STid
Electronic Identification

SEGIC

Designed in France
Made in France

www.stid.com

Acknowledgment

You have purchased SEGIC software; it will allow you to configure Legic reader.

We thank you for the confidence you place in us and hope that this solution developed by STid will satisfy you.

We remain at your disposal for any questions about using this software or our range of products.

We look forward to seeing you for more information on our website www.stid.com.

STid Team

Introduction

SEGIC is software dedicated to the reader configuration ARC Legic in Read Only.

SEGIC does not allow the encoding of user cards.

Some user cards are protected reading, writing or both by a master badge called SAM that once accepted by the reader allows access to the user data secure on card. Without SAM, reader will read only plain data and UID.

Reader retrieve configuration from SAM either time (to point out that reader successfully gets settings from SAM, reader beeps 5 times and LED is blinking quickly).

Informations

1 - PC requirements

- A PC with Windows XP, VISTA, 7 or 8.x operating system.
- An USB or RS232 available communication port.

2 – USB key Content

- FTDI USB Driver for Windows XP, VISTA, 7 and 8.x.
- SEGIC Version 1.1.x.

4 - Installation

- Insert the SEGIC USB key in an USB port of your PC.
- Wait for the automatic opening of the browser window.
- Or if the automatic opening doesn't start, launch SEGICV1.1.xx_setup.exe located in USB key.
- Follow the instructions on the screen.

Warning

Make sure that the installation is done with administrator rights

If you don't have administrator rights: Right click on SEGICV1.1.xx_setup.exe and "Run as administrator".

5 – Reader's Compatibility

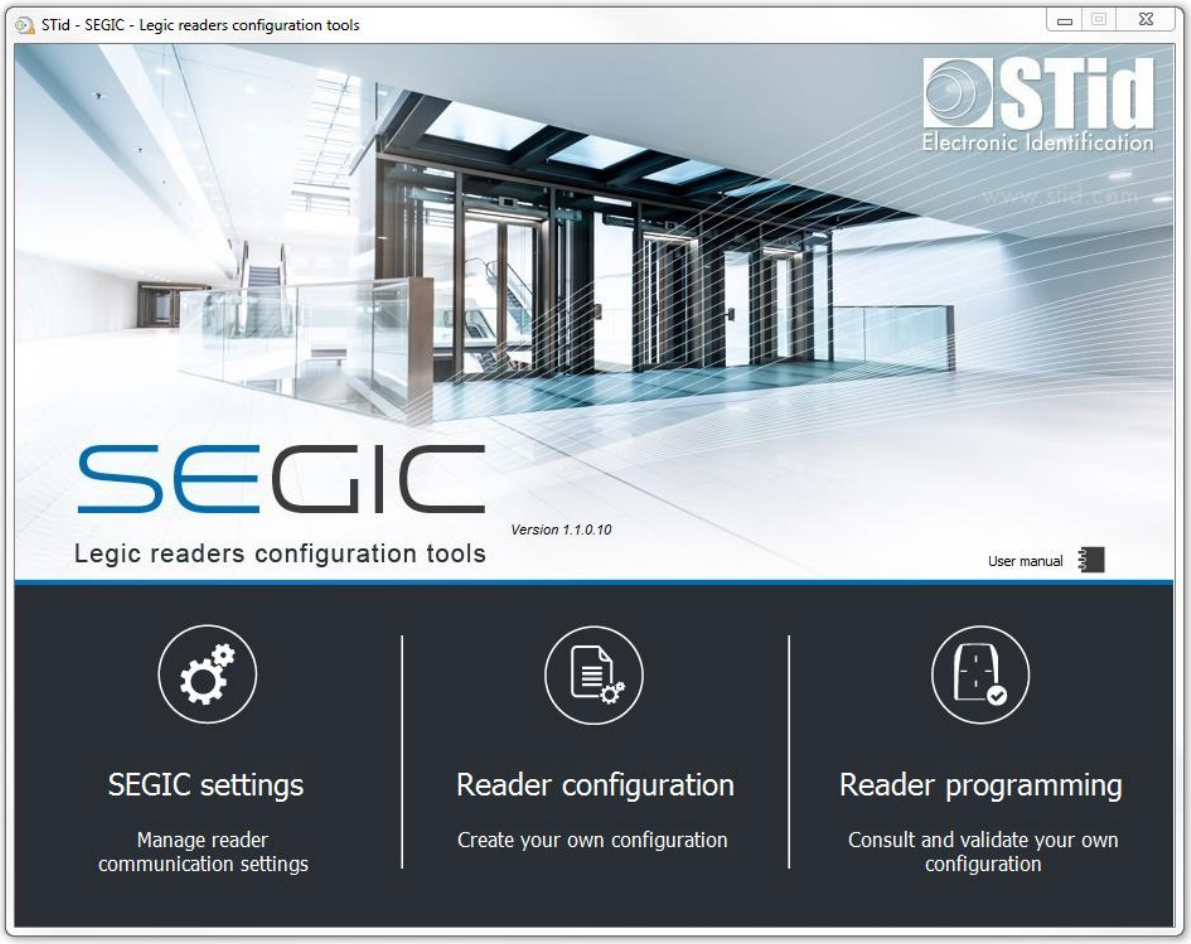
SEGIC software is available to configure Legic reader:

- ARC-R3x-L/Le2-xx: reader
- ARC-R3x-M/Le2-xx: reader + keypad
- ARC-R3x-N/Le2-xx: reader + Touchscreen
- ARC-R3x-O/Le2-xx: reader + biometrics
- ARC-R3x-P/Le2-xx: reader + keypad + biometrics
- ARC-R3x-Q/Le2-xx: reader + Touchscreen + biometrics
- ARC-R35-L/Le2-5AB: desktop reader

Warning: the version of the reader is not compatible with all versions of SEGiC refer to the compatibility table below:

	SEGIC 1.0	SEGIC1.1			
FW version 1	V	x			
FW version 2	V	x			
FW version 3	V	V			

6 - Open software



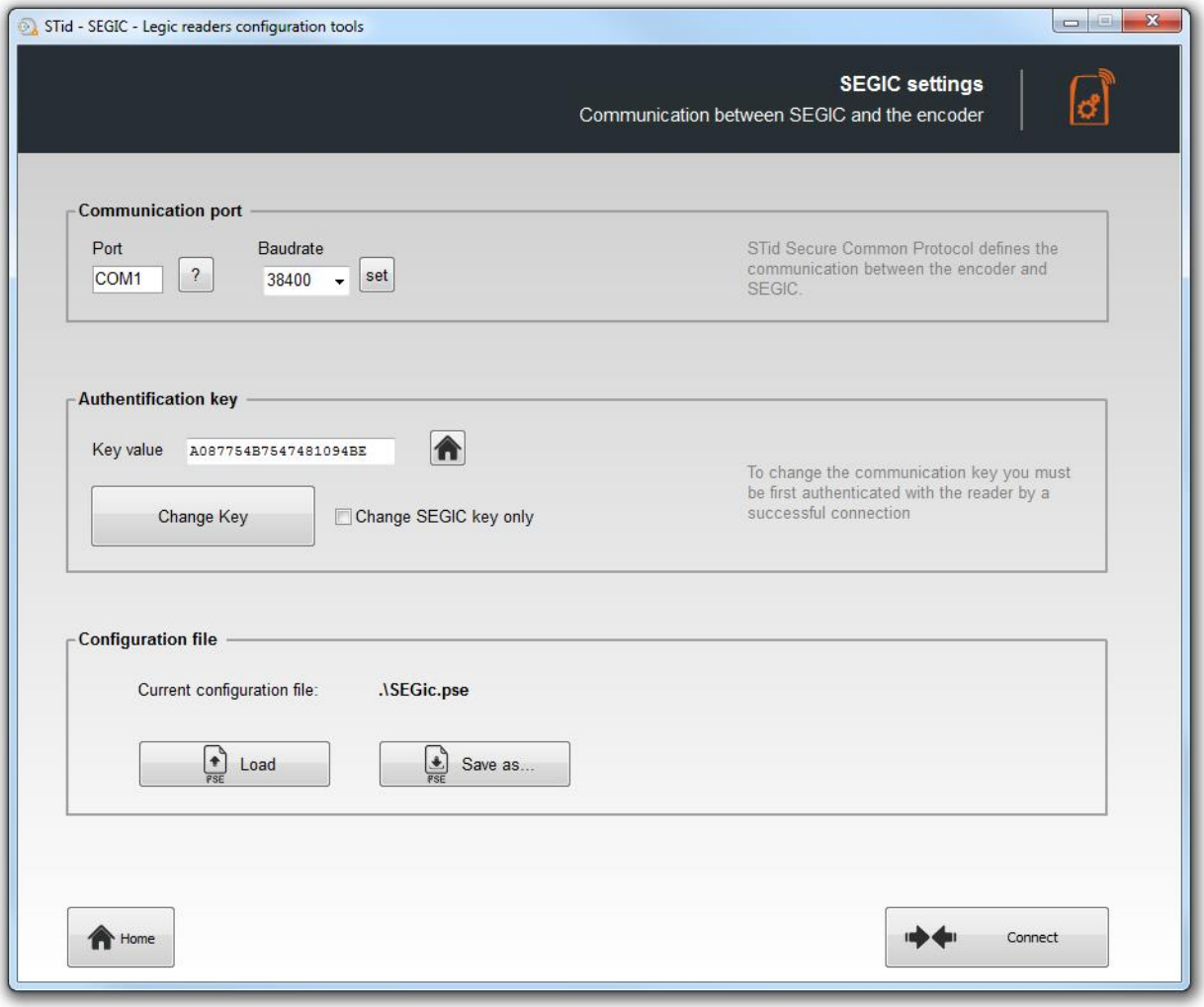
The software is divised into three parts:

SEGIC settings

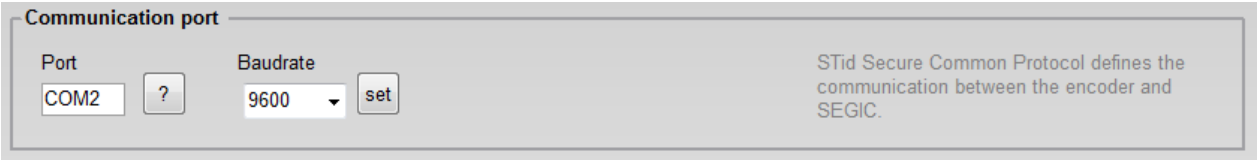
Reader configuration

Reader programming

SEGIC Settings



Communication port



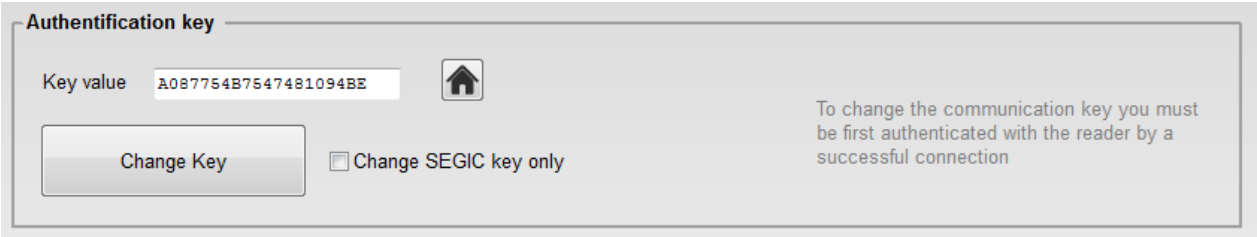
Set the communication between reader and SEGIC.

The default baudrate of the reader is 9600 bauds.
Caution, this baudrate must be exactly the same as that defined in the software.

Note:

- If you don't know the correct communication port connected to the reader, it is possible to find it by clicking on the button ?.
- By pressing the left CTRL key and by using the ? button SEGIC will search for a connected reader on all serial com ports and all speed rates. This can take some time.

Authentification key

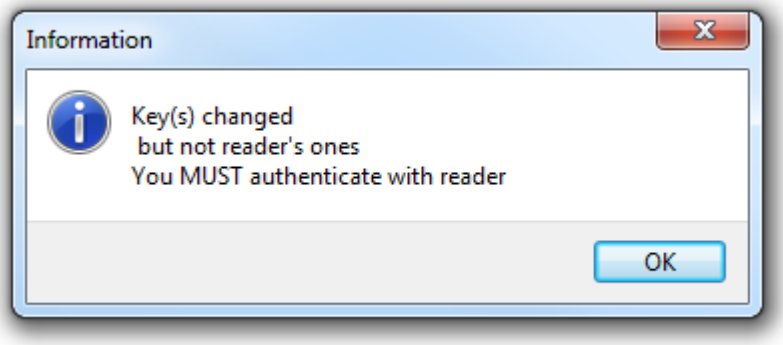


The communication between SEGIC software and reader is done by serial link or USB; it is based on the communication protocol SSCP (STid Secure Common Protocol).
Readers integrate public signature algorithms (HMAC-SHA1), which is used to secure data in serial communication between the reader and SEGIC.

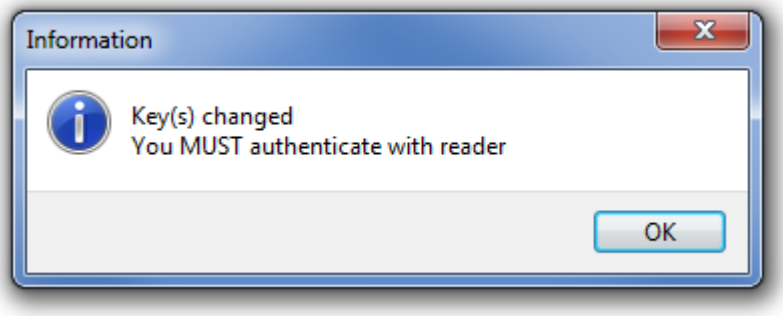
To change the communication key value, simply write the new value and then click-on the button “Change key”.

Note:

- ✓ The button  allows you to restore default value.
- ✓ Software **and** reader keys must be the same so that the two parts can communicate.
- ✓ If the box "Change SEGIC key only" is checked, only the key of the software will be changed.



- ✓ When changing user key and software encoder, a window will appear requesting authentication.

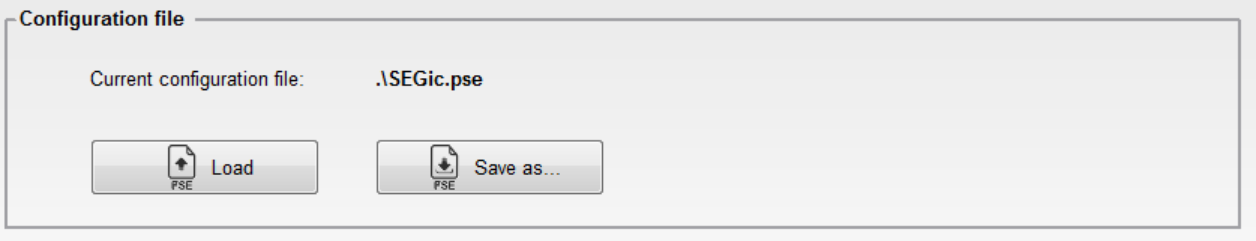


Warning

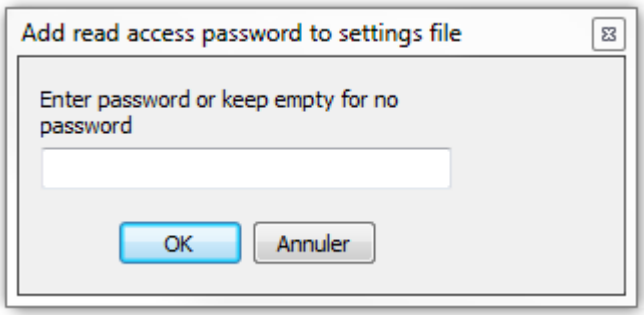
It is important to know the current user key.

If lost, it would not be possible to communicate securely with the reader.

Configuration file



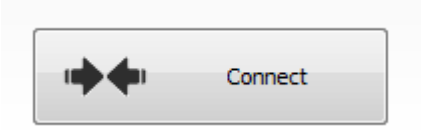
Allows to see the current configuration file.
You have the possibility to save the configuration file containing all the current configuration settings (formats, reader...) by informing a location, and read access password.



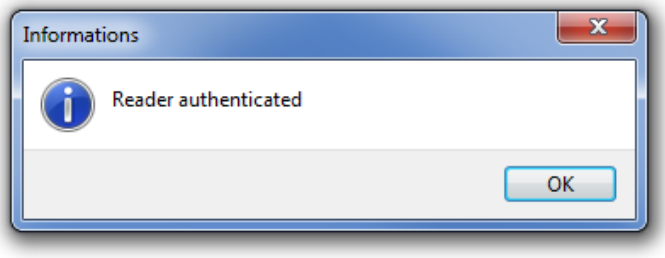
Connect

At power up the reader enters an initialization phase: Activating white LED, and activates buzzer for 100 ms.

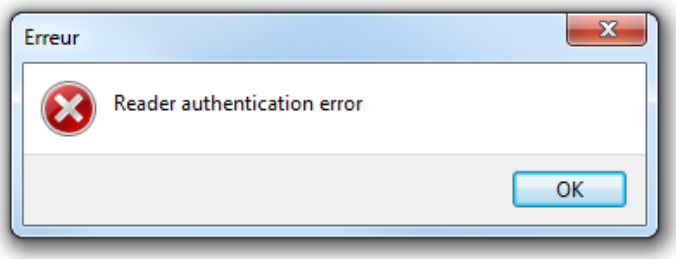
To verify the communication parameters between SEGIC and reader use the button:



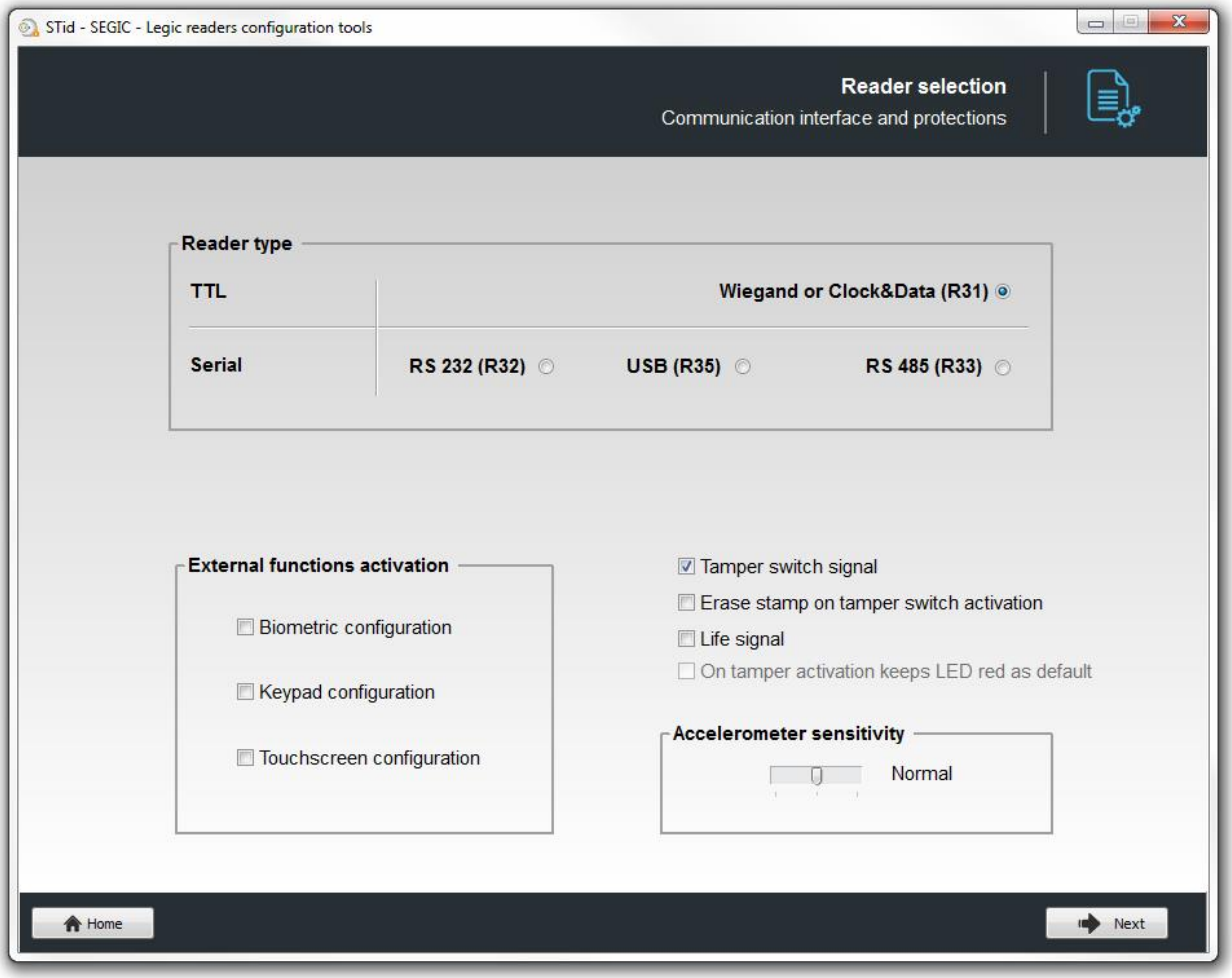
If the communication configuration is right, the reader will respond with light and sound signals and an acknowledgment window will appear.



Else you have an error message:



Reader selection



This step allows you:

- To choose the type of reader to configure.
- To activate external functions: Biometric configuration (available for Prime and Advent chips only) and/or Keypad configuration and/or Touchscreen configuration..
- To choose reader protection.

Management of biometric + Keypad

When biometric **and** keypad (Card THEN PINCode) are activate, the sequence is Card + BIO + PINCode.

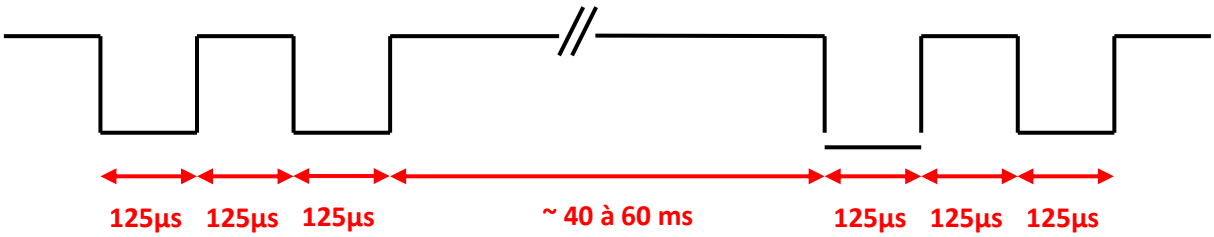
Accelerometer sensitivity

Readers of the ARC range include an accelerometer to detect the reader tearing.

Depending on the support / installation location of the reader, it may be necessary to adjust the sensitivity of the sensor so that only an effective tear is detected.

Reader protection option

- **Tamper switch signal**: When this feature is enabled, the reader remembers (at startup) its initial state of accelerometer. At each moment when that state changes the reader sends a Tamper signal.
  - The TTL reader will send the tamper signal on the Data/DATA1 lines. During the breakout, by default or if the option is enabled, the shape of the signal on the line "Data / Data 1" is as follows:



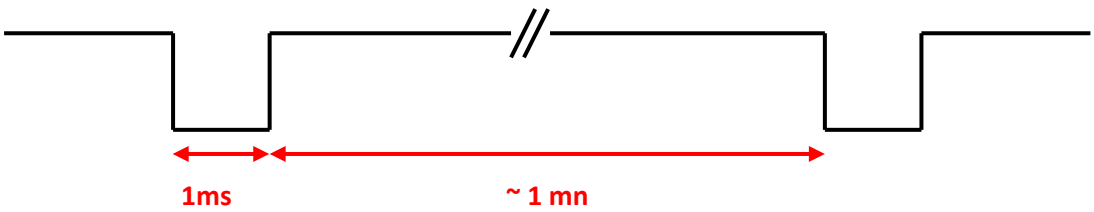
  - The serial reader will send a byte 0xAA.

- **Erase stamp on tamper switch activation**: if the status of accelerometer is changed, erase the stamps stored in the memory of Legic module.

- **Life signal**:

  If the option “Life signal” is activated the reader sends a signal about every minute.

  - The TTL reader will send the signal on the Data/DATA1 lines.

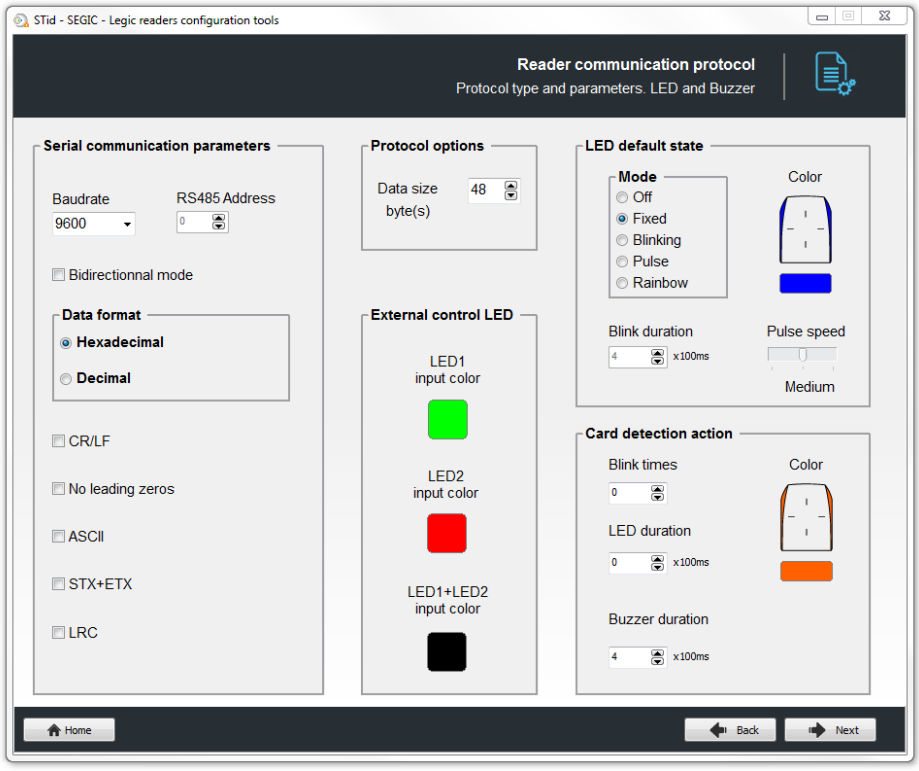


  - The serial reader will send a byte 0x61 every minute to indicate its presence

- **On tamper activation keeps LED red as default**: if the “Erase stamp on tamper switch activation” is enabled, and if the status of accelerometer is changed, the RED LED is activated, until an ON/OFF or an update (SAM or SEGiC).

Reader communication protocol

Depending on the choice of reader type selected at the previous step, you can have two different windows, one for TTL reader and one for Serial readers.



TTL reader communication protocol

This box contains the various communication protocols supported by the reader.

For more information about the protocols, refer to Annex *1 - About TTL communication protocols.*

Protocol options
Data size byte(s): adjust the size for custom protocol.

Maximum size in Wiegand: 48 bytes
Maximum size in Data/Clock: 10 bytes

Serial reader communication protocol

The reader has two communication modes (unidirectional and bidirectional) whose can be activated by ticking the case “Bidirectional mode”.

For more information about the serial communication refer to Annex *2 - About serial communication protocols*

Protocol options
Data size byte(s): adjust the size of data.

Maximum size in Hexadecimal: 48 bytes
Maximum size in Decimal: 10 bytes

LED default state

Define the LED default state (color and blink mode).

Several modes of “lighting” are available:

- Off
- Fixed
- Blinking
- Pulse
- Rainbow

The picture on the right allows you to view the selected effect: blinking and color.

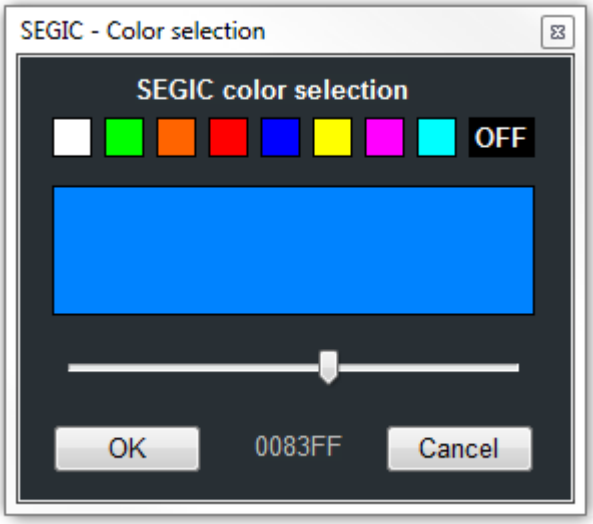
Card detection action

Define the LED and buzzer state (color and blink) when a card is detected.
This information is independent of the acceptance of the identifier.

External control LED color

Define the color of LED1, LED2 and both LED if they are controlled simultaneously.

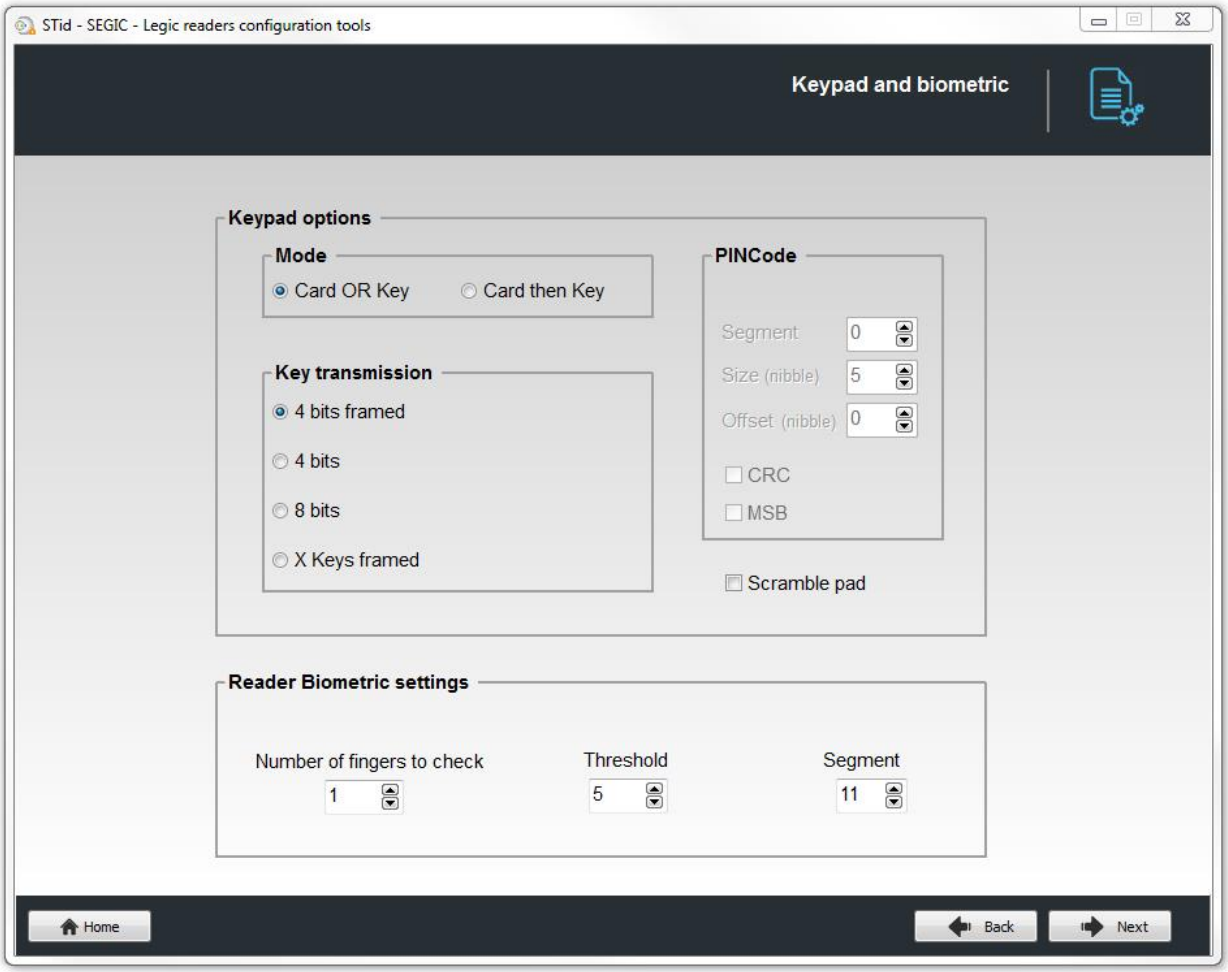
Note: To modify and select a color, click on the symbol ARC or color buttons, the following window opens:



To select a predefined color click on one of the colored squares color.

To choose a different color, move the cursor to the desired color.
The value displayed corresponds to the RGB hexadecimal code of the selected color. It is possible to copy the value by double clicking.

Keypad and biometric



Keypad options

Choose the mode "**Card OR Key**" or "**Card then Key**" between:

- **Card OR Key + Key transmission**:

  Once the reader detects a card, its ID number is sent according the selected protocol.
  Each time a key is pressed, its number is sent to the host according the selected protocol and key transmission followed by an acknowledgement sound expect in X keys format.

  For more detail on the functioning and format refer to Annex 3 - About keypad readers.

❖ **Card then Key + PINCode (only available for Legic Advant)**:

Once the reader detects a card, the reader expects a pin code entered by user for a period of 6 seconds (a beep sound to indicate pending identifier) and verify with the pin code in segment card before transmitting ID data.

PINCode parameters:

- Segment: Segment number in which is written the PIN code.
- Size: Number of PIN code data bytes
- Offset: PIN code Offset
- CRC: Activate / Desactivate CRC control on PIN code
- MSB: Activate / Desactivate MSB first on PIN code

Reader Biometric settings:

- Number of fingers to check: Number of finger to check without sending Private ID.
- Threshold: Quality level of the fingerprints.
  Morpho Sagem recommendation: 5
- Segment: Number of the segment contains the fingerprints

Templates should be in the following format:

The biometric templates must be written according to the Morpho Sagem format (PK_COMP).

MSB *LSB*

[**LenTotale**] | [**Nb Template**] | [**LenTemplate$_x$** | **Template$_x$**] ^{n}

- **LenTotale** is the total length biometric data on the chip, on 2 bytes.
- **Nb Template** is the number of template (max 5), on 1 byte.
- **LenTemplate$_x$** is the size of the Xth template on 1 byte.
- **Template$_x$** is the Xth template with **LenTemplate$_x$**.
- n is the number of templates.

The maximum size for a template is 170 bytes.

Management of biometric + Keypad

- **(Card AND biometric) OR Key:**:

  The operation is identical to Card OR Key, with the addition of the reading of the fingerprint after reading the card.

- **(Card AND biometric) then Key + PINCode:**

  The operation is identical to Card then Key, with the addition of the reading of the fingerprint after reading the card.

Scramble Pad:

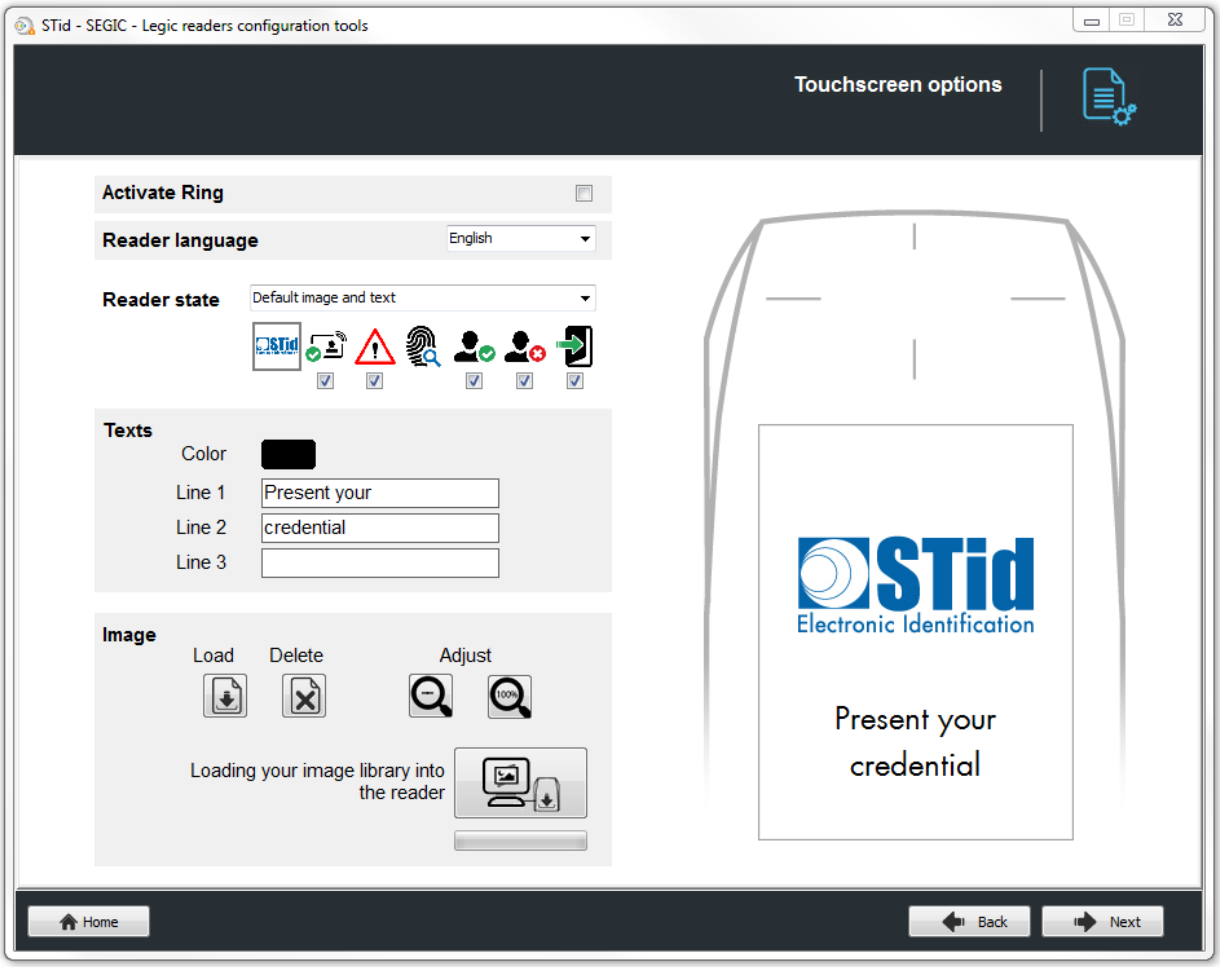
Only available for screentouch ARC. Allows to activate the scramble on keypad.

The scramble is performed:

- Card THEN Key:
  - After each sequence: enter the number of configured keys and reading a valid card.
  - After a time out of 6s after the seizure of configured keys without presentation of a valid card.
  - Following the annulment by the * or # key.

- Card OR Key:
  - After reading a valid card.
  - Every 30s. Pressing a key or reading a card resets the timer

Touchscreen options



Activate Ring

Activate / desactivate the ring.
When you press the ring, it will be activate during 1s.

	Headband appearance
Keypad inactive and Ring inactive	
Keypad active in Card then Key mode and Ring inactive	
Keypad inactive and Ring active	

Warning

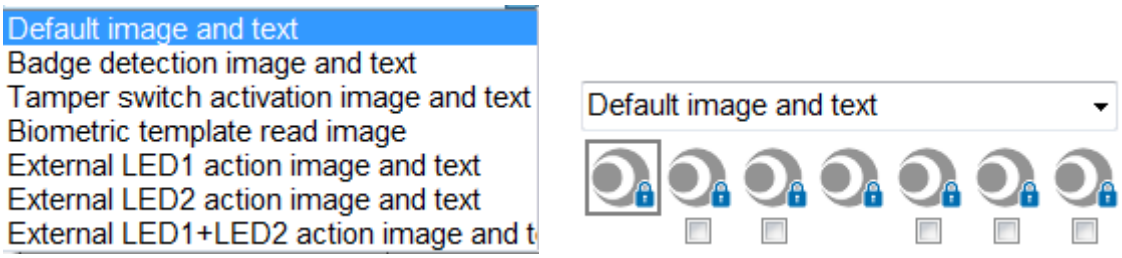
When the ring is active and if the reader has a screen then the tearing will not be effective on static relay (used for the ring).

Reader Language

Choose the language used to display the text on the screen: English (default) or French.

Reader State

Select the state to change, either from the drop down menu, or by clicking on the corresponding icon.



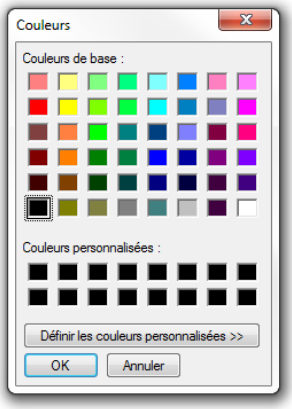
The check boxes allow you to select the states will be activated by the SCB and validate the image on the screen.

For each state, it's possible to change the image, the text and the text color.

Note: for the biometric, the text is not modifiable because it takes into account the number of fingers defined in the configuration wizard.

Texts

To change the text color click on the color button.

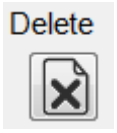
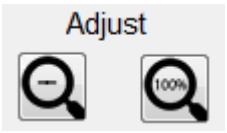


(the language of this windows depend of your Windows language)

The color applies to the three lines of text.

Image

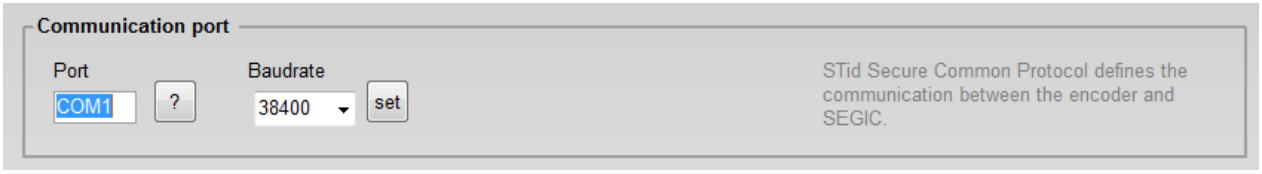
Load an image file in the reader memory

	Load an image file for the selected state.
	Delete the image file to the selected state.
	Reduces the image on the screen.

Loading image into the reader

After loading the images into SEGiC for the seven states, they must be loaded into the reader.

1 - Connect the screen touch reader to your computer with the reader serial link and set the communication in “SEGiC Settings”



2 - Power on the reader and click on  while the LED blinks orange for serial readers or at any time for TLL readers.

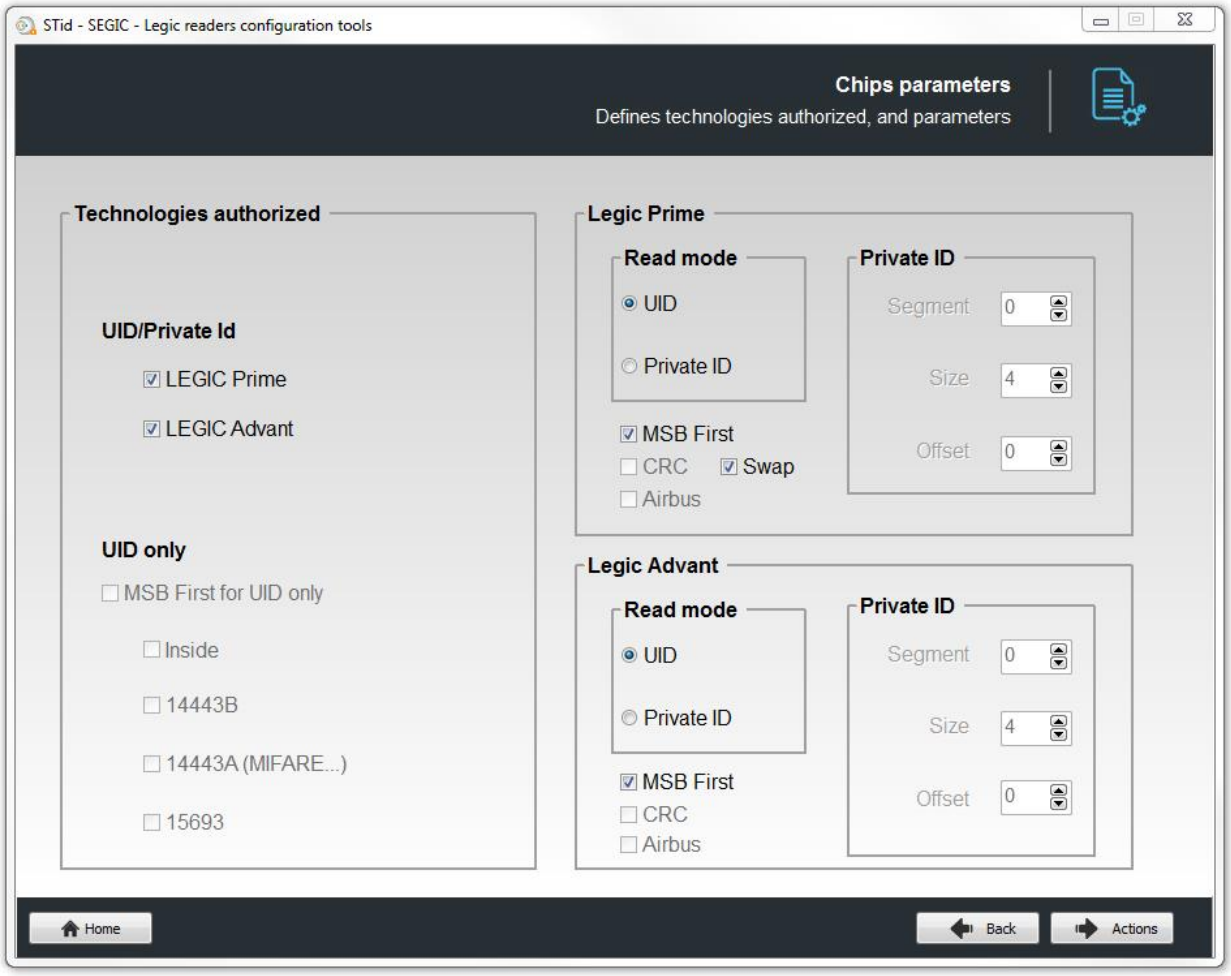
3 - The charging progress is indicated by the progress bar:
The operation is repeated seven times, once for each image.

Note:

- Each image has an index, a new load erase the image loaded before.
- If the image has been loaded into SEGiC was moved, the preview will not be available and the next image will be displayed in the IHM SECard.

- The display time of the state “Badge detection image and text” is defined in step 5 “LED and Buzzer” with “Blinks times” if the blink is activated or “LED duration”.
-

Chips parameters



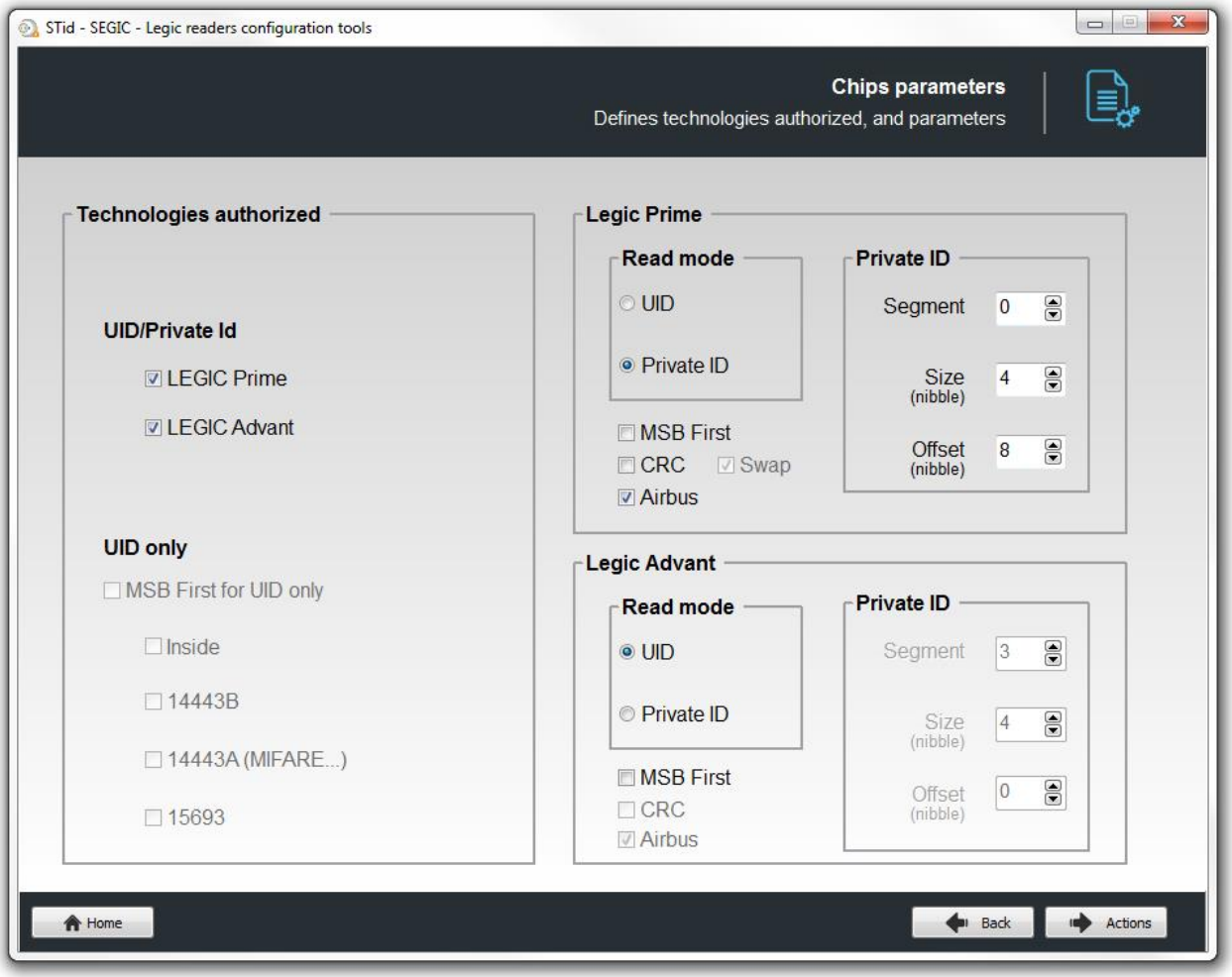
UID only

MSB First for UID only: If the box is checked the reader reads the UID Most Significant Byte First.
If the box is unchecked the reader reads the UID Least Significant Byte First.

- Inside: Reader configured in “read-only serial number” for Inside chip.
- 14443B: Reader configured in “read-only serial number” for 14443B chip.
- 14443A (MIFARE): Reader configured in “read-only serial number” for 14443A chip.
- 15693: Reader configured in “read-only serial number” for 15693 chip.

Note: It’s possible to authorize the four technologies in the same time.

Legic Prime /



Read mode UID: Reader configured in “read-only serial number”.

MSB First If the box is checked the reader reads the UID Most Significant Byte First.
If the box is unchecked the reader reads the UID Least Significant Byte First.

Swap Swap between byte 1 and 3.
Ex: 3D 02 5F 20 ➔ 3D 20 5F 02

Read mode Private ID: Reader configured in “read-only private code”.

MSB First If the box is checked the reader reads the Private ID Most Significant Byte First.
If the box is unchecked the reader reads the Private ID Least Significant Byte First.

CRC If the box is checked a CRC 16-bytes is used for a data consistency check.
Note: CRC must be written immediately after the Id in the segment.

Airbus If the box is checked the reader reads the Private ID with Airbus format.

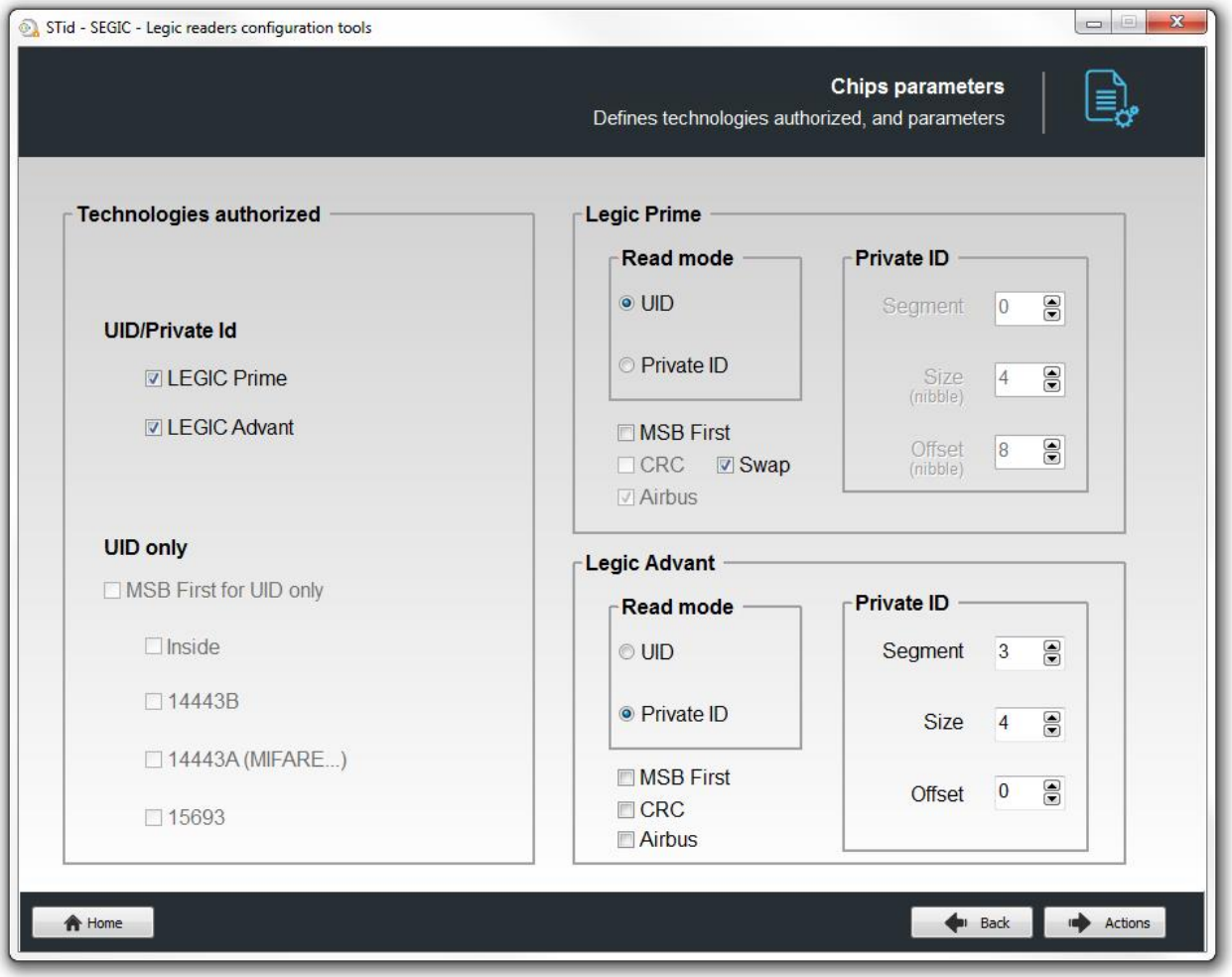
Private ID:

Segment: Segment number to be read by the reader.

Size: Determines the length of the ID read in the segment. The value corresponds to the protocol selected in the configuration of the reader. However it is possible to choose a different size by entering another value, in this case the reader will read the ID to the size specified in this field and will return to the format defined by the protocol.

Offset: Define an offset from the first byte in the segment.

Legic Advant



Read mode UID: Reader configured in “read-only serial number”.

MSB First If the box is checked the reader reads the UID Most Significant Byte First.
If the box is unchecked the reader reads the UID Least Significant Byte First.

Read mode Private ID: Reader configured in “read-only private code”.

MSB First If the box is checked the reader reads the Private ID Most Significant Byte First.
If the box is unchecked the reader reads the Private ID Least Significant Byte First.

Airbus If the box is checked the reader reads the Private ID with Airbus format.

CRC If the box is checked a CRC 16-bytes is used for a data consistency check.
Note: CRC must be written immediately after the Id in the segment.

Private ID

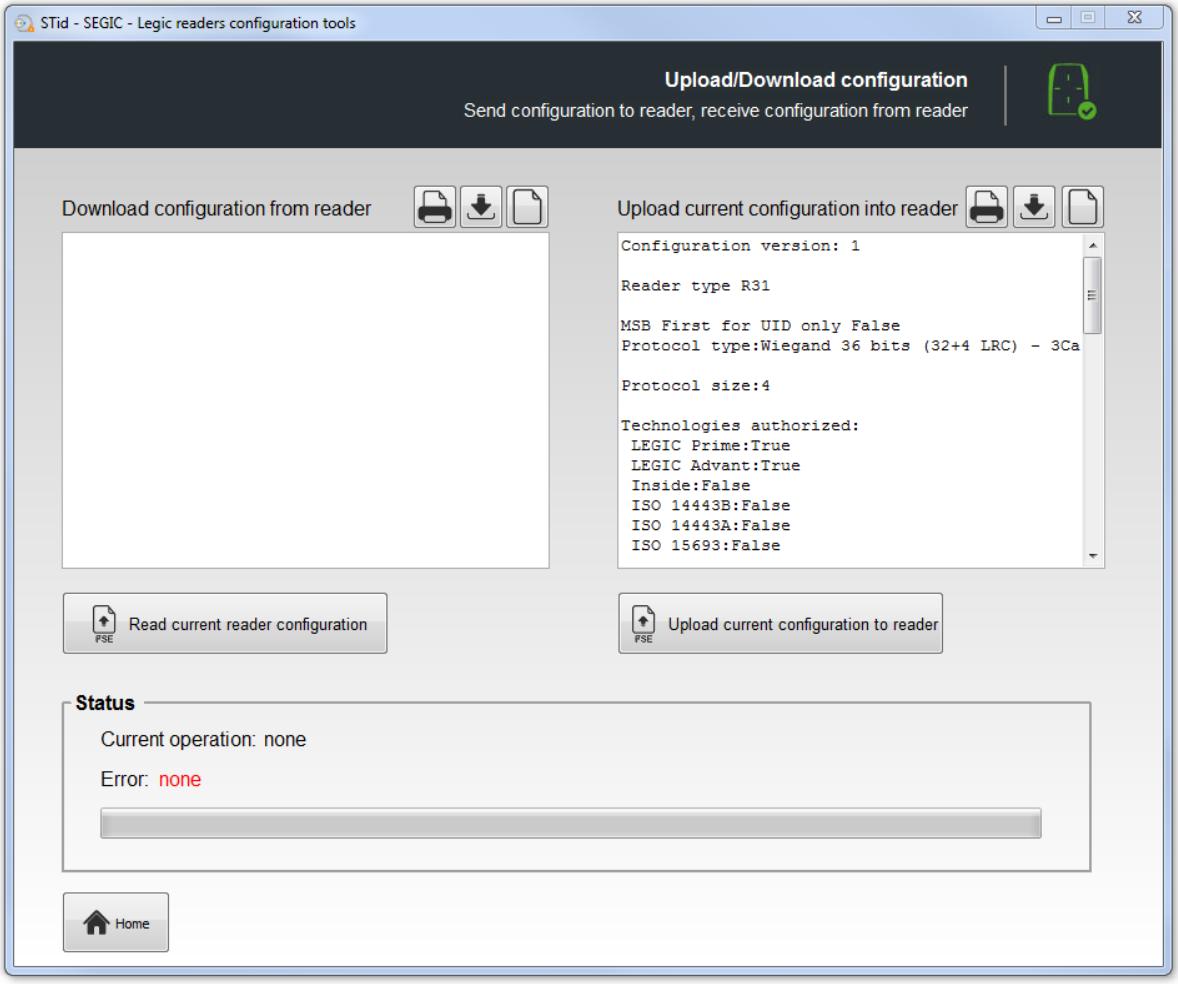
Segment: Segment number to be read by the reader.

Size: Determines the length of the ID read in the segment. The value corresponds to the protocol selected in the configuration of the reader. However it is possible to choose a different size by entering another value, in this case the reader will read the ID to the size specified in this field and will return to the format defined by the protocol.

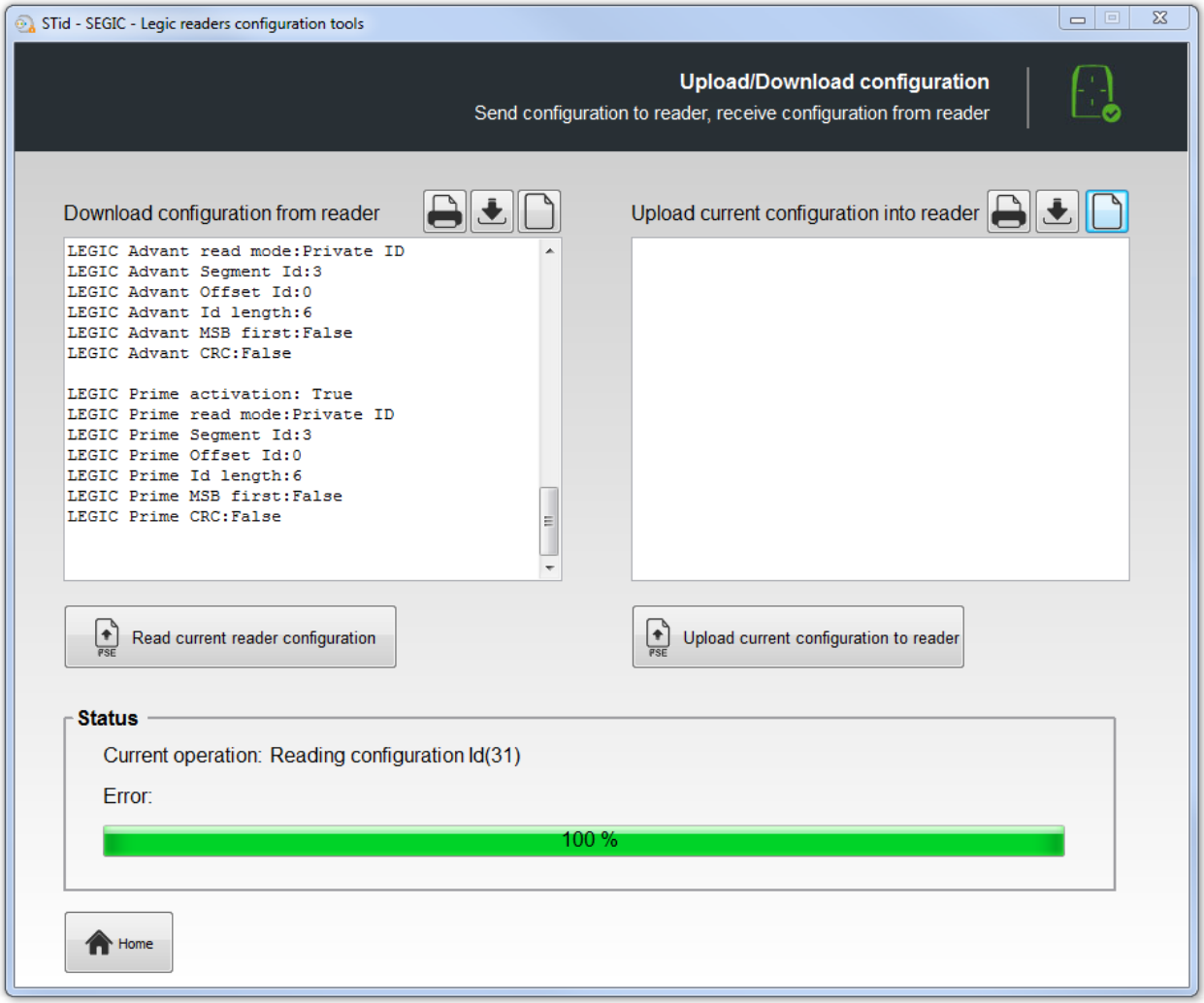
Offset: Define an offset from the first byte in the segment.

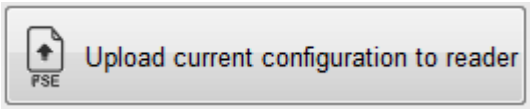
Note: It’s not authorized to read Private ID Prime and Private ID Advant
The possibilities available are UID Prime and Private ID Advant or Private ID Prime and UID Advant.

Reader programming

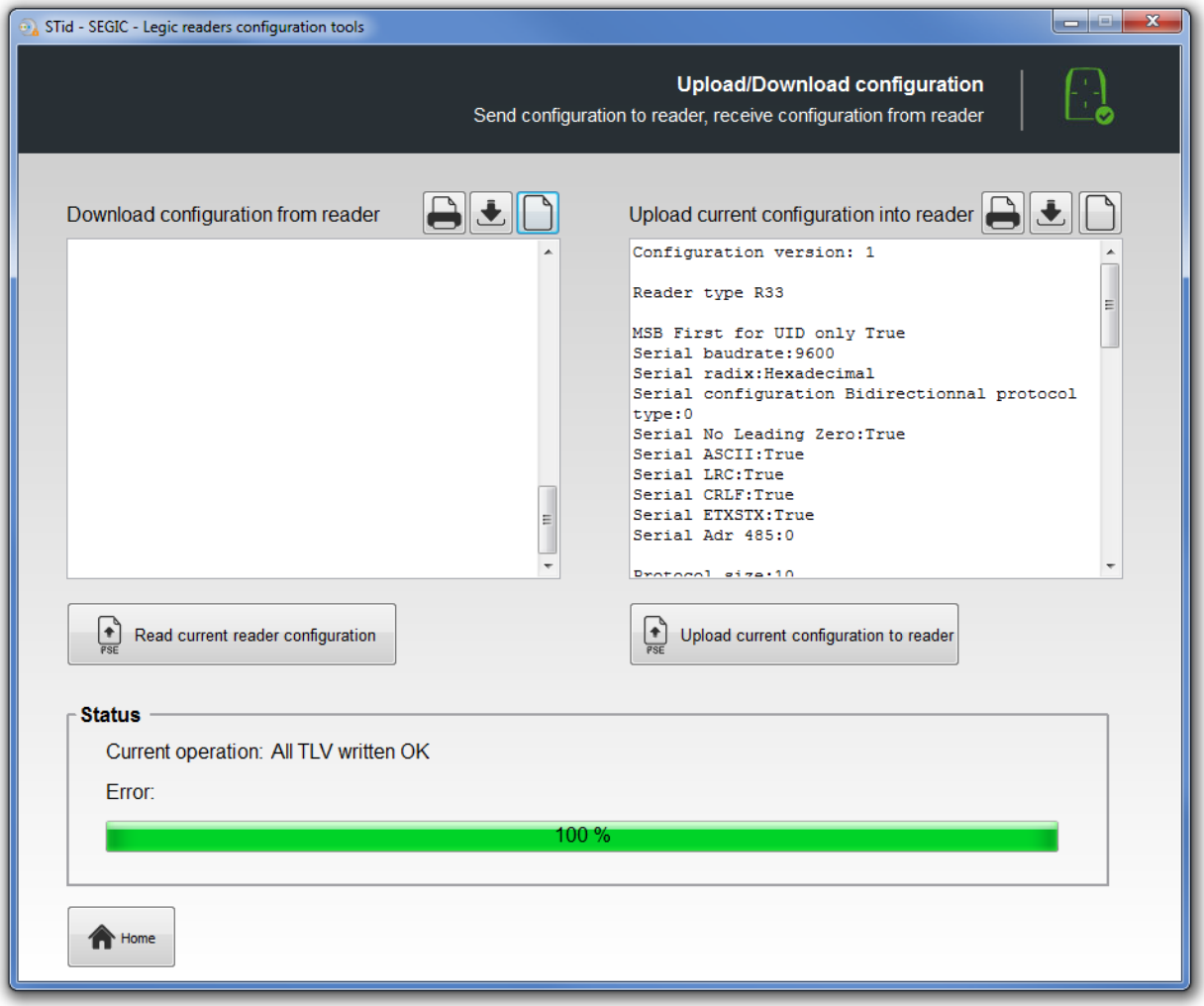


Connect the reader and click on this button to read the current reader configuration.





Connect the reader and click on this button to upload the current configuration to reader.



	Print the configuration list displayed.
	Save in .rtf file the configuration list displayed.
	Cleat the configuration list displayed.

STid
Electronic Identification

SEGIC

ANNEXES

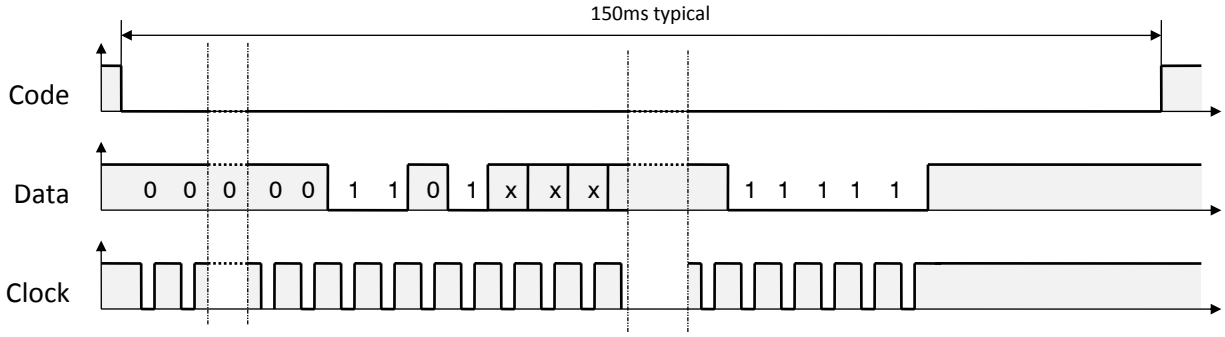
Designed in France
Made in France

www.stid.com

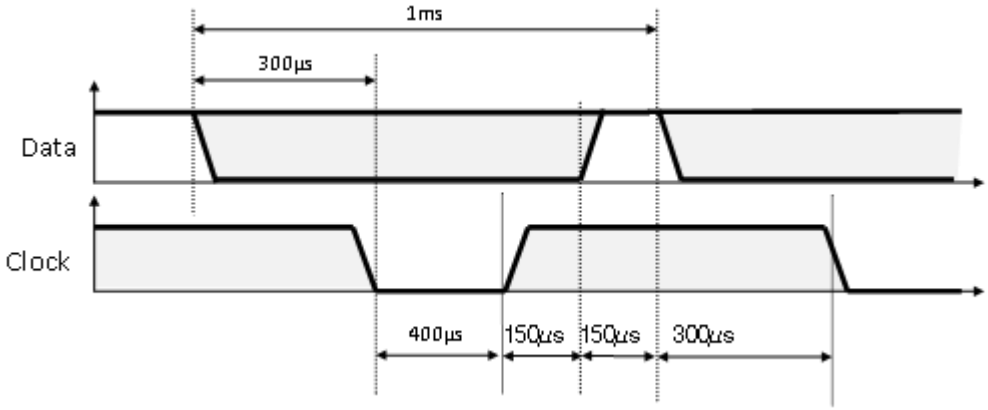
1 - About TTL communication protocols

1.1 - ISO2 Clock&Data protocol

Chronograms



Clock details



Message structure (*2B & 2H)*

Leading zeroes	*Start Sentinel*	*Datas*	*End Sentinel*	*LRC*	*Trailing zeroes*

Message description

The frame is made of a first series of 16 zero followed by synchronization characters of 5 bits (4 bits, LSB first, plus 1 parity bit). It ends the frame with trailing zero without a clock. The message consists of the following:

Start Sentinel:	1 character 1011b (0x0B) – parity bit 0. Transmission 1101 0
Data:	According to ID type: 13 or 10 decimal characters
End Sentinel:	1 character 1111b (0x0F) - parity bit 1. Transmission 1111 1
LRC:	1 control character, which is the « XOR » of all characters.

2B protocol (13 characters)

Reading an ID of 5 bytes (40 bits) and convert to decimal.

Variant	Decoding	Full frame of 112 bits	Values
2B	*Decimal (BCD)*	*13 characters*	*0 to 9*

Example:

For a hexadecimal user code of « 0x187E775A7F », the output code will be: « *0105200966271* ».
Frame sent by reader will be:

000...	1101 0	0000 1	1000 0	0000 1	1010 1	. . . .	0110 1	0100 0	1110 0	1000 0	1111 1	1111 1	000...
	B	*0*	*1*	*0*	*5*	*2 0 09 6*	*6*	*2*	*7*	*1*	*F*	*F*	
Zero	*S.S*	*Char.1*	*Char.2*	*Char.3*	*Char.4*	*Char ….*	*Char.10*	*Char.11*	*Char.12*	*Char.13*	*E.S*	*LRC*	*Zero*

2H protocol (10 characters)

Reading an ID of 4 bytes (32 bits) and convert to decimal.

Variant	Decoding	Full frame of 97 bits	Values
2H	*Decimal (BCD)*	*10 characters*	*0 to 9*

Example:

For a hexadecimal user code of « 0x06432F1F», the output code will be: « *0105066271* ».

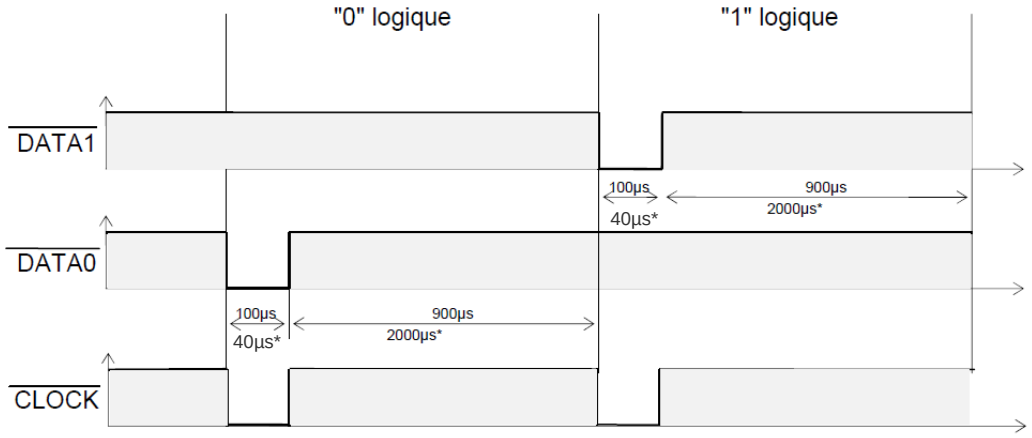
Frame sent by reader will be:

000...	1101 0	0000 1	1000 0	0000 1	1010 1	. . . .	0110 1	0100 0	1110 0	1000 0	1111 1	0010 1	000...
	B	*0*	*1*	*0*	*5*	*0 6*	*6*	*2*	*7*	*1*	*F*	*4*	
Zero	*S.S*	*Char .1*	*Char .2*	*Char .3*	*Char .4*	*Char ….*	*Char .7*	*Char .8*	*Char .9*	*Char .10*	*E.S*	*LRC*	*Zero*

In the case of 5 bytes (40 bits) ID, reader will truncate the MSB byte (8 bits) before decimal conversion.

1.2 - Wiegand Protocol

Chronograms



* variant 3i timings

Wiegand 3i protocol

Variant	Decoding	24 bits data	Values
3i	*Hexadecimal*	*6 characters*	*0 to F*

Message structure

Bit 1	Bit 2 ... Bit 25	Bit 26
Even parity from bit 2 to bit 13	*Data (24 bits)*	*Odd parity from bit 4 to bit 25*

Message description

The frame consists of 26 bits as follows:

First parity: *1 bit even parity of next 12 bit*
Data: *6 hexadecimal characters " MSB first "*
Last parity: *1 bit odd parity of previous 12 bits*

Example: for the hexadecimal code « *0x0FC350* », frame sent will be:

0	0000	1111	1100	0011	0101	0000	1
	0	*F*	*C*	*3*	*5*	*0*	
Parity	*Char.1*	*Char.2*	*Char.3*	*Char.4*	*Char.5*	*Char.6*	*Parity*

Note:
A site code is generally associated with the third octet (byte [2]). In the example above, it is 0x0F or 15 in decimal (up to 255 decimal - 0xFF in hexadecimal).

The card code is generally associated with the first and second byte (byte [1] and byte [0]). In the example above, it is 0xC350, 50000 in decimal (decimal max is 65535 - 0xFFFF in hexadecimal).

Wiegand 3CB protocol

Bit 1 ... Bit 40	Bit 41... Bit 44
Data « MSB first »	*LRC*

Message description

The frame consists of 44 bits as follows:

Data: 10 *hexadecimal characters « MSB first »*
LRC : *1 control char , all characters « XORed»*

Example: for the hexadecimal code « *0x01001950C3* », frame sent will be:

0000	0001	0000	0000	0001	1001	0101	0000	1100	0011	0011
0	*1*	*0*	*0*	*1*	*9*	*5*	*0*	*C*	*3*	*3*
Char.1	*Char.2*	*Char.3*	*Char.4*	*Char.5*	*Char.6*	*Char.7*	*Char.8*	*Char.9*	*Char.10*	*LRC*

Wiegand 3CA protocol

Bit 1 ... Bit 36	Bit 37... Bit 40
Data « MSB first »	*LRC*

Message description

The frame consists of 40 bits as follows:

Data: *8 hexadecimal characters « MSB first » (32 bits)*
LRC: *1 control char , all characters « XORed »*

Example: for the hexadecimal code « *0x001950C3* », the frame sent will be:

0000	0000	0001	1001	0101	0000	1100	0011	0010
0	*0*	*1*	*9*	*5*	*0*	*C*	*3*	*2*
Char.1	*Char.2*	*Char.3*	*Char.4*	*Char.5*	*Char.6*	*Char.7*	*Char.8*	*LRC*

Note: in the case of 5 bytes (40 bits) ID, reader will truncate the MSB byte (8 bits) before decimal conversion.

Wiegand 3LA protocol

Same as « *Wiegand 3CA* » WITHOUT LRC.

Wiegand 3LB protocol

Same as « *Wiegand 3CB* » WITHOUT LRC.

Wiegand 3T protocol

Bit 1 ... Bit 8	Bit 9 ... Bit 64	Bit 65... Bit 68
Chip type	*Data « MSB first »*	*LRC*

The frame consists of 68 bits as follows:

RFId Chip Type: *1byte (8 bits)*
Data: *14 hexadecimal characters « MSByte first » (56 bits)*
LRC: *1 control character, , all characters(4 bits) « XORed»*

« *Chip type*» indicates the type of chip read by the reader:

- 0x40 ➔ MIFARE® Classic
- 0x41 ➔ MIFARE® DESFire® / DESFire® Ev1
- *0x42* ➔ *125 kHz (EM/Nedap/HID) (only range E)*
- 0x43 ➔ MIFARE Ultralight® / Ultralight® C
- 0x44 ➔ MIFARE Plus® Level 0 / Level 2 / Level 3
- 0x45 ➔ PUPI ISO 14443-3B + iClass + inside
- *0x46* ➔ *CPS3 (range E or ARC Mifare)*
- *0x47* ➔ *Moneo (range E or ARC Mifare)*
- *0x4A* ➔ *3.25 MHz chips (only range E)*
- 0x4B ➔ Legic Prime (only Legic reader)
- 0x4C ➔ Legic Advant (only Legic reader)
- 0xE0 ➔ ISO 15693 (only Legic reader)
- 0x50 ➔ Undefined chip

Example for MIFARE® DESFire® chip:

For the hexadecimal code « 0x80AF01001950C3 », frame sent will be 0x41 80 AF 01 00 19 50 C3.

Example for MIFARE® Classic chip:

For the hexadecimal code « 0xA771FE4C », frame sent will be 0x40 00 00 00 A7 71 FE 4C.

2 - About serial communication protocols

2.1 - Unidirectional communication mode

The communication is made from the reader to the system.

LED and buzzer are managed by the reader through the configuration.

It is possible to configure the structure of the message sent by the reader and with the following:

- ✓ No leading zeros: Add on the frame leading zero (on start of frame).
- ✓ STX+ETX: Add STX (0x02) and ETX (0x03) on start and end of the frame.
- ✓ CR/LF: Carriage return option (0x0D + 0x0A)
- ✓ LRC: Checksum byte by XORing of all previously characters without the STX.
- ✓ ASCII: If this option is activated, the Data will be sent in ASCII mode.
- ✓ Base: Data sent in decimal or hexadecimal format.
- ✓ Baudrate: 9600, 19200, 38400, 57600 or 115200 bauds.

"Data" part is the identifier code read or keys reader Card or Key mode

1 byte	X bytes	1 byte	1 byte	1 byte	1 byte
STX	Data	LRC	0x0D	0x0A	ETX

Note:

- ✓ R33 readers are not addressable in this mode.
- ✓ The data size is doubled if the *ASCII* is activated.
- ✓ The field "*Size"* allows the modification of the data size sent by the reader.

2.2 - Bidirectional communication mode

In this mode, the communication is done from the reader to the system for the transmission of the data and from the system to the reader for the LEDs and buzzer management.

In an idle state, the reader is going to manage the LEDs and buzzer according the configuration defined in the tab "*Default LED action*".

When reading a valid code (depending on the configuration defined), it is transmitted to system by the reader. It is then possible at this time and for a period of 1.5s to command the buzzer and LED via the transmission of a frame of the system.

Complete frame sent by the reader

#02	Len	CTRL	CMD	Reserved	L_{out}	$Data_{out}$	CRC
1 byte	2 bytes	2 bytes	4 bytes	2 bytes	2 bytes	L_{out} bytes	2 bytes

Complete frame sent by the system

#02	Len	CTRL	ACK	L_{in}	$Data_{in}$	Status	CRC
1 byte	2 bytes	2 bytes	2 bytes	2 bytes	L_{in} bytes	2 bytes	2 bytes

The information transmitted by the reader is formatted as follows:

#02	*Len*	*CTRL*	*CMD*	*Reserved*	L_{out}	$Data_{out}$	*CRC*
1 byte	2 bytes	2 bytes	4 bytes	2 bytes	2 bytes	L_{out} bytes	2 bytes

- ✓ #02: Start Of Frame (SOF) delimiter (on byte 02h).
- ✓ Len: Defines the length of the command to be sent (two bytes)
- ✓ CTRL: Two-byte word, with one byte that defines the communication mode and one byte that defines the serial link type used (RS485 or RS232).

  - CTRL Mode 00h ➔ Non-secure mode - message sent in plain text.

  - CTRL @ b0 ➔ ***« 0 »*** RS232 | ***« 1 »*** RS485
    b7 – b1 ➔ ***« 1111 111 »*** à ***« 0000 000 »***

b7 – b1	*b0*
Reader adress (RS485)	*Serial link used*

- ✓ CMD : Four-byte word: two bytes (Type) for commande type and two bytes (Code) for command code to transmitte.
  - Type 00h 00h ➔ Reader command
- ✓ Reserved: AAh 55h (2 bytes)
- ✓ Lout: Determines the size of the data sent by the reader (2 bytes)
- ✓ $Data_{out}$: Represents the data sent by the host (in the case of writing, for example)
- ✓ CRC: CRC-16-CCITT [Len….Command] ***[Polynomial "$x^{16} + x^{12} + x^5 + 1$"*** 0x1021] ; *Initial value 0xFFFF*

The information transmitted to the reader is formatted as follows:

#02	*Len*	*CTRL*	*ACK*	L_{in}	$Data_{in}$	*Status*	*CRC*
1 byte	2 bytes	2 bytes	2 bytes	2 bytes	L_{in} bytes	2 bytes	2 bytes

- ✓ #02: Start Of Frame (SOF) delimiter (on byte 02h).
- ✓ Len: Defines the length of the command to be sent (two bytes)
- ✓ CTRL: Two-byte word, with one byte that defines the communication mode and one byte that defines the serial link type used (RS485 or RS232).

  - CTRL Mode 00h ➔ Non-secure mode - message sent in plain text.

  - CTRL @ b0 ➔ ***« 0 »*** RS232 | ***« 1 »*** RS485
    b7 – b1 ➔ ***« 1111 111 »*** à ***« 0000 000 »***

b7 – b1	*b0*
Reader adress (RS485)	*Serial link used*

- ✓ ACK: Start of Frame acknowledgement, identical to the command code sent by host (2 bytes).
- ✓ L_{in}: Defines the length of data to be received by the host (2 bytes).
- ✓ $Data_{in}$: Data sent by the reader in response to the host command (L_{in} bytes).
- ✓ Status: Two-byte word, representing the status type and the command result code:
  - RFU ➔ 00h
  - Type ➔ 00h reader command (1 byte)
  - Code ➔ Defines the command code to be sent to the reader (one byte)
- ✓ CRC: CRC-16-CCITT [Len….Command] ***[Polynomial "$x^{16} + x^{12} + x^5 + 1$"*** 0x1021] ; *Initial value 0xFFFF*

Available commands in plain mode

Output_Protocol

Description

This command is sent by the reader when it reads a valid tag and / or pin number. It's transmitted in hexadecimal

Reader: CTRL CMD AAh 55h L_{out} $Data_{out}$

CMD *2 bytes*: 01h 00h

L_{out} *2 bytes:* $Data_{Len}$ Equal to the number of bytes of Data

$Data_{out}$ *x bytes*: Id value read in hexadecimal.

System: CMD L_{in} LedColor LedDuration BuzzerDuration 00h 00h

CMD *2 bytes*: 01h 00h

L_{in}*2 bytes*: 00h 03h (LedColor + LedDuration + BuzzerDuration)

LedColor *1 byte:* Byte indicating the LED color.
[00h … 03h]

- 00h Led off
- 01h Green Led
- 02h Red Led
- 03h Orange Led

LedDuration *1 byte*: This byte defines the LED colour-change duration in multiples of 100 ms [00h … FFh] where the value FFh keeps the LED on with the same colour for an indefinite period (until the next reader reset or the next time a value other than FFh is sent).

BuzzerDuration *1 byte*: This byte defines the buzzer activation duration in multiples of 100 ms [00h … FFh] where the value FFh keeps the buzzer on for an indefinite period (until the next reader reset or the next time a value other than FFh is sent).

Note

The reader has a 1.5s timeout to receive the response of the system for the control of LEDs and buzzer. Once this deadline has passed, it will not accept any frame until the next issue of the Output_Protocol order.

Life_Signal

Description

This command is sent by the reader each minute to keep the system informed about its presence.

Reader: CTRL CMD AAh 55h L_{out} $Data_{out}$

CMD *2 bytes*: 01h 02h

L_{out} *2 bytes:* 00h 02h Equal to the number of bytes of Data

$Data_{out}$ *2 bytes:* 00h + 07h

System: CMD L_{in} 00h 00h

CMD *2 bytes:* 01h 02h

L_{in} *2 bytes:* 00h 00h

Note

It is necessary to activate this option through SEGIC software.

Wrenching_Signal

Description

This command is sent by the reader when it detects a state changing on accelerometer. That informs the system about an potential wrenching of the reader.

Reader: CTRL CMD AAh 55h L_{out} 00h

CMD *2 bytes:* 01h 03h

L_{out} *2 bytes:* 00h 01h Equal to the number of bytes of Data

System: CMD L_{in} 00h 00h 00h 00h

CMD *2 bytes:* 01h 03h

L_{in} *2 bytes:* 00h 00h

Read_input

Description

This command is sent periodically by the reader to the system. It allows the system to control the activation of the LEDs and buzzer.

Reader: CTRL CMD AAh 55h 00h

CMD *2 bytes:* 01h 04h

System: CMD L_{in} LedGreen LedRed Buzzer 00h 00h

CMD *2 bytes:* 01h 04h

L_{in} *2 bytes:* 00h 03h

LedGreen *1 byte*: 01h inactive
00h active

LedRed *1 byte* : 01h inactive
00h active

Buzzer *1 byte:* 01h inactive
00h active

3 - About keypad readers

3.1 - TTL Readers - R31 - Card OR Keys

The reader works in mode a Card OR Key. If a valid card is presented or if a key is pushed (according the encoding mode), the code will be sent immediately, followed by a short beep of the reader.

About the encoding mode type 4, a keys sequence written is confirmed by pushing the key **'★'**. In this case, the code is transmitted according the encoding mode. There is a Timeout between two keys pushing for 6 seconds. If it happens, the sequence is cancelled.

Formats available

- **'1': « 4 bits framed »**
  Value is coded by 4 bits which are sent within a frame according the chosen protocol.

Format ISO2 LSB … MSB		
'0'	0000	0x00
'1'	1000	0x01
'2'	0100	0x02
'3'	1100	0x03
'4'	0010	0x04
'5'	1010	0x05
'6'	0110	0x06
'7'	1110	0x07
'8'	0001	0x08
'9'	1001	0x09
'#'	1101	0x0B

In this case, 4 bits are sent LSB First within a frame according the chosen protocol. For more details, refer to the specification protocols.

Example: Frame of the key '5' according the protocol ISO2 / 2b.

Format WIEGAND MSB … LSB		
'0'	0000	0x00
'1'	0001	0x01
'2'	0010	0x02
'3'	0011	0x03
'4'	0100	0x04
'5'	0101	0x05
'6'	0110	0x06
'7'	0111	0x07
'8'	1000	0x08
'9'	1001	0x09
'#'	1011	0x0B

In this case, 4 bits are sent MSB First within a frame according the chosen protocol. For more details, refer to the specification protocols**.**

Example: Frame of the key '5' according the protocol Wiegand / 3i.

000…	1101 0	1010 1	1111 1	xxxx x	000…
Zeros	**Start**	**'5'**	**End**	**LRC**	**Zeros**

0	0000	0000	0000	0000	0000	0101	1
Parity	**'0'**	**'0'**	**'0'**	**'0'**	**'0'**	**'5'**	**Parity**

✓ **'2'** : **« 4 bits »**
Value is coded by 4 bits only which are sent according the chosen protocol.

Format ISO2 LSB … MSB		
'0'	0000	0x00
'1'	1000	0x01
'2'	0100	0x02
'3'	1100	0x03
'4'	0010	0x04
'5'	1010	0x05
'6'	0110	0x06
'7'	1110	0x07
'8'	0001	0x08
'9'	1001	0x09
'#'	1101	0x0B

In this case, 4 bits are sent LSB First within a frame according the chosen protocol. For more details, refer to the specification protocols.

Example: Frame of the key '5' according the protocol ISO2 / 2b.

0010
'4'

Format WIEGAND MSB … LSB		
'0'	0000	0x00
'1'	0001	0x01
'2'	0010	0x02
'3'	0011	0x03
'4'	0100	0x04
'5'	0101	0x05
'6'	0110	0x06
'7'	0111	0x07
'8'	1000	0x08
'9'	1001	0x09
'#'	1011	0x0B

In this case, 4 bits are sent MSB First within a frame according the chosen protocol. For more details, refer to the specification protocols**.**

Example: Frame of the key '5' according the protocol Wiegand / 3i.

0100
'4'

✓ **'3'** : **« 8 bits »**
Value is coded by 8 bits which are sent according the chosen protocol (default configuration)

Format ISO2 LSB … MSB		
'0'	11110000	0xF0
'1'	01111000	0xE1
'2'	10110100	0xD2
'3'	00111100	0xC3
'4'	11010010	0xB4
'5'	01011010	0xA5
'6'	10010110	0x96
'7'	00011110	0x87
'8'	11100001	0x78
'9'	01101001	0x69

In this case, 8 bits are sent LSB First according the timings of chosen protocol. For more details, refer to the specification protocols.

Example: Frame of the key '4' according the protocol ISO2 / 2b.

11010010
'4'

Format WIEGAND MSB … LSB		
'0'	11110000	0xF0
'1'	11100001	0xE1
'2'	11010010	0xD2
'3'	11000011	0xC3
'4'	10110100	0xB4
'5'	10100101	0xA5
'6'	10010110	0x96
'7'	10000111	0x87
'8'	01111000	0x78
'9'	01101001	0x69

In this case, 8 bits are sent MSB First according the timings of chosen protocol. For more details, refer to the specification protocols.

Example: Frame of the key '4' according the protocol Wiegand 3i.

10110100
'4'

✓ **‘4’** : **« X Keys framed »**
4 bits keys framed – x keys within a frame according the chosen protocol.

Format ISO2 LSB … MSB		
‘0’	0000	0x00
‘1’	1000	0x01
‘2’	0100	0x02
‘3’	1100	0x03
‘4’	0010	0x04
‘5’	1010	0x05
‘6’	0110	0x06
‘7’	1110	0x07
‘8’	0001	0x08
‘9’	1001	0x09

In this case, 4 bits of n keys are sent LSB First within a frame according the chosen protocol. For more details, refer to the specification protocols. Only the keys ‘0’ to ‘9’ are available.

‘★’ Confirms the sequence. If **x=8**, the procedure is automatically confirmed and the code is sent.

‘#’ Cancels the current sequence.

Example: ‘4’ ‘5’ ‘9’ ‘★’ keys are pushed. The frame sent is 4 bits by keys according the protocol ISO2 / 2b.

000…	1101 0	0010 0	1010 1	1001 1	1111 1	xxxx x	000…
Zeros	**Start**	**‘4’**	**‘5’**	**‘9’**	**End**	**LRC**	**Zeros**

Format WIEGAND MSB … LSB		
‘0’	0000	0x00
‘1’	0001	0x01
‘2’	0010	0x02
‘3’	0011	0x03
‘4’	0100	0x04
‘5’	0101	0x05
‘6’	0110	0x06
‘7’	0111	0x07
‘8’	1000	0x08
‘9’	1001	0x09

In this case, 4 bits of n keys are sent MSB First within a frame according the chosen protocol. For more details, refer to the specification protocols. Only the keys ‘0’ to ‘9’ are available

‘★’ Confirms the sequence. If **x=8**, the procedure is automatically confirmed and the code is sent.

‘#’ Cancels the current sequence.

Example: ‘4’ ‘5’ ‘9’ ‘★’ keys are pushed. The frame sent is 4 bits by keys according the protocol Wiegand 3i.

0	0000	0000	0000	0100	0101	1001	1
Parity	**‘0’**	**‘0’**	**‘0’**	**‘4’**	**‘5’**	**‘9’**	**Parity**

Note

✓ *Maximum number of key = 8*

✓ ***xmax** = 6 maximum number of key for Wiegand 3i protocol. In this case values of keys are not automatically sent. It is necessary to confirm the sequence.*

3.2 - Serial reader - R32/R33 - Card OR Keys

No difference between hexadecimal and decimal mode.

The data are coded by 8 bits as shown below:

Value of the key MSB … LSB		
'0'	11110000	0xF0
'1'	11100001	0xE1
'2'	11010010	0xD2
'3'	11000011	0xC3
'4'	10110100	0xB4
'5'	10100101	0xA5
'6'	10010110	0x96
'7'	10000111	0x87
'8'	01111000	0x78
'9'	01101001	0x69

Mono directional mode

Refer to the annex *2.1 - Unidirectional communication mode* for more details about the options of the frame.

Regarding the Card OR Keys configuration, the structure of the frame is:

1 byte	***1 byte** **	***1 byte***	***1 byte***	***1 byte***	***1 byte***
STX	*Key*	*LRC*	*0x0D*	*0x0A*	*ETX*

**Doubled if the ASCII option is activated.*

Bidirectional mode

Refer to the annex *2.2 - Bidirectional communication mode* for more details about the bidirectional communication of the reader.

In Card OR Keys mode, the card data is sent through the ***Output_Protocol.*** The **keyboard data** are sent through the command described below:

Output_Keyboard

Description

This command is generated by the reader when you press a keyboard key in Card OR Key mode. .

Reader: CTRL CMD AAh 55h L_{out} $Data_{out}$

CMD *2 bytes:* 01h 07h

L_{out} *2 bytes:* 00h 01h

$Data_{out}$ *1 byte:* Value of key pressed 8 bits format.

Systeme: CMD L_{in} 00h 00h

CMD *2 bytes:* 01h 07h
L_{in} *2 bytes:* 00h 00h

3.3 - Card THEN Keys

Following the passage of a card, a key sequence is entered on the keyboard (keys 1-8 depending on the configuration). When the reader is waiting for a key sequence, it flashes green quickly.
The keys '0' to '9' can be entered. The keys '★' and '#' cancel the current operation and the entire entry is to start again. It is beyond a 6-second timeout between two keys pressed.

Beyond this timeout, the reader cancels the current operation, indicated by a beep.
Whole sequence 'badge and input key (s) is to start again.

If the input sequence on the keyboard is the data read from the card, the transaction is validated the reader back private ID and green LED lights, otherwise nothing is sent, the refusal of the badge is indicated by the LED red and beep

4 - Glossary

- ✓ **Authentication:** Security mechanism based on an algorithm (AES, Crypto1 etc. ...) using a key.
- ✓ **Private ID:** Private (user) Code.
- ✓ **PUPI:** 14443-B chip serial number .
- ✓ **SSCP:** STid Secure Common Protocol.
- ✓ **UID:** Unique ID, unique chip identification number.

REVISION

Date	Version	Description
12/11/2014	*1.0*	*Initial version*
11/05/2015	*1.1*	*Compatibility SEGIC/ Firmware reader added p.7 / Touchscreen configuration added p12,20,21,22/ On tamper activation keeps LED Red p12,13 / Scramble pad added p19 / Swap added p.23,24 / airbus mode added p23,24,25.*

CONTACT

▶ Plus de 50 pays desservis
More than 50 countries served

SIÈGE SOCIAL
HEADQUARTERS

20, Parc d'Activités des Pradeaux
13850 Gréasque, France
+33 (0)4 42 12 60 60
+33 (0)4 42 12 60 61

AGENCE PARIS - IDF
PARIS - IDF OFFICE

Immeuble Le Trisalys
416 avenue de la division Leclerc
92290 Chatenay Malabry, France
+33 (0)1 43 50 11 43
+33 (0)1 43 50 27 37

AGENCE AUSTRALIE - APAC
AUSTRALIA - APAC OFFICE

5/8 Anzed Court
Mulgrave, 3170
Victoria, Australia
+61 3 8588 4500
+61 3 9560 9055

AGENCE AMÉRIQUE
NORTH & LATIN AMERICA OFFICE

Varsovia 57, Interior 501
Colonia Juárez, CP 06600
Delegación Cuauhtemoc
Mexico, D.F.
+52 (55) 52 56 47 06
+52 (55) 52 56 47 07

AGENCE UK
UK OFFICE

Innovation centre
Gallows Hill, Warwick
CV34 6UW, United Kingdom
+44 (0) 1926 217 884
+44 (0) 1926 217 701

CONTACTS

Sales : info@stid.com
Marketing : marketing@stid.com
Support : support@stid.com

Pour plus d'informations sur les distributeurs, connectez-vous sur www.stid.com
For more information about our distributors, visit www.stid.com