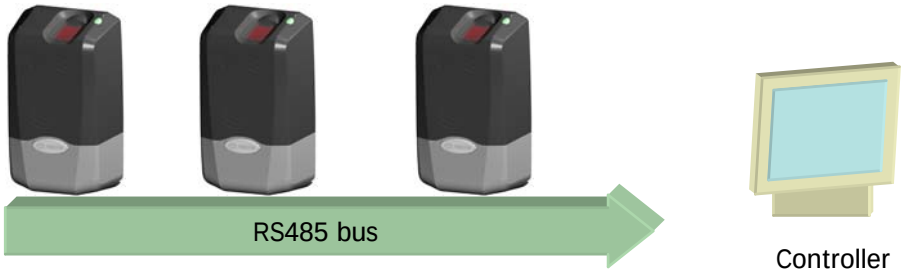
RFU

These 4 bytes must be set to 0.

SERIAL REMOTE MESSAGES

PRESENTATION

The MA^{1XX} can also send information through the serial link (RS485) on a RS485 bus.



ACTIVATION

A configuration entry allows enabling the RS485 remote messages.

app/send ID RS485/enabled	
0	Messages sending deactivated on the serial link.
1	Messages sending activated on the serial link.

TERMINAL IDENTIFIER

This parameter allows setting an address defining a MA^{1XX} on the RS485 network.

app/send ID RS485/terminal identfier	
0-255	This value defines a MA^{1XX} on the RS485 network.

Terminal identifier set to DLE[0x1B], XON[0x11] or XOFF[0x13] is forbidden.

This *Terminal Identifier* can be retrieved in the low layer serial protocol described in Appendix 2: Serial Protocol.

SERIAL PORT SETTINGS

It is possible to set the baud rate, the data and stop bits size and to select the parity type.

app/send ID RS485/baudrate
1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600 or 115200 bps

app/ send ID RS485/databits

7 or 8 databits

app/ send ID RS485/parity	
0	No
1	Odd
2	Even

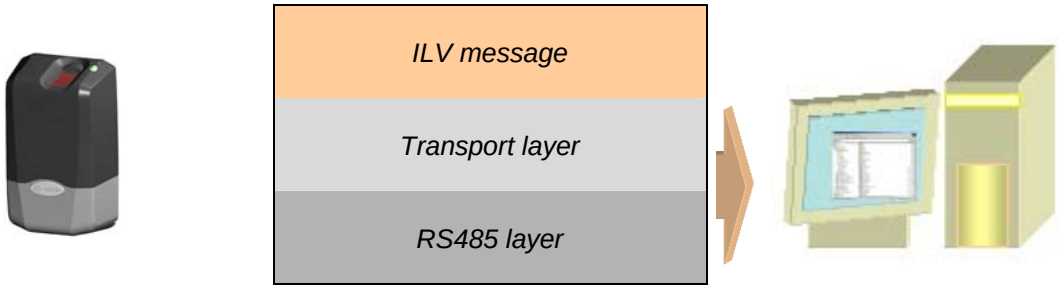
app/ send ID RS485/stopbits
1 or 2 stop bits.

It is possible to reemit the frame in case of failure. The terminal uses an anti-collision protocol to prevent collisions on the bus.

The serial protocol is described in Appendix 2: Serial Protocol.

MESSAGE FORMAT

Remote information are sent on the RS485 serial layer. The transport layer ensures the transmission.



Messages sent though RS485 have the following format named ILV.

ILV messages		
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 byte,
- **Value** called V; this is the data or parameters.

This data structure is variable. Its length is variable. Message data will depend on the biometric control result.

The transport layer is described in Appendix 2: Serial Protocol.

The transport layer ensures frame acknowledge If *app/ send ID RS485/nb try* is set to 0 no “ACK” is waited.

RS485 - MESSAGE SENT WHEN CONTROL IS OK

Description

This frame is sent to the controller when the user is recognized. The format of the ID can be specified.

Command

Frame sent

Identifier value	**0x00**: User ID is sent in ASCII format	1 byte
Length value	L	2 bytes
Value (Parameters)	User ID	L bytes

User ID

User identifier:

- “27321” will be 0x32 0x37 0x33 0x32 0x31 in ASCII.

RS485 - MESSAGE SENT WHEN CONTROL IS NOK

Description

This frame is sent to the controller when the control failed. The format of the ID can be specified.

Command

Frame sent

Identifier value	**0x10**: User ID is sent in ASCII format	1 byte
Length value	1+L	2 bytes
Value (Parameters)	Biometric Error Code	1 byte
	User ID (according to the configuration)	*L bytes*

Biometric Error Code

Failure: CONTROL_FAILED [0x01]

“Timeout”: CONTROL_TIMEOUT [0x19]

User not in base: LOG_NOT_IN_BASE [0x12]

Generic error: LOG_ IDENT_ERROR [0xFF]

User ID

User identifier (returned for a verification only):

- “27321” will be 0x32 0x37 0x33 0x32 0x31 in ASCII.

RS485 – TAMPER SWITCH ALARM

Description

This frame is sent to the controller when the tamper switch is activated.

The device can send an alarm trough RS485 port. It can also play a sound alarm whilst sending the alarm message

To send an alarm, the key *app/send ID RS485/enabled* must be set to 1, otherwise no alarm will be sent, even if the key *app/tamper alarm/level* is set to send an alarm.

Because wiegand, dataclock, and rs485 are multiplexed on the same lines, only one of these protocol shall be enabled at one time, else priority is given to *Wiegand* then *Dataclock*, then *RS485*.

After enabling the RS485, the key *app/tamper alarm/level* must be set to 1 (Send Alarm) or 2 (Send Alarm and Buzzer).

After setting these two keys the terminal is ready to send an alarm through RS485 in case of intrusion detection.

Command

Frame sent

Identifier value	**0xC1**: ILV_ALARM_ID..	1 byte
Length value	0x04	2 bytes
Value (Parameters)	RFU	4 bytes

RFU

These 4 bytes must be set to 0..

APPENDIX 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.

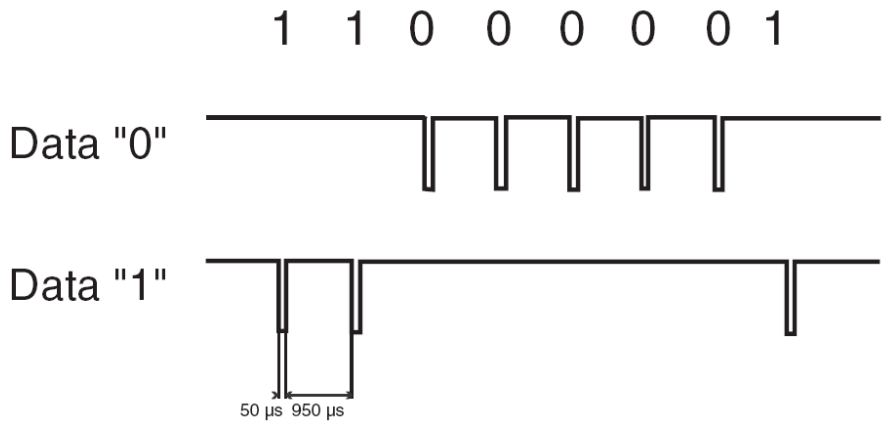
The 16 other bits are data bits.

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.



APPENDIX 2: SERIAL PROTOCOL

DEFINITION

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	**0x61**
<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 Little Endian: multi bytes data are sent least significant byte first (LSB).

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.

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

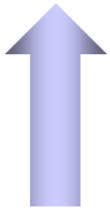
The packet identifier byte is 0x61.

Terminal Identifier

The terminal identifier byte defines the MA^{1XX} “address» on the RS485 network.

FRAMES SEQUENCE

Frame creation
CRC calculation
DLE Stuffing
XON / XOFF replacement



Frame process
CRC verification
DLE Unstuffing
<DLE> <XON+1> / <DLE> <XOFF+1> replacement

FRAME EMISSION

3 attempts are made in case of failure.

EXAMPLES

User “094066” has been recognized by a MA^{1XX} number 89 (TID).

STX	ID	TID	Data: ILV									CRC	CRC	DLE	ETX
02	61	59	00	06	00	*30*	*39*	*34*	*30*	*36*	*36*	CE	D1	1B	03

User “62487” has been recognized by a MA^{1XX} number 89 (TID).

STX	ID	TID	Data: ILV								CRC	CRC	DLE	ETX
02	61	59	00	05	00	*36<*	*32*	*34*	*38*	*37*	A0	AA	1B	03

Identification failed on MA^{1XX} number 89 (TID).

STX	ID	TID	Data: ILV				CRC	CRC	DLE	ETX
02	61	59	10	01	00	01	B6	3C	1B	03

APPENDIX 3 -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	Llyll
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

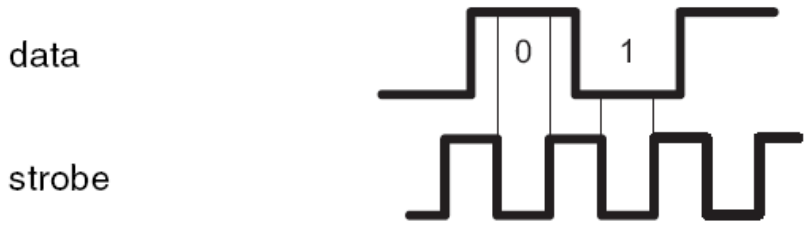
The LRC is calculated by the following procedure: each of the 4 bits in the LRC character is an even parity bit of the equivalent bits in the telegram including start and stop sentinel.

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

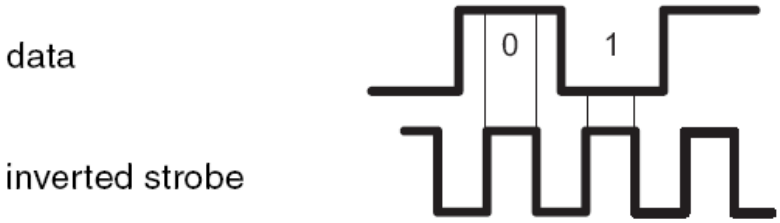
Input data should be preceded and followed by a clock synchronization pattern (NULL data).

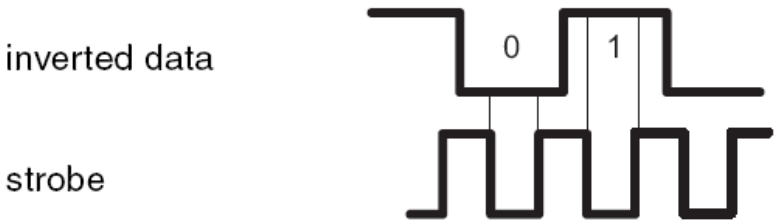
DATACLOCK LEVELS

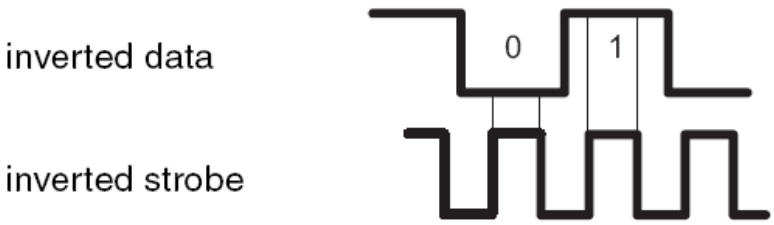
In normal operation mode (default) input and output signals are defined:



Other modes are (only for output):







Siège social : Le Ponant de Paris

27, rue Leblanc - 75512 PARIS CEDEX 15 - FRANCE