CR08 Card Reader

User Manual

Version v1.0

Disclaimer

Before using the product, please read all the contents in this product manual carefully to ensure the safe and effective use of the product. Do not disassemble the product or tear up the seal on the device by yourself, or Beijing Vguang Internet Technology Co., Ltd. will not be responsible for the warranty or replacement of the product.

The pictures in this manual are for reference only. If any individual pictures do not match the actual product, the actual product shall prevail. For the upgrade and update of this product, Beijing Vguang Internet Technology Co., Ltd. reserves the right to modify the document at any time without notice.

Use of this product is at the user's own risk. To the maximum extent permitted by applicable law, damages and risks arising from the use or inability to use this product, including but not limited to direct or indirect personal damage, loss of commercial profits, Beijing vguang Internet Technology Co., Ltd. will not bear any responsibility for trade interruption, loss of business information or any other economic loss.

All rights of interpretation and modification of this manual belong to Beijing Vguang Internet Technology Co., Ltd.

Edit history

Change date	Version	Description	Responsible
2021.10.18	V1.0	Initial version	
2022.5.16	V2.0	Update protocol	

Catalog

1. Preface

Thanks for using the CR08 card reader. Reading this manual carefully can help you understand the function and features of this device, and quickly master the use and installation of the device.

1.1. Product introduction

CR08 card swiping device is a product specially developed for the field of access control card swiping. It has two output interfaces, Wiegand and RS485, and supports Wiegand 26/34 protocol switching.

1.2. Product features

1, Supports Wiegand/RS485 output interfaces.

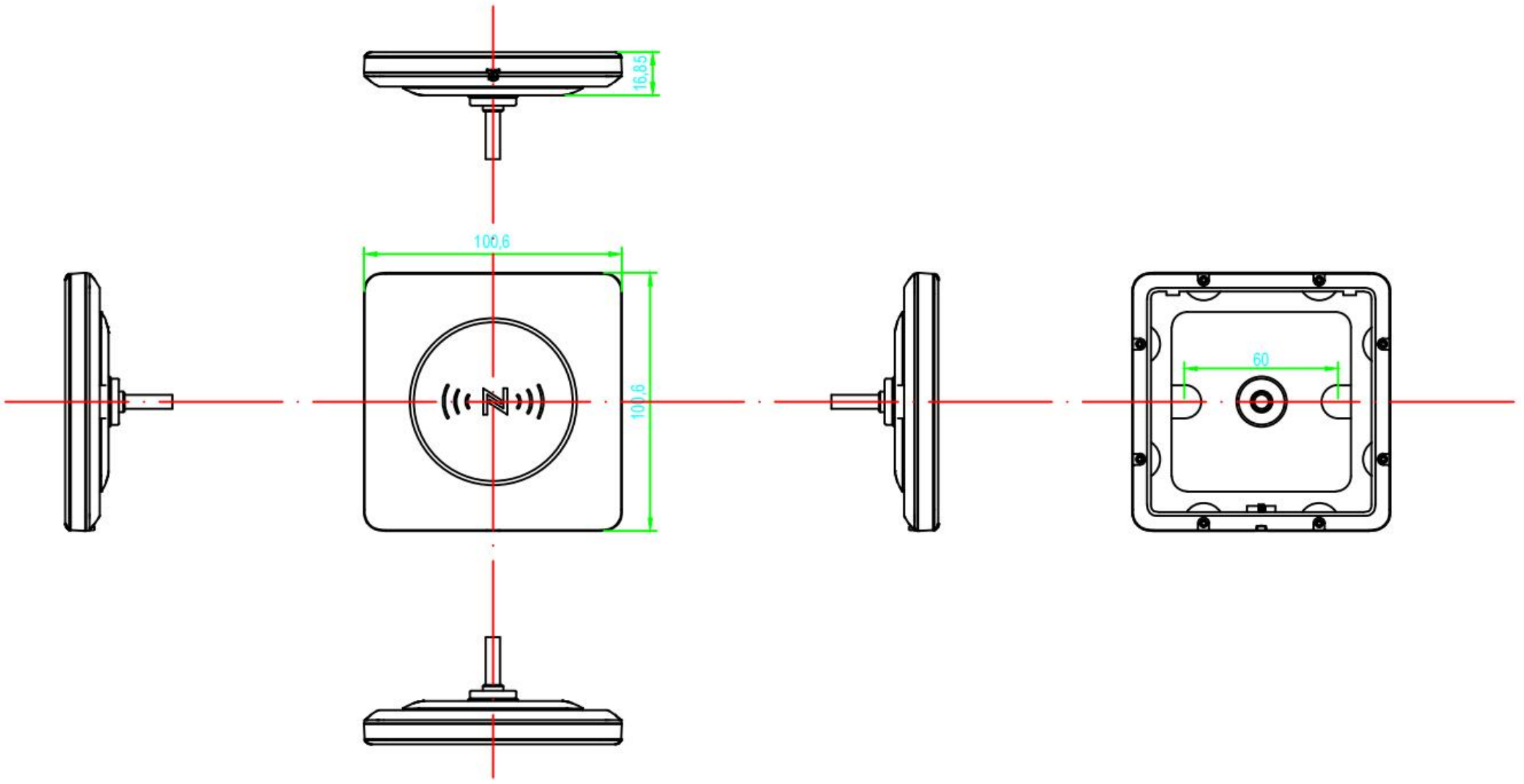
2, Supports PSAM card verification.

3, 86 box installation.

2. Product appearance

2.1.1. APPEARANCE PICTURE

2.1.2. PRODUCT SIZE CHART



3. Product parameters

3.1. General parameters

General parameters	
Support interface	RS485, Wiegand
Indication mode	Red light, green light, white light, blue light, buzzer
Installation method	Embedded
Product size	100.60mm*100.60mm*16.85mm

3.2. Reading parameters

Reading parameters of RF card	
Identification card type	ISO 14443A protocol card, ISO 14443B protocol card, ID card (read physical card number only)
Operation card mode	Read UID, read and write M1 card sector, PSAM authentication
RF operating frequency	13.56MHz
Reading distance	＜5cm, related to card type and specification

3.3. Electrical parameters

The power input can be provided only when the device is connected properly. If the device is plugged or unplugged while the cable is live (hot plugging), its electronic components will be damaged. Make sure that the power is turned off when plugging and unplugging the cable.

Poor power connection, power off and on operation with too short interval, or excessive voltage drop pulse may cause the device to be unable to work stably and normally, so it is necessary to keep the power input stable. After the power input is turned off, the power input can be turned on again after an interval of more than 2 seconds.

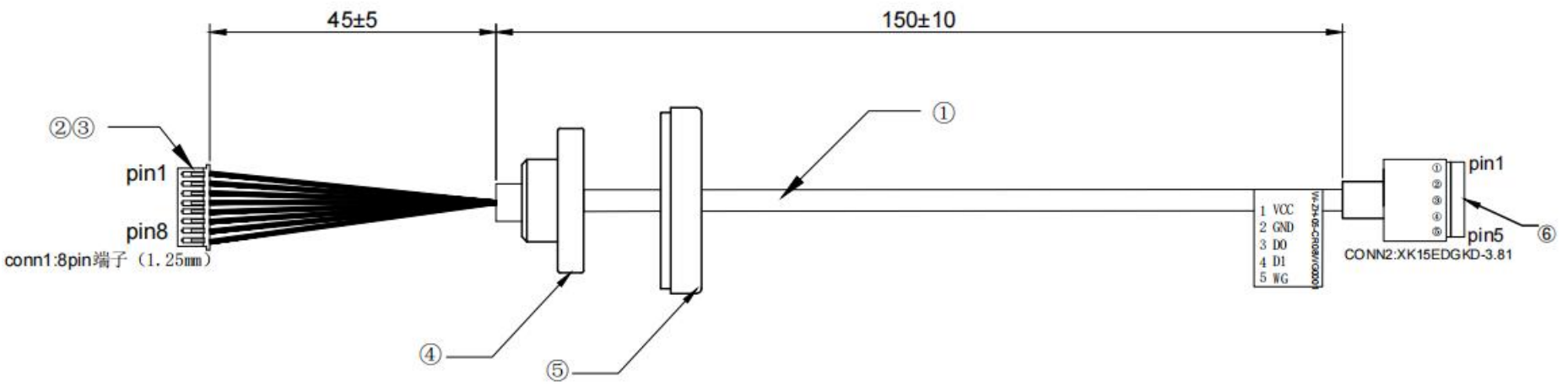
Electrical parameters	
Working voltage	DC 5V-15V
Working current	150mA（Typical 12V power supply）
Rated power consumption	1800mW（Typical 5V power supply）

3.4. Working environment

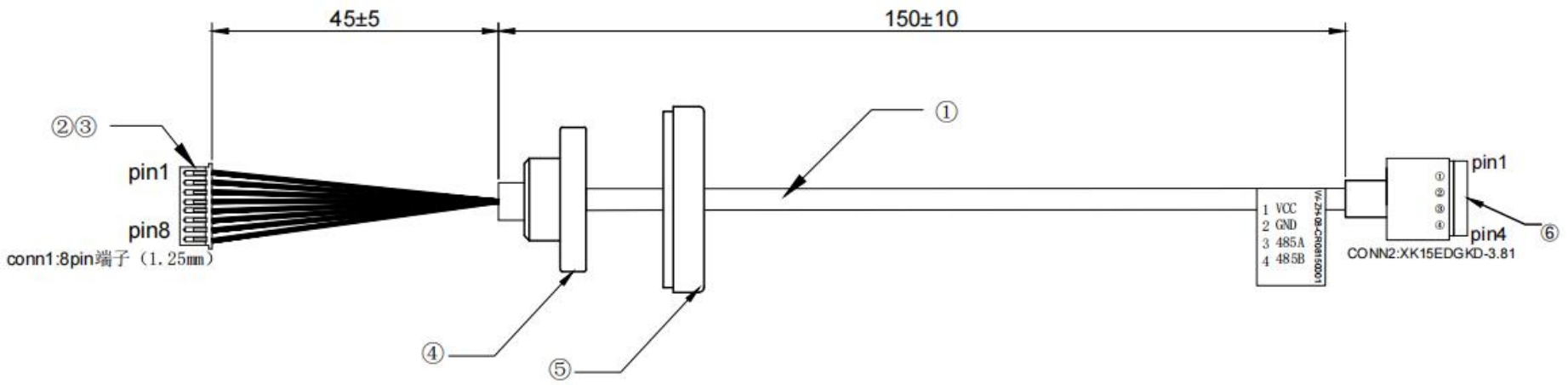
Working environment parameters	
Electrostatic protection	±15kV（Air discharge）, ±6kV（Contact discharge）
Working temperature	-20° C-70° C
Storage temperature	-20° C-80° C
Relative humidity	5%-95%（Non condensing）

4. Connector

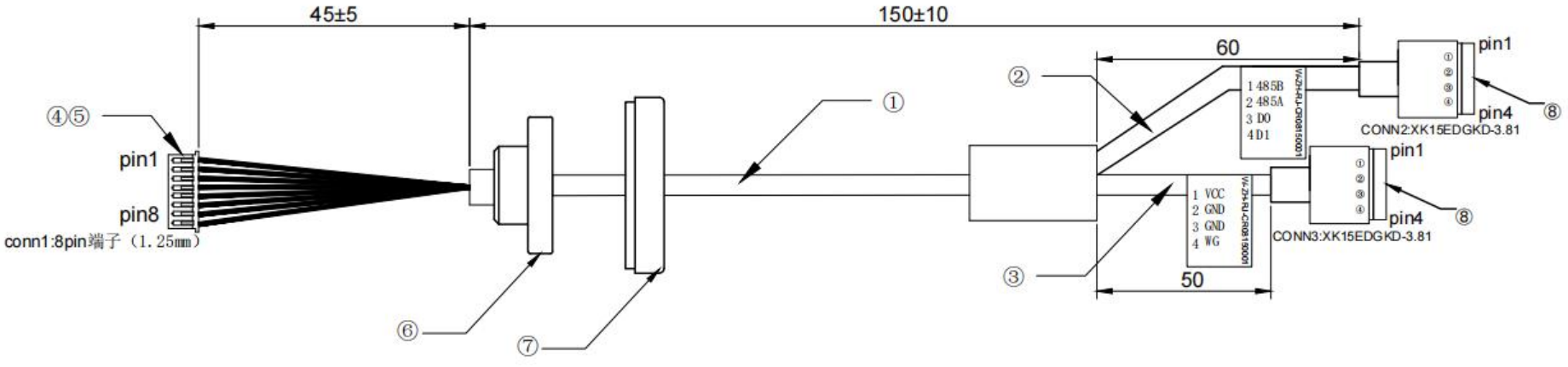
4.1. Wiegand connector



4.2. 485 connector



4.3. Wiegand+485 connector

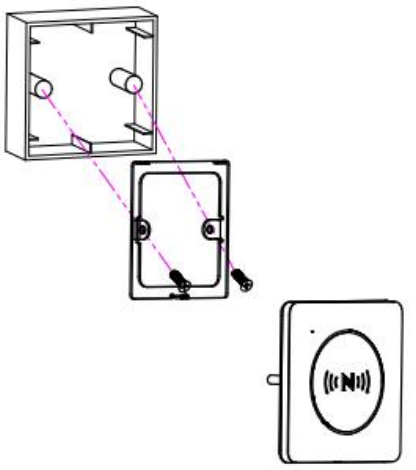
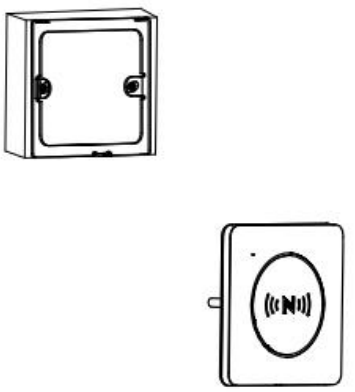
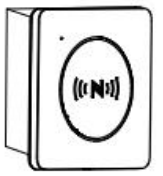


4.4. Pin description

Pin definition	Pin description
VCC	Power supply positive pole
GND	Power supply ground
D0	Wiegand 0
D1	Wiegand 1
485A	485A cable
485B	485B cable
WG	Wiegand protocol for setting pin Overhang:Wiegand 26 Low level:Wiegand 34

5. Installation method

The RF card reading antenna is located at the lower side of the panel. During installation, no metal and magnetic substances should be found within 10cm, or the card swiping performance will be seriously reduced.

			
Step 1: Confirm that the removal hole is downward.	Step 2: Fix the bracket to the 86 box.	Step 3: Lock the bracket with 2 M4 * 20 cross countersunk screws.	Step 4: Fix the product to the bracket and complete the installation.

6. Configuration Instructions

6.1. Data transmission protocol

6.1.1. REQUEST DATA FORMAT

Command header + command word + identification word + length word + data field + check word

Command header: two bytes, the default is 0x55, 0XAA

Command word: one byte

Length word: two bytes, indicating the number of bytes of this command from the end of the length word to the verification word (excluding the verification word), with the low order first

Data field: this item can be blank (For data field contents, refer to Table 6.2)

Check word: one byte, byte by byte XOR value from the command header to the last byte of the data field

6.1.2. RESPONSE DATA FORMAT

Command header + command word + identification word + length word + data field + check word

Command header: two bytes, the default is 0x55, 0XAA

Command word: one byte

Identification word: one byte, 0x00 represents successful response, other failures or errors

Length word: two bytes, indicating the number of bytes of this command from the end of the length word to the verification word (excluding the verification word), with the low order first

Data field: this item can be blank

Check word: one byte by byte XOR value from the command header to the last byte of the data field

6.2. Configuration item description

The table is the content order of configuration items from top to bottom: command type, baud rate（115200-8-n-1 by default）, device number（1 by default）, light configuration (including backlight, card swiping action light), other configurations, modification of package header, prefix, suffix, command mode acquisition interval, and output configuration.

<table>
<tr><th>Configuration Item</th><th>Data length</th><th>Instructions</th></tr>
<tr><td>Command type</td><td>1 byte</td><td>0x00：Get device configuration
0x01：Reconfigure the device according to the Instructions</td></tr>
<tr><td>Baud rate</td><td>4 bytes</td><td>Baud rate 115200 → 00 01 C2 00</td></tr>
<tr><td>Device number</td><td>4 bytes</td><td>Device number (not effective at present)</td></tr>
<tr><td>Light configuration</td><td>1 byte</td><td>
<table>
<tr><th colspan="4">Backlight</th><th colspan="4">Card swiping action light</th></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Blue</td><td>Green</td><td>Red</td><td>White</td><td>Blue</td><td>Green</td><td>Red</td><td>White</td></tr>
</table>
</td></tr>
<tr><td>Other configuration</td><td>1 byte</td><td>
<table>
<tr><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>Distinguish card type</td><td>0x01:Active Report Card Number
0x00:Command mode</td><td colspan="4">Empty</td><td>Relay</td><td>Buzzer</td></tr>
</table>
</td></tr>
<tr><td>Change header</td><td>2 bytes</td><td>Header (default is 0x55 0xaa)
Note: If you modify the header, all subsequent protocols sent will need to use the modified header</td></tr>
<tr><td>Prefix</td><td>3 bytes</td><td>
<table>
<tr><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th></tr>
<tr><td>Effective length</td><td>Prefix 1</td><td>Prefix 1</td></tr>
</table>
</td></tr>
<tr><td>Suffix</td><td>3 bytes</td><td>
<table>
<tr><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th></tr>
<tr><td>Effective length</td><td>Suffix 1</td><td>Suffix 2</td></tr>
</table>
</td></tr>
<tr><td>command mode acquisition interval</td><td>1 byte</td><td>Unit 25 ms</td></tr>
</table>

<table>
<tr>
<td>Output configuration</td>
<td>1 byte</td>
<td>
<table>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td></tr>
<tr><td colspan="2">Zero setting</td><td>Output card number in protocol mode</td><td>Output card number</td></tr>
<tr><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td>Reverse sequence output</td><td>String output</td><td>Decimal output</td><td>Hex output</td></tr>
</table>
Bit 7 Bit6 must zero setting
Bit 5 outputs the card number in the communication protocol format, refer to the communication protocol description
Bit 4 only uploads card number data
Bit 2 ~ Bit0 have and only one can be set to 1, others must be set to 0.
</td>
</tr>
</table>

6.3. Example of configuration Instructions

6.3.1. RECONFIGURE DEVICE

The settings are as follows:

Command Type Reconfigure Device: 01

Baud rate 115200: 00 01 C2 00

Device number 1: 00 00 00 01

Light configuration （backlight）blue light is always on, (card swiping action) flashing green light: 84

Other configurations buzzer ring, actively report the card number, not distinguish type: 41

Head of contracted labour 0x55 0xaa: 55 AA

No prefix and suffix: 00 00 00 00 00 00

Command mode acquisition interval 2 seconds: 50

Output configuration Output card number (non protocol mode) in hex hex (positive sequence): 11

PC->Reader： 55 AA B0 15 00 01 00 01 C2 00 00 00 00 01 84 41 55 AA 00 00 00 00 00 00 50 11 E2

（The data length is 21, the hexadecimal bit is 0x15, the length word is two bytes, the low bit is first, it is 15 00）

Reader->PC： 55 AA B0 00 00 00 4F （The fourth byte 00 represents successful setting, and other values represent failure）

6.3.2. GET DEVICE CONFIGURATION

PC->Reader： 55 AA B0 01 00 00 4E

（When the instruction type is 00, the data length is only 1, so the length word is 01 00）

Reader->PC： 55 AA B0 00 15 00 01 00 01 C2 00 00 00 00 01 84 41 55 AA 00 00 00 00 00 00 50 11 E2

（The fourth byte 00 represents successful setting, and other values represent failure）

7. Communication protocol

7.1. Data transmission protocol

7.1.1. REQUEST DATA FORMAT

Command header + command word + identification word + length word + data field + check word

Command header: two bytes, the default is 0x55, 0XAA

Command word: one byte

Length word: two bytes, indicating the number of bytes of this command from the end of the length word to the verification word (excluding the verification word), with the low order first

Data field: this item can be blank

Check word: one byte, byte by byte XOR value from the command header to the last byte of the data field

7.1.2. RESPONSE DATA FORMAT

Command header + command word + identification word + length word + data field + check word

Command header: two bytes, the default is 0x55, 0XAA

Command word: one byte

Identification word: one byte, 0x00 represents successful response, other failures or errors

Length word: two bytes, indicating the number of bytes of this command from the end of the length word to the verification word (excluding the verification word), with the low order first

Data field: this item can be blank

Check word: one byte by byte XOR value from the command header to the last byte of the data field

7.2. Card number reporting format in protocol mode

7.2.1. NOT DISTINGUISH CARD TYPE

The instruction 0x30 obtains the result regardless of the data source

<table>
<tr><td colspan="3">0x30</td><td colspan="3">The upper computer actively polls or the device actively reports the results.</td></tr>
<tr><td colspan="6">Note: The data returned by this instruction does not distinguish between card types.</td></tr>
<tr><td colspan="3">PC->Reader(Send)</td><td colspan="3">Reader->PC(Receive)</td></tr>
<tr><td>Project</td><td>Byte</td><td>Instructions</td><td>Project</td><td>Byte</td><td>Instructions</td></tr>
<tr><td>Packet header</td><td>2 Byte</td><td>Default: 0x55 0xAA</td><td>Packet header</td><td>1 Byte</td><td>Default: 0x55 0xAA</td></tr>
<tr><td>Command word</td><td>1 Byte</td><td>0x30</td><td>Command word</td><td>1 Byte</td><td>0x30</td></tr>
<tr><td>Data field length</td><td>2 Byte</td><td>0x00 0x00</td><td>Identification word</td><td>1 Byte</td><td>0x00 : success
Not 0 : failed</td></tr>
<tr><td rowspan="2">Data field</td><td rowspan="2">0 Byte</td><td rowspan="2">No such item</td><td>Data field length</td><td>2 Byte</td><td>N</td></tr>
<tr><td>Data field</td><td>N Byte</td><td>When data N=0, no such item</td></tr>
<tr><td>Check word</td><td>1 Byte</td><td></td><td>Check word</td><td>1 Byte</td><td></td></tr>
</table>

For example：
In command mode, the upper computer sends the command to acquire data by polling
PC-->Reader :55 AA 30 00 00 CF
Reader-->PC :55 AA 30 00 00 00 CF No data
Reader-->PC :55 AA 30 00 08 00 37 36 64 30 33 34 39 31 9D Feedback data

7.2.2. DISTINGUISH CARD TYPE

The instruction 0x33 obtains the result and distinguishes the card type

For example：

<table>
<tr><td colspan="3">0x33</td><td colspan="4">The upper computer actively polls or the device actively reports the results.</td></tr>
<tr><td colspan="7">Note: The data returned by this instruction is distinguish between card types.</td></tr>
<tr><td colspan="3">PC->Reader(Send)</td><td colspan="4">Reader->PC(Receive)</td></tr>
<tr><td>Project</td><td>Byte</td><td>Instructions</td><td>Project</td><td>Byte</td><td colspan="3">Instructions</td></tr>
<tr><td>Packet header</td><td>2 Byte</td><td>Default: 0x55 0xAA</td><td>Packet header</td><td>1 Byte</td><td colspan="3">Default: 0x55 0xAA</td></tr>
<tr><td>Command word</td><td>1 Byte</td><td>0x33</td><td>Command word</td><td>1 Byte</td><td colspan="3">0x33</td></tr>
<tr><td>Data field length</td><td>2 Byte</td><td>0x00 0x00</td><td>Identification word</td><td>1 Byte</td><td colspan="3">0x00 : success
Not 0 : failed</td></tr>
<tr><td rowspan="3">Data field</td><td rowspan="3">0 Byte</td><td rowspan="3">No such item</td><td>Data field length</td><td>2 Byte</td><td colspan="3">When data N=0, no such item</td></tr>
<tr><td rowspan="2">Data field</td><td rowspan="2">N Byte
(When data N=0, no such item)</td><td>Data distinguishing mark</td><td>1 Byte</td><td>0x40: NFC card</td></tr>
<tr><td>Result</td><td>X Byte</td><td>Result</td></tr>
<tr><td>Check word</td><td>1 Byte</td><td></td><td>Check word</td><td>1 Byte</td><td colspan="3"></td></tr>
</table>

For example：

In command mode, the upper computer sends the command to acquire data by polling

（blue--card type,red--data）

PC-->Reader :55 AA 33 00 00 CC

Reader-->PC :55 AA 33 00 00 00 CC No data

Reader-->PC :55 AA 33 00 09 00 40 37 64 39 30 64 61 36 31 DD Swipe card data

7.3. 0x01 Device status query

0x01					
Instructions：Identification word 00 indicates that the device is normal, not 0 is abnormal.					
PC->Reader(Send)			Reader->PC(Receive)		
Project	Byte	Instructions	Project	Byte	Instructions
Packet header	2 Byte	Default: 0x55 0xAA	Packet header	1 Byte	Default: 0x55 0xAA
Command word	1 Byte	0x01	Command word	1 Byte	0x01
Data field length	2 Byte	0x00 0x00	Identification word	1 Byte	0x00 : success； Not 0 : failed
Data field	0 Byte	No such item	Data field length	2 Byte	N
			Data field	N Byte	When data N=0, no such item
Check word	1 Byte		Check word	1 Byte	

For example：
PC-->Reader :55 AA 01 00 00 FE
Reader-->PC :55 AA 01 00 02 00 55 AA 03

7.4. 0x02 Get device ID

0x02					
Instructions：Chapter 6 ID of Setting Instruction					
PC->Reader(Send)			Reader->PC(Receive)		
Project	Byte	Instructions	Project	Byte	Instructions
Packet header	2 Byte	Default: 0x55 0xAA	Packet header	1 Byte	Default: 0x55 0xAA
Command word	1 Byte	0x02	Command word	1 Byte	0x02

Data field length	2 Byte	0x00 0x00	Identification word	1 Byte	0x00 : success； Not 0 : failed
Data field	0 Byte	No such item	Data field length	2 Byte	N
			Data field	N Byte	N>0 device ID, with the lower order first
Check word	1 Byte		Check word	1 Byte	

For example：
PC-->Reader :55 AA 02 00 00 FD
Reader-->PC :55 AA 02 00 04 00 80 00 00 00 79
The red part represents the device ID, and the low bit comes first. 80000000 represents the device ID is 128.

7.5. 0x04 LED and buzzer control

0x04					
Instructions： Confirm that the device has lights of corresponding colors.					
PC->Reader(Send)			Reader->PC(Receive)		
Project	Byte	Instructions	Project	Byte	Instructions
Packet header	2 Byte	Default: 0x55 0xAA	Packet header	1 Byte	Default: 0x55 0xAA
Command word	1 Byte	0x04	Command word	1 Byte	0x04
Data field length	2 Byte	0x05 0x00	Identification word	1 Byte	0x00 : success Not 0 : failed
Data field	5 Byte	1 Byte: Switch: 0 off, 1 enable bit0： retain bit1： red light control bit bit2： green light control bit bit3： buzzer control bit bit4： blue light control bit	Data field length	2 Byte	N
		1 Byte: Times	Data field	N Byte	When data N=0, no such item
		1 Byte: Duration (unit: 50MS)			
		1 Byte: Interval time (unit: 50MS)			
		1 Byte: Retain			
Check word	1 Byte		Check word	1 Byte	

For example： Each flash 0x50*50ms（decimal 80） interval 0x0A*50 ms（decimal 10）

55 AA 04 05 00 02 03 50 0A 00 A5 Control the red light to flash three times for 4 seconds with an interval of 0.5s
55 AA 04 05 00 08 03 50 0A 00 AF The buzzer sounds three times for 4 seconds with an interval of 0.5s
55 AA 04 05 00 04 03 50 0A 00 A3 Control the green light to flash three times for 4 seconds with an interval of 0.5s
55 AA 04 05 00 0A 03 50 0A 00 AD The red light and buzzer act for 4 seconds with an interval of 0.5s
55 AA 04 05 00 0C 03 50 0A 00 AB The green light and buzzer act for 4 seconds with an interval of 0.5s
55 AA 04 05 00 06 03 50 0A 00 A1 Red and green flash three times for 4 seconds with an interval of 0.5s
55 AA 04 05 00 0E 03 50 0A 00 A9 The red, green light and buzzer act three times for 4 seconds with an interval of 0.5s
55 AA 04 05 00 18 03 50 0A 00 BF The blue light and buzzer act three times for 4 seconds with an interval of 0.5s

7.6. NFC module operation

NFC module can support Mifare 1 card block reading and writing, and CPU card sending APDU commands. See specific commands for details.
Glossary：
Task start flag bit--This flag bit is used to tell the code scanner when to start the card operation and when to end the card operation, or to tell the code scanner that the instructions for operating the card are independent and there is no dependency between instructions.
This flag is used to set the operating environment of the card. There are three values of this flag:
0x00->AUTO Inform the code scanner that the instruction can be executed independently without dependency between instructions.
0x01->START Inform the scanner that the card operation has started or has not ended, and there may be dependencies between instructions.
0x02->FINISH Inform the code scanner that this instruction is the last instruction of the operation card, and restore the card operation environment to the default state.
If the instruction of the operation card is independent, such as reading and writing a piece of data of M1 card, this flag can be set to AUTO or FINISH.

1. If START is used to start the operation card, FINISH must be used to end the operation; otherwise, the NFC module will work abnormally and can be used again after restart.
2. If more than one card operation instruction is involved in the card operation process, the task start flag of the instruction issued in the process is START, and the last instruction is FINISH.

7.6.1. 0X53CARD NUMBER REPORTING SWITCH

<table>
<tr><td colspan="2">0x53</td><td colspan="4"></td></tr>
<tr><td colspan="6">Instructions： When the value of data field is set to 0x01 or 0x00 (i.e. entering or exiting command mode), all operations are null and the scanner replies success. It is designed to be compatible with v2.10 communication protocol
Note: By default, the card number reporting function is enabled. If the card number reporting function is disabled, the card number will not be obtained in any mode. At this time, the code scanner is mostly used to directly read and write M1 cards or operate CPU cards without obtaining the card number.</td></tr>
<tr><td colspan="3">PC->Reader(Send)</td><td colspan="3">Reader->PC(Receive)</td></tr>
<tr><td>Project</td><td>Byte</td><td>Instructions</td><td>Project</td><td>Byte</td><td>Instructions</td></tr>
<tr><td>Packet header</td><td>2 Byte</td><td>Default: 0x55 0xAA</td><td>Packet header</td><td>1 Byte</td><td>Default: 0x55 0xAA</td></tr>
<tr><td>Command word</td><td>1 Byte</td><td>0x53</td><td>Command word</td><td>1 Byte</td><td>0x53</td></tr>
<tr><td>Data field length</td><td>2 Byte</td><td>0x01 0x00</td><td>Identification word</td><td>1 Byte</td><td>0x00 : success
Not 0 : failed</td></tr>
<tr><td rowspan="2">Data field</td><td rowspan="2">1 Byte</td><td rowspan="2">0x01 ： Module enters command mode
0x00 ： Module exits command mode
0x02：Report by swiping card
0x03： Close reporting</td><td>Data field length</td><td>2 Byte</td><td>N</td></tr>
<tr><td>Data field</td><td>N Byte</td><td>When data N=0, no such item</td></tr>
<tr><td>Check word</td><td>1 Byte</td><td></td><td>Check word</td><td>1 Byte</td><td></td></tr>
</table>

For example：

PC-->Reader :55 AA 53 01 00 02 AF Enable reporting card number

PC-->Reader :55 AA 53 01 00 03 AE Close reporting card number

Reader-->PC :55 AA 53 00 00 00 AC

7.6.2. READ A PIECE OF DATA FROM M1 CARD

<table>
<tr><td colspan="5">0x51</td><td colspan="3">Read a piece of data from M1 card</td></tr>
<tr><td colspan="8">Instructions: The task start flag field is optional. When the instruction does not contain this flag bit, it is executed according to the AUTO flag by default</td></tr>
<tr><td colspan="5">PC->Reader(Send)</td><td colspan="3">Reader->PC(Receive)</td></tr>
<tr><td>Project</td><td>Byte</td><td colspan="3">Instructions</td><td>Project</td><td>Byte</td><td>Instructions</td></tr>
<tr><td>Packet header</td><td>2 Byte</td><td colspan="3">Default: 0x55 0xAA</td><td>Packet header</td><td>1 Byte</td><td>Default: 0x55 0xAA</td></tr>
<tr><td>Command word</td><td>1 Byte</td><td colspan="3">0x51</td><td>Command word</td><td>1 Byte</td><td>0x51</td></tr>
<tr><td>Data field length</td><td>2 Byte</td><td colspan="3">N</td><td>Identification word</td><td>1 Byte</td><td>0x00 : success
Not 0 : failed</td></tr>
<tr><td rowspan="4">Data field</td><td rowspan="4">N Byte</td><td>Key type</td><td>1 Byte</td><td>0x60 -> KEY A
0x61 -> KEY B</td><td rowspan="2">Data field length</td><td rowspan="2">2 Byte</td><td rowspan="2">N</td></tr>
<tr><td>Block number</td><td>1 Byte</td><td>0 ~ 0xFF</td></tr>
<tr><td>Secret key</td><td>6 Byte</td><td></td><td rowspan="2">Data field</td><td rowspan="2">N Byte</td><td rowspan="2">When data N=0, no such item</td></tr>
<tr><td>Task start flag (optional)</td><td>1 Byte</td><td>0x00 -> AUTO
0x01 -> START
0x02 -> FINISH</td></tr>
<tr><td>Check word</td><td>1 Byte</td><td colspan="3"></td><td>Check word</td><td>1 Byte</td><td></td></tr>
</table>

For example:
Use the A (0x60) key for authentication, read the second block of sector 6 (that is, the absolute block number is 0x19)
Authentication key is FF FF FF FF FF FF, flag bit is optional.
PC-->Reader :55 AA 51 09 00 60 19 FF FF FF FF FF FF 00 DE Include flag bit
PC-->Reader :55 AA 51 08 00 60 19 FF FF FF FF FF FF DF No flag bit
Reader-->PC :55 AA 51 00 10 00 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 Card reading succeeded
Reader-->PC :55 AA 51 FF 00 00 51 Failed or no card

7.6.3. WRITE A PIECE OF DATA FROM M1 CARD

<table>
<tr><td colspan="5">0x52</td><td colspan="3">Write a piece of data from M1 card</td></tr>
<tr><td colspan="8">Instructions: The task start flag field is optional. When the instruction does not contain this flag bit, it is executed according to the AUTO flag by default</td></tr>
<tr><td colspan="5">PC->Reader(Send)</td><td colspan="3">Reader->PC(Receive)</td></tr>
<tr><td>Project</td><td>Byte</td><td colspan="3">Instructions</td><td>Project</td><td>Byte</td><td>Instructions</td></tr>
<tr><td>Packet header</td><td>2 Byte</td><td colspan="3">Default: 0x55 0xAA</td><td>Packet header</td><td>1 Byte</td><td>Default: 0x55 0xAA</td></tr>
<tr><td>Command word</td><td>1 Byte</td><td colspan="3">0x52</td><td>Command word</td><td>1 Byte</td><td>0x52</td></tr>
<tr><td>Data field length</td><td>2 Byte</td><td colspan="3">N</td><td>Identification word</td><td>1 Byte</td><td>0x00 : success
Not 0 : failed</td></tr>
<tr><td rowspan="5">Data field</td><td rowspan="5">N Byte</td><td>Key type</td><td>1 Byte</td><td>0x60 -> KEY A
0x61 -> KEY B</td><td rowspan="2">Data field length</td><td rowspan="2">2 Byte</td><td rowspan="2">N</td></tr>
<tr><td>Block number</td><td>1 Byte</td><td>0 ~ 0xFF</td></tr>
<tr><td>Secret key</td><td>6 Byte</td><td></td><td rowspan="3">Data field</td><td rowspan="3">N Byte</td><td rowspan="3">When data N=0, no such item</td></tr>
<tr><td>Data</td><td>16 Byte</td><td></td></tr>
<tr><td>Task start flag (optional)</td><td>1 Byte</td><td>0x00 -> AUTO
0x01 -> START
0x02 -> FINISH</td></tr>
<tr><td>Check word</td><td>1 Byte</td><td colspan="3"></td><td>Check word</td><td>1 Byte</td><td></td></tr>
</table>

For example:
Use the B (0x61) key for authentication, and write data to the second block of sector 6 (that is, the absolute block number is 0x19)
Authentication key is FF FF FF FF FF FF, flag bit is optional.
PC-->Reader :55 AA 52 19 00 61 19 FF FF FF FF FF FF 11 11 11 11 11 11 11 11 22 22 22 22 22 22 22 22 00 CC Include flag bit
PC-->Reader :55 AA 52 18 00 61 19 FF FF FF FF FF FF 12 34 56 78 90 12 34 56 12 34 56 78 90 12 34 56 CDNo flag bit
Reader-->PC :55 AA 52 00 00 00 AD Card writing succeeded
Reader-->PC :55 AA 52 FF 00 00 52 Failed or no card

7.6.4. READ MULTIPLE BLOCKS IN M1 CARD SECTOR

0xA0	Read multiple blocks in M1 card sector

Instructions: Can read S50/S70 cards，values of sector number, offset, number of blocks depend on the card type.
Offset--Calculate the base address of the block to be read with the selected sector 0 block as the starting address.
Number of blocks--Take the selected base address block as the card reading start block, and continuously read the selected number of blocks.
Command Analyzing: Read 2 sectors, 1 piece and 2 pieces of data of a card.
55 AA A0 0B 00 00 60 02 01 02 FF FF FF FF FF FF 35

55 AA	A0	0B 00	00	60	02	01	02	FF ~FF	35
Command header	Instructions	Data length	AUTO	Key type	Block number	Base address of block to be read	Read several blocks continuously from the base address	Secret key	Check word

Note: the number of read blocks cannot be 0. If it is 0, it will be regarded as invalid instruction. Block data cannot be read across sectors in one instruction.

PC->Reader(Send)					Reader->PC(Receive)		
Project	Byte	Instructions			Project	Byte	Instructions
Packet header	2 Byte	Default: 0x55 0xAA			Packet header	1 Byte	Default: 0x55 0xAA
Command word	1 Byte	0xA0			Command word	1 Byte	0x51
Data field length	2 Byte	N			Identification word	1 Byte	0x00 : success Not 0 : failed
Data field	11 Byte	Task flag bit	1 Byte	0x00 -> AUTO 0x01 -> START 0x02 -> FINISH	Data field length	2 Byte	N
		Key type	1 Byte	0x60 -> KEY A 0x61 -> KEY B			
		Sector number	1 Byte	S50 -> 0x00~0x0F S70 -> 0x00~0x27			
		Offset	1 Byte	S50 -> 0x00~0x03 S70 -> 0x00~0x03 or 0x00~0x0F	Data field	N Byte	When data N=0, no such item
		Block number	1Byte	S50 -> 0x01~0x04 S70 -> 0x01~0x04 or 0x01~0x10			
		Secret key	6 Byte				
Check word	1 Byte				Check word	1 Byte	

For example:
Authenticate with A(0x60) key, read 2 sectors of 0 blocks, 1 block, 2 blocks of data, that is, read 3 blocks in succession with 0 blocks as base address.
The authentication key is FF FF FF FF FF FF, and the flag bit is set to AUTO.
PC-->Reader :55 AA A0 0B 00 00 60 02 00 03 FF FF FF FF FF FF 35
Reader-->PC :55 AA A0 00 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6F Data reading succeed
Reader-->PC :55 AA A0 FF 00 00 A0 Failed or no card

7.6.5. WRITE MULTIPLE BLOCKS IN M1 CARD SECTOR

0xA1	Write multiple blocks in M1 card sector

Instructions: Can read S50/S70 cards， sector number, offset, number of blocks depend on card type.
Offset--Calculates the base address of the block to be written from the selected sector 0 block as the starting address.
Number of blocks--Continuously write data to the selected number of blocks starting with the selected base address block.
Command analyzing:
Write data to 2 sectors, 1 block and 2 blocks of a card(See examples for instructions)
55 AA A1 2B 00 00 60 02 01 02 FF FF FF FF FF FF 36

55 AA	A1	2B 00	00	60	02	01	02	FF ~FF	... ...	36
Command header	Instructions	Data length	AUTO	Key type	Sector number	Base address of block to be written	Write several blocks continuously from the base address	Secret key	Data to be written	Check word

Note： The number of blocks to be written cannot be 0, If it is 0, it will be regarded as invalid instruction. Data cannot be written across sectors in one instruction.

PC->Reader(Send)					Reader->PC(Receive)		
Project	Byte	Instructions			Project	Byte	Instructions
Packet header	2 Byte	Default: 0x55 0xAA			Packet header	1 Byte	Default: 0x55 0xAA
Command word	1 Byte	0xA1			Command word	1 Byte	0xA1
Data field length	2 Byte	N			Identification word	1 Byte	0x00:success Not 0:failed
Data field	N Byte	Task flag bit	1 Byte	0x00 -> AUTO 0x01 -> START 0X02 -> FINISH	Data field length	2 Byte	N
		Key type	1 Byte	0x60 -> KEY A 0x61 -> KEY B			
		Sector number	1 Byte	S50 -> 0x00~0x0F S70 -> 0x00~0x27			
		Offset	1 Byte	S50 -> 0x00~0x03 S70 -> 0x00~0x03 or 0x00~0x0F	Data field	0 Byte	When data N=0, no such item
		Block number	1Byte	S50 -> 0x01~0x04 S70 -> 0x01~0x04 or 0x01~0x10			
		Secret key	6 Byte				
		Data	N Byte	N = 16 * block numbers			
Check word	1 Byte				Check word	1 Byte	

For example：
Authenticate with A(0x60) key, write data to 2 sectors 1 block, 2 blocks, that is, write 2 blocks in succession with 1 block as base address.
The authentication key is FF FF FF FF FF FF, and the flag bit is set to AUTO.
PC-->Reader :55 AA A1 2B 00 00 60 02 01 02 FF FF FF FF FF FF 11 11 11 11 11 11 11 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 33 33 33 33 33 33 33 33 33 36
For example2：
55 AA A1 2B 00 00 60 02 01 02 FF FF FF FF FF FF 44 44 44 44 44 44 44 44 44 55 55 55 55 55 55 55 55 55 55 55 55 55 55 66 66 66 66 66 66 66 66 66 36
Reader-->PC : 55 AA A1 00 00 00 5E Data writing succeed
Reader-->PC :55 AA A1 FF 00 00 A1 Failed or no card

7.6.6. 0XA6 SEND APDU INSTRUCTION

<table>
<tr><td colspan="3">0xA6</td><td colspan="4"></td></tr>
<tr><td colspan="7">Instructions: For communication with CPU cards, APDU instruction can be found in FMCOS2.0 user manual.</td></tr>
<tr><td colspan="4">PC->Reader(Send)</td><td colspan="3">Reader->PC(Receive)</td></tr>
<tr><td>Project</td><td>Byte</td><td colspan="3">Instructions</td><td>Project</td><td>Byte</td><td>Instructions</td></tr>
<tr><td>Packet header</td><td>2 Byte</td><td colspan="3">Default: 0x55 0xAA</td><td>Packet header</td><td>1 Byte</td><td>Default: 0x55 0xAA</td></tr>
<tr><td>Command word</td><td>1 Byte</td><td colspan="3">0xA6</td><td>Command word</td><td>1 Byte</td><td>0xA6</td></tr>
<tr><td>Data field length</td><td>2 Byte</td><td colspan="3">N</td><td>Identification word</td><td>1 Byte</td><td>0x00 : success
Not 0 : failed</td></tr>
<tr><td rowspan="2">Data field</td><td rowspan="2">N Byte</td><td>Task flag bit</td><td>1 Byte</td><td>0x01 -> START
0X02 -> FINISH</td><td rowspan="2">Data field length</td><td rowspan="2">2 Byte</td><td rowspan="2">N</td></tr>
<tr><td>APDU DATA</td><td>N Byte</td><td>Data structure conforming to ISO7816-4</td></tr>
<tr><td>Check word</td><td>1 Byte</td><td colspan="3"></td><td>Data field</td><td>N Byte</td><td>When data N=0, no such item</td></tr>
<tr><td>Check word</td><td>1 Byte</td><td colspan="3"></td><td>Check word</td><td>1 Byte</td><td></td></tr>
</table>

For example：The red part is APDU instruction
Select application catalogue：
PC-->Reader :55 AA A6 08 00 01 00 A4 00 00 02 3F 01 C8
Reader-->PC : 55 AA A6 00 11 00 6F 0D 84 05 41 44 46 30 31 A5 04 9F 08 01 02 90 00 4C

Get 4-bit random number:
Reader-->PC : 55 AA A6 06 00 01 00 84 00 00 04 DE
Reader-->PC :55 AA A6 00 06 00 7C C9 56 38 90 00 14

External authentication: four digit random number is used for external authentication.
The authentication method is DES single length, and the default key is (112233445667788)
PC-->Reader :55 AA A6 0E 00 01 00 82 00 00 08 71 7E B1 7D 4C F6 81 17 33
Reader-->PC : 55 AA A6 00 02 00 90 00 CB

Select binary file:
PC-->Reader :55 AA A6 06 00 02 00 B0 83 00 00 6E
Reader-->PC : 55 AA A6 00 12 00 11 22 33 44 55 66 77 88 00 00 00 00 00 00 00 00 90 00 53