IWM-K531 Wall-mount contactless reader

Reference manual

PMAA060 revision BB
02/08/2007

TABLE OF CONTENT

1. INTRODUCTION

This document provides detailed technical information for use of the Pro-Active wall-mount contactless proximity card reader (IWM-K531).

1.1. AUDIENCE

This reference manual assumes that the reader has expert knowledge of electronics. It is designed for use by system integrators.

1.2. PRODUCT BRIEF

IWM-K531 is a wall-mount proximity reader. It reads serial number from any standard ISO/IEC 14443 contactless card, including popular Philips MIFARE and DESFire families. This reader is primarily dedicated to residential or corporate access control, but can also be used in cash or vending machines.

1.3. OPERATING MODES

Depending on software configuration (stored in EEPROM), the same reader can be operated into 3 modes :

- Wiegand (output only), with configurable frame length,
- Dataclock or ISO2 / Magstripe (output only),
- Serial input/output.

Depending on the underlying hardware (**IWM-K531-WD** or **IWM-K531-SU**), the serial input/output can either be RS-232, RS-485, or USB (USB to serial bridge).

2. CONFIGURING IWM-K531

2.1. HARDWARE JUMPERS

4 jumpers are available for basic configuration of the device.

2.1.1. RS-485, Wiegand and Dataclock hardware

Jumper	ON	OFF
1	Dataclock or Wiegand *according to software configuration*	RS-485
2	Red LED input disabled	Red LED input enabled
3	Green LED input disabled	Green LED input enabled
4	Buzzer enabled	Buzzer disabled

2.1.2. RS-232 and USB hardware

Jumper	ON	OFF
1		
2	Flash mode	Normal mode
3	Buzzer enabled	Buzzer disabled
4		

Switch JP3 allows selection between USB and RS-232 mode.

2.2. CONNECTING IWM-K531 TO A COMPUTER

2.2.1. RS-485, Wiegand and Dataclock hardware

Use one of following Pro-Active interface to connect the reader (through its PC-Link Connector) to a Windows-based computer :

- INT-USB-232 (USB)
- INT-232 (RS-232)

When using INT-USB-232 (USB) interface, you must install the USB Virtual Serial Device driver (VCP).

2.2.2. RS-232 and USB hardware

Connect the reader directly to the computer using either the serial line (RS-232) or the USB interface (type B connector).

When using USB interface, you must install the USB Virtual Serial Device driver (VCP).

2.2.3. *Common information*

Use HyperTerminal or any compliant terminal emulator to get connected onto the reader through the serial port. Default communication settings are :

- 8 data bits, 1 stop, no parity, no flow control ;
- Baudrate = 38400bps.

Note that baudrate may have been changed to 9600bps.

2.3. RETRIEVING IWM-K531 SOFTWARE CONFIGURATION

Connect IWM-K531 to your terminal emulator software according to 2.2.

Reset IWM-K531 (or power it OFF, then ON again). On start-up it displays the string :

`Pro-Active K531 ISO 14443 Reader [xx/yy]`

Part `[xx/yy]` must be understood as follow :

- `xx` is the hardware release ;
- `yy` is the software release.

This documentation applies only to hardware release `AB` or later, and software release `08` or later.

Send the command :

`info` <enter>

to retrieve IWM-K531 information (K531 core firmware release version, RC531 hardware identifier and serial number, explanation of active configuration).

Send the command :

`cfg` <enter>

to read current IWM-K531 software configuration constant (see 2.5 for explanation of IWM-K531 software configuration constant).

IWM-K531 doesn't echo back the received characters.

2.4. SETTING IWM-K531 SOFTWARE CONFIGURATION

2.4.1. Through serial line

Connect IWM-K531 to your terminal emulator software according to 2.2.

Send the command :

`cfg=xxxxxxxx` <enter>

where `xxxxxxxx` is the new software configuration constant (see 2.5 for explanation of IWM-K531 software configuration constant). IWM-K531 acknowledges with "OK".

Reset IWM-K531 to activate new configuration.

IWM-K531 doesn't echo back the received characters.

Be sure to send the configuration constant without error.

2.4.2. Using a configuration tag

Configuration tags can be bought at Pro-Active.

Put the relevant configuration tag in front of the reader.

Reset IWM-K531 (or power it OFF, then ON again). On start-up it reads the configuration tag, and load the new configuration constant.

Remove the configuration tag and reset IWM-K531 to activate new configuration.

2.5. IWM-K531 SOFTWARE CONFIGURATION CONSTANT

The IWM-K531 software configuration constant is a 4-byte value (8 hexadecimal digits) that is stored in reader's non-volatile memory (E^2PROM).

Bytes are sent MSB first (most significant bit first).

Byte	*Defines...*
0	Serial configuration
1	Reader operating mode
2	Output format
3	Handled tags

2.5.1. Serial configuration

First byte defines serial configuration.

Description of the bits :

Bit	***Name***	***Function***
7		RFU *(must be 0)*
6-4	SC.ADR	**Address of device on the bus *(Serial only)* :** 000 addressing disabled (single device) 001 to 111 selected device address
3		RFU *(must be 0)*
2-0	SC.BDR	**Baud rate *(Serial only)* :** 000 38400 bps 001 19200 bps 010 9600 bps 011 4800 bps 100 115200 bps 101 to 111 RFU

The baudrate parameter is common to USB, RS-232 and RS-485 interfaces.

Do not set baudrate to 115200bps when working with RS-485 interface, as the hardware and the characteristics of the bus aren't able to support it.

2.5.2. Reader operating mode

Second byte defines reader-operating mode.

Description of the bits :

Bit	Name	Function
7	RO.ACT	**Accept configuration tags** 1 Configuration through *configuration tags* is disabled 0 *Configuration tags* can be used to configure the reader
6-5		RFU *(must be 000)*
4	RO.PWM	**Power up mode *(Serial only)*** 1 Reader waits for a command from the host before starting up. 0 Reader starts up automatically at power up.
3	RO.RMV	**Tag filtering policy :** 1 Tag must be removed before being read again. 0 Tag are continuously read after the "filtering delay".
2-1	RO.FLT	**Tag filtering delay :** 00 Tag are filtered for 1 second 01 Tag are filtered for 2 seconds 10 Tag are filtered for 5 seconds 11 Tags are not filtered (100 to 200ms delay).
0	RO.MOD	**Output mode select *(Wiegand / Dataclock only)*** 1 Device is a Wiegand reader. 0 Device is a Dataclock reader. **Output mode select *(Serial only)*** 1 Device is an RS-232 reader. 0 Device is an RS-485 reader.

2.5.3. Output format

Third byte defines reader output format.

Description of the bits :

Bit	***Name***	***Function***
7	OF.RAW	**Decorate output frames *(Serial only)*** 1 Serial output uses only STX and ETX as frame markers. 0 Frames are "decorated" with BEL and CR LF markers.
6	OF.ACK	**Acknowledgement mode *(Serial only)*** 1 Pending transmissions are discarded by host acknowledge. 0 Host acknowledge has no effect.
5	OF.RAI	**ISO/IEC 14443-A UID bytes order** 1 Reader sends type A UIDs MSB first. 0 Reader sends type A UIDs LSB first. This is default behaviour since Philips literature uses this order for Mifare serial numbers.
4	OF.ATT	**Add tag type to identifier** 1 Tag identifier output is prefixed with tag type : • 'A0' + ATQ + SAK for ISO/IEC 14443-A, • 'B0' + whole ATQB for ISO/IEC 14443-B, • 'B176' for ST Microelectronics SR 176 and compliant, • 'C0' for Calypso 0 Tag identifier is sent without prefix.
3		RFU *(must be 0)*
2-0	OF.FMT	**Output size and transcription** 000 RFU 001 RAW output, 4 bytes (32 bits) 010 RAW output, 8 bytes (64 bits) 011 RAW output, 16 bytes (128 bits) 101 BCD output, 10 digits 110 BCD output, 12 digits 111 RAW output, 6 bytes (48 bits)

2.5.4. Handled tags

Fourth byte defines which tags the reader handles.

Description of the bits :

Bit	***Name***	***Function***
7		RFU *(must be 0)*
6	HT.ICA	1 Calypso identifiers are read *(feature subject to Calypso licence agreement and fee).* 0 Calypso cards are ignored.
5		RFU *(must be 0)*
4		RFU *(must be 0)*
3	HT.IAS	1 Read ASK CTS256 identifiers. 0 Ignore ASK CTS256 cards.
2	HT.ISR	1 Read ST Microelectronics SR176 identifiers. 0 Ignore ST Microelectronics SR176 cards.
1	HT.IIB	1 Read ISO/IEC 14443-B Pseudo-Unique Identifiers (PUPI). 0 Ignore ISO/IEC 14443-B.
0	HT.IIA	1 Read ISO/IEC 14443-A Unique Identifiers (UID or SNR). 0 Ignore ISO/IEC 14443-A.

3. RS-485 APPLICATION NOTE

3.1. WIRING

3.1.1. *Data stream format : binary*

a. Fixed length binary

Data is transmitted as an ASCII string. Each character is the hexadecimal representation (from '0' to 'F') of half a byte. Transmission is done most significant byte first.

If data is shorter than specified length, it is left-prefixed with binary 0s.

If data is longer than specified length, it is right truncated (only lowest significant bytes are transmitted).

DF.FMT values for RS-485 fixed-length binary

value for DF.FMT	*Meaning*	*Sample output (ASCII strings)*
001	4 bytes	*7990D030*
010	8 bytes	*000467257990D030*
011	16 bytes	*00000000000000000467257990D030*

b. Variable length binary

Data is transmitted as an ASCII string. Each character is the hexadecimal representation (from '0' to 'F') of half a byte. Transmission is done most significant byte first.

The exact length of available data is transmitted.

DF.FMT values for RS-485 variable-length binary

value for DF.FMT	*Meaning*	*Sample output (ASCII string)*
100	var. length	*0467257990D030*

3.1.2. *Data stream format : BCD*

a. Fixed length BCD

Only 32 lowest significant bits of data are used (4 bytes).

Data is first converted to a 32-bits number (from 0 to 4294967295).

Date is transmitted as an ASCII string. It provides the fixed-length decimal representation of the number.

DF.FMT values for RS-485 fixed-length BCD

value for DF.FMT	*Meaning*	*Sample output (ASCII string)*
101	10 BCD digits	*2039533616*
110	12 BCD digits	*002039533616*

3.1.3. Frame format without addressing

Serial frames are built according to following format :

1. Start sentinel : ASCII STX character (`0x02`),
2. Actual data string (ASCII characters, '0' to '9' and 'A' to 'F'),
3. End sentinel : ASCII ETX character (`0x03`).

If OF.RAW is set to 0,

- ASCII BEL character (`0x07`) is sent before the start sentinel,
- ASCII CR and LF characters (`0x0D` and `0x0A`) are sent after the end sentinel.

3.1.4. Frame format with addressing enabled

Serial frames are built according to following format :

1. Start sentinel : ASCII SOH character (`0x01`),
2. Reader address (ASCII character, from '1' to '7'),
3. Separator : ASCII STX character (`0x02`),
4. Actual data string (ASCII characters, '0' to '9' and 'A' to 'F'),
5. End sentinel : ASCII ETX character (`0x03`).

If OF.RAW is set to 0,

- ASCII BEL character (`0x07`) is sent before the start sentinel,
- ASCII character '>' (`0x3E`) is sent between the address and the separator,
- ASCII CR and LF characters (`0x0D` and `0x0A`) are sent after the end sentinel.

3.2. COMMAND SET

IWM-K531 accepts commands from the host.

Command transmission format is <command> <CR> <LF>.

Please observe that IWM-K531 never echoes received data.

If received command has been understood by IWM-K531, it replies with <ACK> before executing the requested action. Otherwise, it doesn't reply and does nothing.

Command	***Action***
A0	Reader goes inactive (tag polling is halted)
A1	Reader goes active
R0	Switch red LED off
R1	Switch red LED on
R2	Red LED blinks slowly
R3	Red LED blinks quickly
G0	Switch green LED off
G1	Switch green LED on
G2	Green LED blinks slowly
G3	Green LED blinks quickly
M*xyz*	Same as sending 3-command sequence A*x* R*y* G*z*
Rst	Reset the reader
Ver	Retrieve reader's version
Cfg	See 2.3

If RO.PWM is set to 1, after power up reader does nothing until it receives the "A1" command from the host.

Set jumpers 2 & 3 to ON to allow device to control its LEDs.

4. WIEGAND APPLICATION NOTE

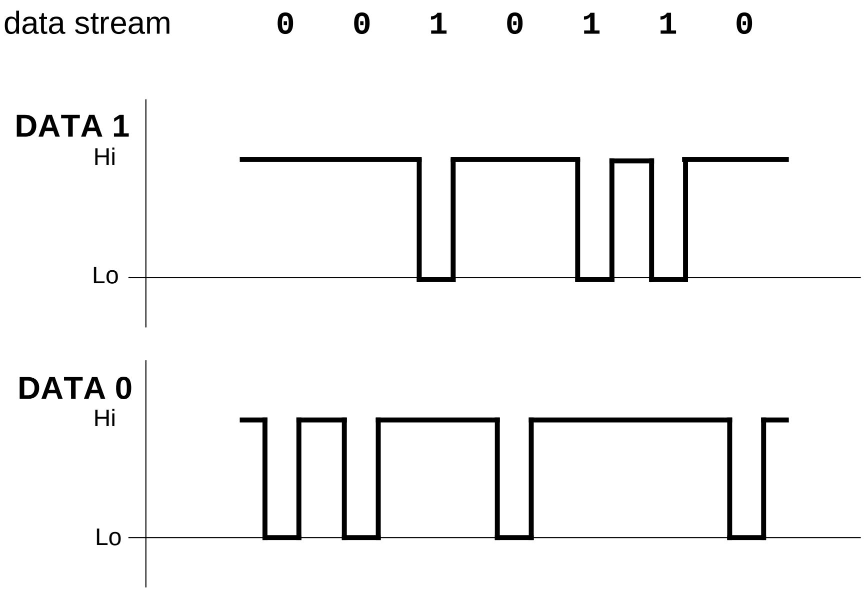
4.1. THE WIEGAND INTERFACE

4.1.1. *Bit format*

Pins 5 and 6 are Wiegand DATA0 and DATA1 outputs, respectively.

- Both pins are at high level when idle,
- A low pulse on DATA0 denotes a bit 0 output,
- A low pulse on DATA1 denotes a bit 1 output.

In normal operation, DATA0 and DATA1 are never at low level simultaneously.

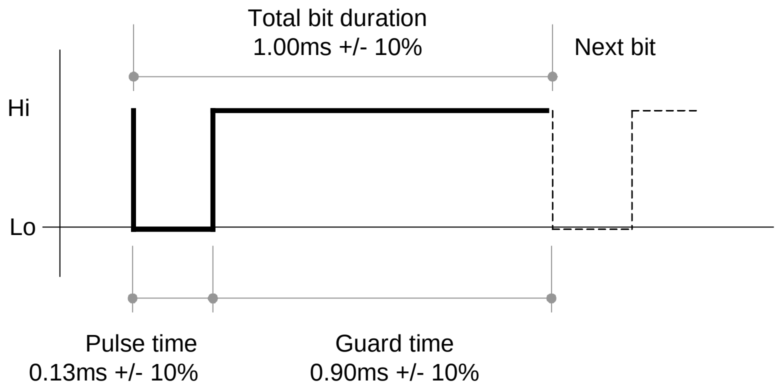


a. *Electrical levels*

	Level to GND
Output level high	4.0V min, 5.5V max
Output level low	0.0V min, 1.0V max

DATA0 and DATA1 are not open collector output. Internal pull-up resistors are included in the reader.

b. Timings



4.1.2. Data stream format

a. Fixed length binary

Data are transmitted in binary, most significant byte first, most significant bit first in each byte.

If data is shorter than specified length, it is left-prefixed with binary 0s.

If data is longer than specified length, it is right truncated (only lowest significant bytes are transmitted).

DF.FMT values for Wiegand fixed-length binary

value for DF.FMT	*Meaning*	*Sample output (raw)*
001	4 bytes	*7990D030*
010	8 bytes	*000467257990D030*
011	16 bytes	*00000000000000000467257990D030*

b. Variable length binary

Data are transmitted in binary, most significant byte first, most significant bit first in each byte.

The exact length of available data is transmitted.

DF.FMT values for Wiegand variable-length binary

value for DF.FMT	*Meaning*	*Sample output (raw)*
100	var. length	*0467257990D030*

c. Fixed length BCD

Only 32 lowest significant bits of data are used (4 bytes).

Data is first converted to a 32-bits number (from 0 to 4294967295).

Date is transmitted in BCD. It provides the fixed-length decimal representation of the number.

DF.FMT values for Wiegand fixed-length BCD

value for DF.FMT	*Meaning*	*Sample output (raw)*
101	10 BCD digits	*2039533616*
110	12 BCD digits	*002039533616*

4.1.3. Frame format

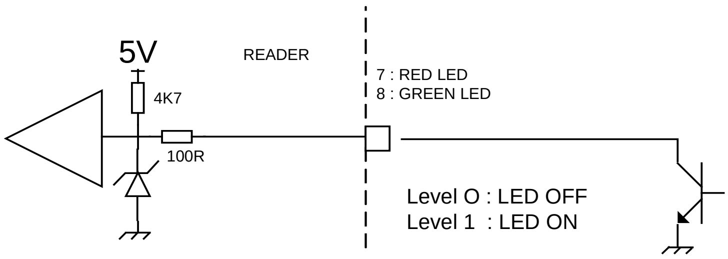
With Wiegand output, there is no frame marker (no preamble, no epilogue), and no error control (no parity, no CRC).

4.2. LED INTERFACE

Pins 7 and 8 are red and green LEDs inputs, respectively.

	Meaning	*Level to GND*
Input level high	LED is ON	3.3V to 5.5V
Input level low	LED if OFF	0.0V to 1.7V

The reader has an internal pull-up resistor to 5V.



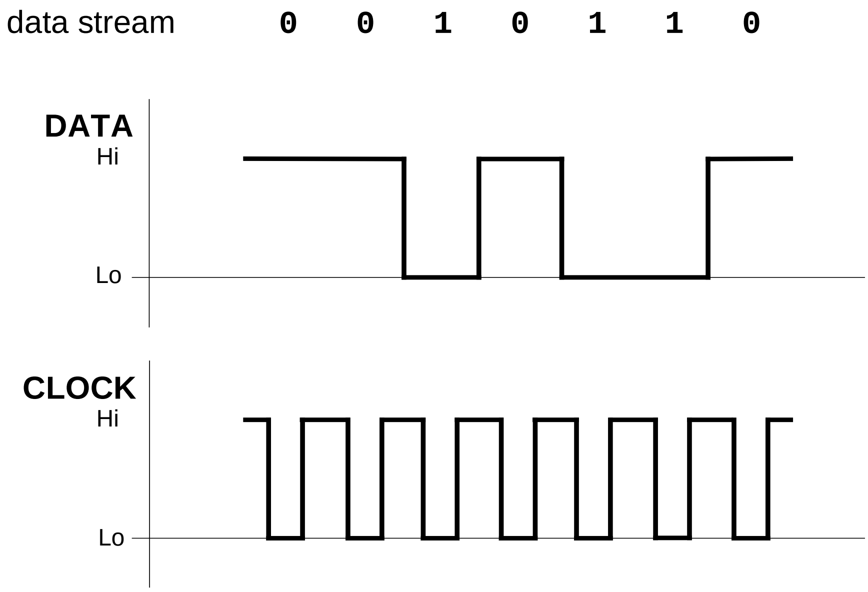
Set jumpers 2 & 3 to OFF to enable LED inputs.

5. DATACLOCK APPLICATION NOTE

5.1. THE DATACLOCK INTERFACE

Pins 5 and 6 are Wiegand DATA0 and DATA1 outputs, respectively.

- Both pins are at high level when idle,
- The CLOCK line is active low,
- The DATA line is inverting (low level means 1, high level means 0).

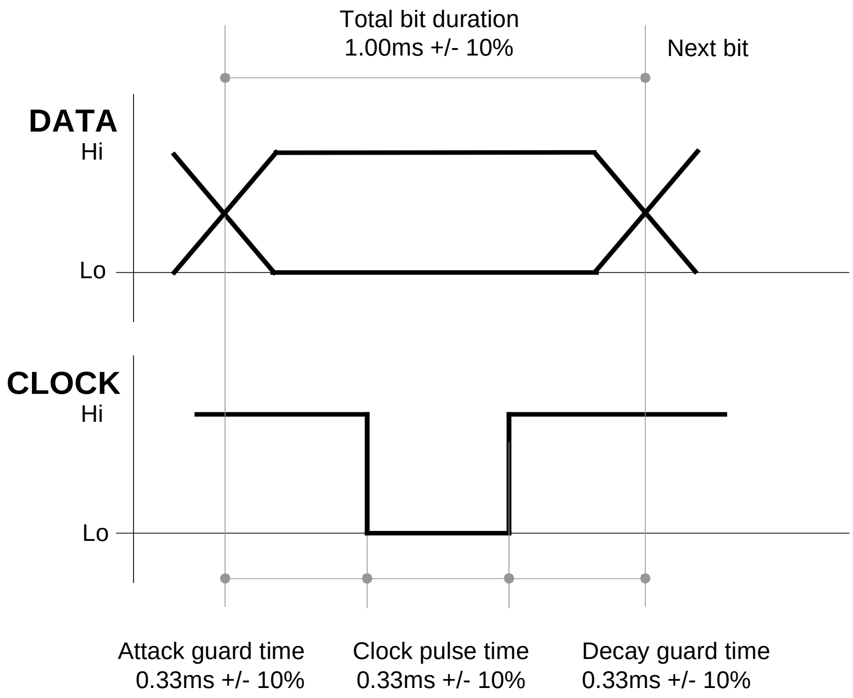


a. Electrical levels

	Level to GND
Output level high	4.0V min, 5.5V max
Output level low	0.0V min, 1.0V max

DATA and CLOCK are not open collector output. Internal pull-up resistors are included in the reader.

b. Timings



5.1.2. Digit format

Dataclock only transmit decimal data. Each digit is transmitted as 5 bits :

- 4 digit bits, least significant bit first,
- 1 parity bit.

Data are BCD-encoded, i.e. only decimal values from 0 to 9 are valid for data digits. Values above 10 (hexadecimal values from A to F) are reserved for start and stop markers ("sentinels") and for frame LRC.

Dataclock digit format

Value	*Bit pattern*
0	0 0 0 0 1
1	1 0 0 0 0
2	0 1 0 0 0
3	1 1 0 0 1
4	0 0 1 0 0
5	1 0 1 0 1
6	0 1 1 0 1
7	1 1 1 0 0
8	0 0 0 1 0
9	1 0 0 1 1

Value	*Bit pattern*
A (10)	0 1 0 1 1
B (11)	1 1 0 1 0
C (12)	0 0 1 1 1
D (13)	1 0 1 1 0
E (14)	0 1 1 1 0
F (15)	1 1 1 1 1

5.1.3. Data stream format

a. Fixed length BCD

Only 32 lowest significant bits of data are used (4 bytes).

Data is converted to decimal and transmitted in BCD, most significant digit first, with parity bits and according to the Dataclock frame output format.

If data is shorter than specified length, it is left-prefixed with decimal 0s.

DF.FMT values for Dataclock fixed-length BCD

value for DF.FMT	Meaning	Sample output (raw)
101	10 BCD digits	*2039533616*
110	12 BCD digits	*002039533616*

b. Fixed length binary

Since Dataclock can only transmit decimal data, a transcription must be done to translate arbitrary digits (from 0 to F) into BCD digits (from 0 to 9).

The transcription is as follow : for each hexadecimal digit is represented as two decimal digits ; first digit is value divided by 10 (either 0 or 1), second digit is value modulo 10.

Transmitted length is twice as fixed length.

DF.FMT values for Dataclock fixed-length binary

value for DF.FMT	Meaning	Sample output (raw)
001	4 bytes	*0709090013000300*
010	8 bytes	*00000004060702050709090013000300*
011	16 bytes	*(…) 0000000000406070205070909013000300*

(Note that D is translated into 13)

c. Variable length binary

Transcription is done in the same way as described in last paragraph.

Transmitted length is twice as actual length of available data.

DF.FMT values for Dataclock variable-length binary

value for DF.FMT	Meaning	Sample output (raw)
100	var. length	*00040607020507090900**13**000300*

(Note that D is translated into 13)

5.1.4. Frame format

Dataclock frames are built according to following format :

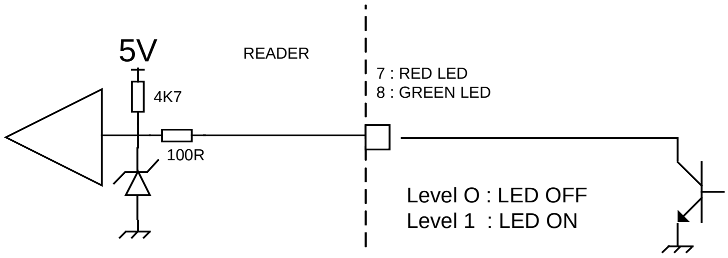
6. Left edge : bit 0 is transmitted 16 times,
7. Start sentinel (hexadecimal digit B, i.e. bit pattern 1 1 0 1 0),
8. Actual data digits,
9. Stop sentinel (hexadecimal digit F, i.e. bit pattern 1 1 1 1 1),
10. LRC of frame (XOR computed over parts 1, 2 and 3),
11. Right edge : bit 0 is transmitted 16 times.

5.2. LED INTERFACE

Pins 7 and 8 are red and green LEDs inputs, respectively.

	Meaning	*Level to GND*
Input level high	LED is OFF	3.3V to 5.5V
Input level low	LED if ON	0.0V to 1.7V

The reader has an internal pull-up resistor to 5V.



Set jumpers 2 & 3 to OFF to enable LED inputs.

6. APPENDIX : MASTER CARDS

6.1. BUILDING A MASTER CARD

- The Master Card must be a Mifare 1k (or 4k).
- Both sector 0 and 1 must be readable with default Mifare key `A0A1A2A3A4A5`.
- Sector 0 must hold the constant string `PRO-ACTIVE_FR:IWM-K531:CONFIG1` starting at offset 18.
- Block 1.0 holds the configuration settings to be applied (4 bytes) repeated 4 times (i.e. 16 bytes).
- Block 1.1, is RFU and set to 16 zero bytes.
- Block 1.2, stores the digital signature used to authenticate the tag.

6.2. DIGITAL SIGNATURE ALGORITHM

- Let *Key* be a 16-byte arbitrary buffer.
- Construct the MD5 hash of *block 1.0* || *block 1.1* || *key* || *block 0.0* || *block 0.1* || *block 0.2*.
- Use this 16-byte hash as the digital signature (stored it in block 1.2 of the tag).

Value of *Key* is confidential. Only Pro-Active genuine software are able to sign Master Cards.