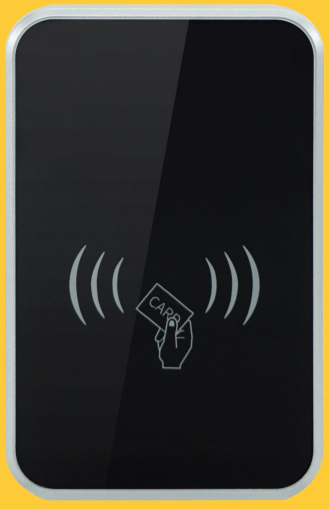
微光互联
二维码扫描专家

SK330

User manual

Please read it carefully and keep it properly.



- [x] Fast recognition.
- [x] Three working modes.
- [x] Support reading and writing M1 card sector.

Beijing Vguang Internet Technology Co., Ltd.

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.

Edit history

Change date	Version	Description	Responsible
2021.3.18	V1.0	Initial version	

Catalog

1. Preface

Thanks for using the SK330 QR code 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

SK300 card swiping device is a product designed for card swiping scenarios. It supports Wiegand, RS232 and RS485 interfaces.

1.2. Product features

1, Support Wiegand 26 and 34 switching.

2, Support Wiegand, RS232 and RS485 interfaces.

3, Support reading and writing M1 card sector.

4, Baud rate 115200.

1.3. Working mode

1, Normal operating mode

The card swiping is directly output. When the card swiping is successful, the buzzer acts and the green LED light is on.

2, Wire control mode

In this mode, when swiping the card, the buzzer will not sound and the green LED light will not be on. Refer to the interface definition and control the buzzer and green LED light action by pulling down and

up the interface level. (Only the Wiegand version of the card reader is supported.)

Setting method of wire control mode:

First pull down the model setting pin (it can be ground mode). After the card reader is powered on, it is the wire control mode.

3, Protocol mode

After swiping the card, the data is output to the upper computer. The upper computer can do corresponding actions through the protocol control equipment, which is only supported by the 485/232 card reader. In this mode, you can also send protocol instructions to read and write sectors.

4, Wiegand protocol switching

Wiegand 26: WG26/34_SEL is suspended and the card reader is powered on, which is the Wiegand 26 protocol.

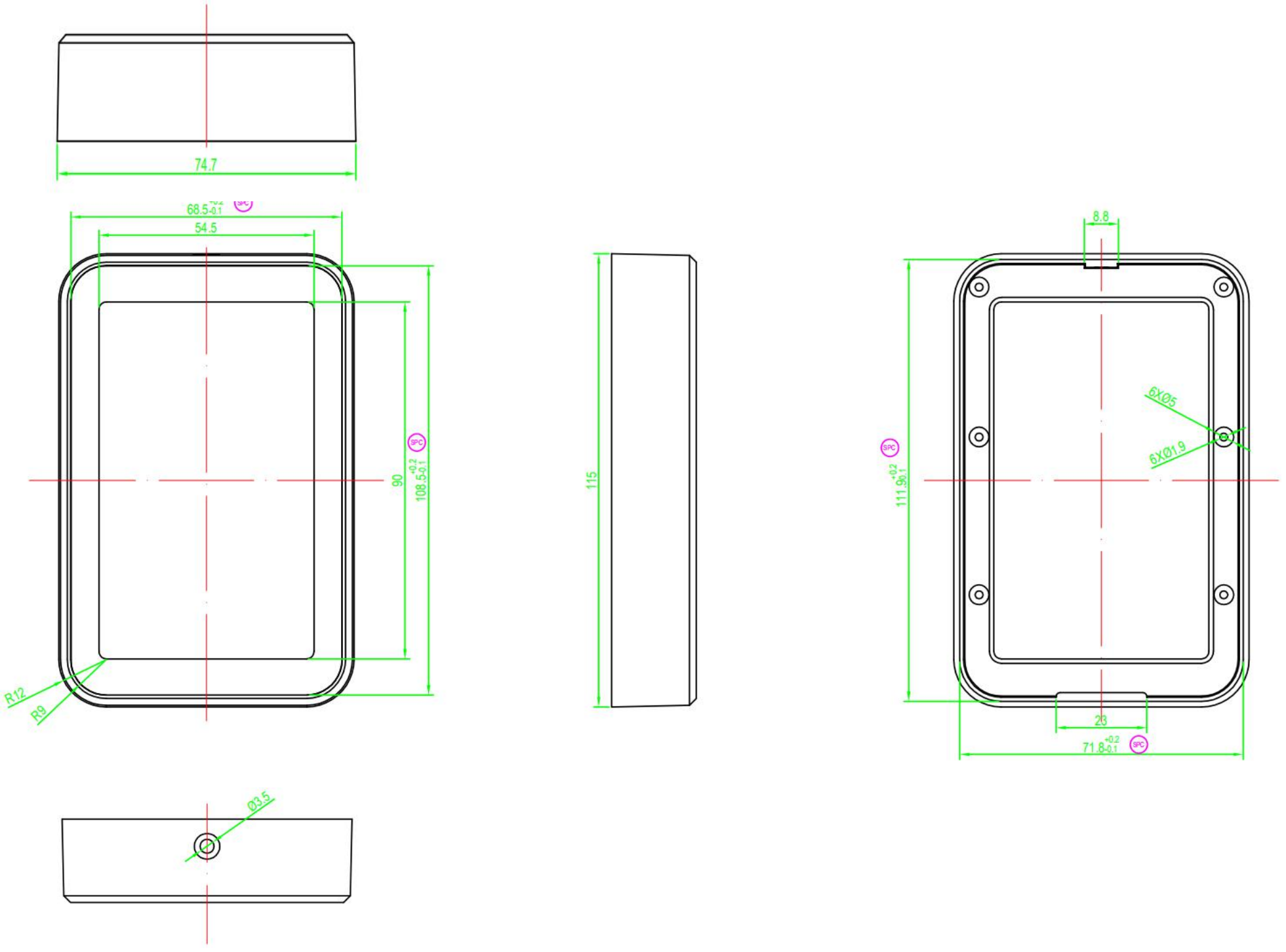
Wiegand 34：WG26/34_SEL is ground mode and the card reader is powered on, which is the Wiegand 34 protocol.

2. Product appearance

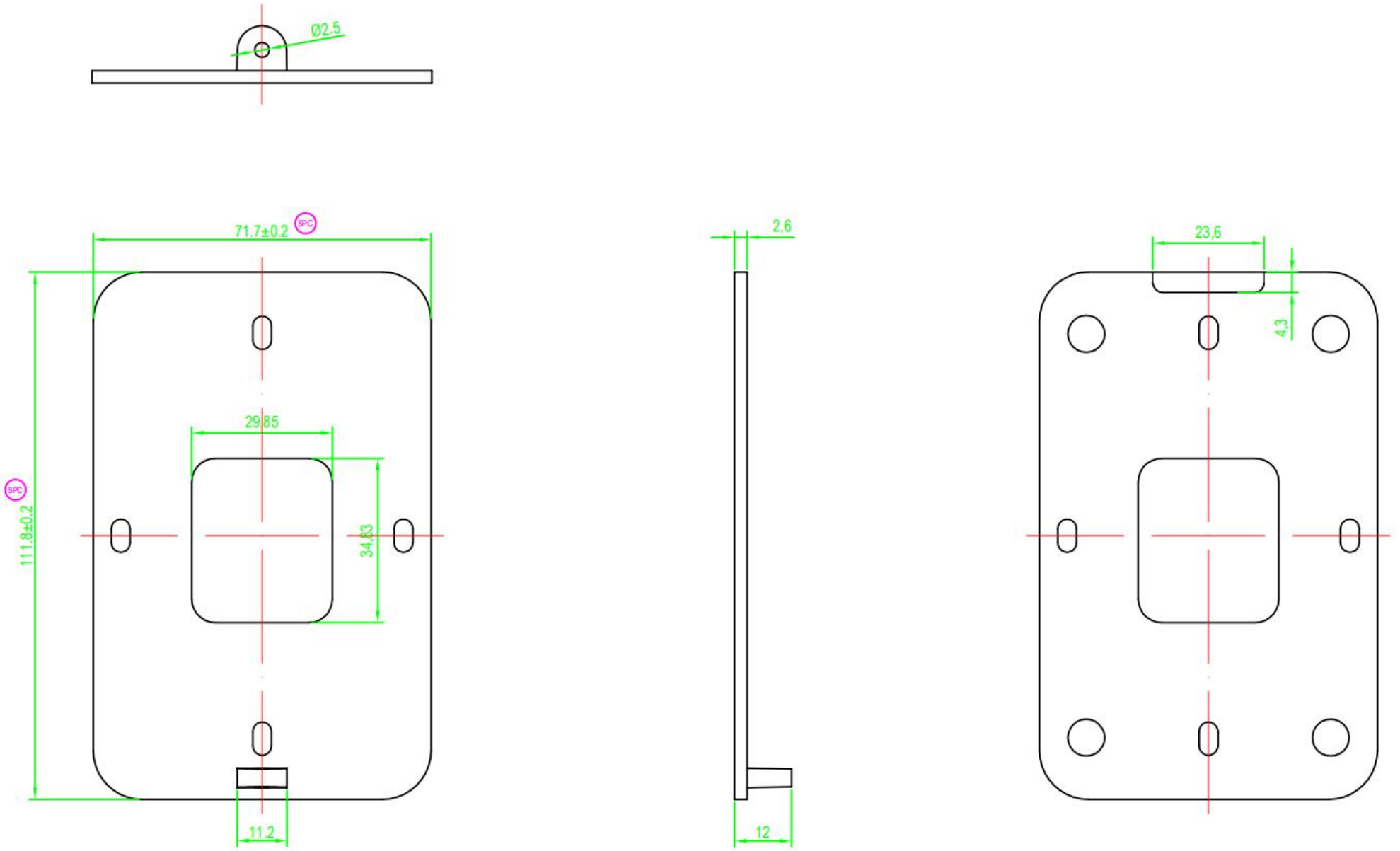
2.1. Appearance picture

2.2. Product size chart

2.2.1. PRODUCT COVER DIAGRAM



2.2.2. PRODUCT LOWER COVER DRAWING



3.Product parameters

General parameters	
Support interface	RS232, RS485, Wiegand
Working voltage	DC 9V-24V
Working current	80mA （Typical 12V power supply）
Indication mode	Red light, green light, buzzer
Installation Method	Wall mounted installation
Product size	109.3mm*75mm*21mm
Supported card type	S70\CPU\FM1208\FM1216-137\MIFARE CLASSIC EV1 4K(S70)\mifare desfire ev2 d42\UL\FM12081K+7K\NTAG216\S50\ultralight c\UL EV1\DESFIRE EV2 D41\Ultralight EV1\ISO14443B CPU\ID card
Operation card mode	Read the physical card number, read and write the sector, and operate the CPU card
Card swiping distance	≤5cm （It is related to the card type, specification and usage scenario）
Antenna specification	13.56MHz

4. Interface definition

Pin#	Definition	Instruction
1	LEN_IN	LED light control pin, low level effective
2	BEEP_IN	Buzzer control pin, low level effective
3	GND	Power supply negative

4	VCC	Power supply positive
5	WG_D0	Wiegand D0
6	WG_D1	Wiegand D1
7	WG26/34_SEL	Wiegand protocol setting pin Hanging: Wiegand 26 Low level: Wiegand 34
8	WG_MODEL	Wiegand wire mode setting pin: low level effective
9	VCC	Power supply positive
10	GND	Power supply negative
11	232_RXD/485B	232 receiving pin / RS485 B line
12	232_TXD/485A	232 receiving pin / RS485 A line

5.Protocol instruction

5.1. Data transmission protocol

5.1.1. 5.1.1.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 by byte XOR value from the command header to the last byte of the data field

5.1.2. 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

Task start flag bit: AUTO/START/Finish marks all task flag bits are not valid at present and can be omitted.

Note: 0x24、0x25、0x53（Set output mode）command will save the configuration and restart after power failure will not be lost.

5.1.3. 0X04 LED AND BUZZER CONTROL

<table>
<tr><td colspan="3">Instruction: 0x04</td><td colspan="3"></td></tr>
<tr><td colspan="6">Note: only one action is supported, and the delay time is the same as that during card swiping.</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>Instruction</td><td>Project</td><td>Byte</td><td>Instruction</td></tr>
<tr><td>Packet headers</td><td>2Byte</td><td>Default: 0x55 0xaa</td><td>Packet headers</td><td>1Byte</td><td>Default: 0x55 0xaa</td></tr>
<tr><td>Command word</td><td>1Byte</td><td>0x04</td><td>Command word</td><td>1Byte</td><td>0x04</td></tr>
<tr><td>Data field length</td><td>2Byte</td><td>0x05 0x00</td><td>Identification word</td><td>1Byte</td><td>0x00 : Success
Not 0: Failed</td></tr>
<tr><td rowspan="2">Data field</td><td rowspan="2">1Byte</td><td rowspan="2">Switch: 0 off, 1 enabled
bit0: Reserved
bit1 : Red light control bit（invalid）
bit2: Green light control bit
bit3: Buzzer control bit</td><td>Data field length</td><td>2Byte</td><td>N</td></tr>
<tr><td>Data field</td><td>N Byte</td><td>Data: not available when n = 0</td></tr>
<tr><td>Check word</td><td>1Byte</td><td></td><td>Check word</td><td>1Byte</td><td></td></tr>
</table>

For example:

Control the buzzer and light to act once: 55 AA 04 01 00 0C F6 （Green light+buzzer）

Control the buzzer to act once: 55 AA 04 01 00 08 F2

Control the buzzer and light to act once: 55 AA 04 01 00 04 FE

Device return: 55 AA 04 00 00 00 FB

5.1.4. 0X24 LED CONFIGURATION

Instruction: 0x24					
Instruction: only the green light is controllable					
PC->Reader(Send)			Reader->PC(Receive)		
Project	Byte	Instruction	Project	Byte	Instruction
Packet headers	2Byte	Default: 0x55 0xaa	Packet headers	1Byte	Default: 0x55 0xaa
Command word	1Byte	0x24	Command word	1Byte	0x24
Data field length	2Byte	0x01 0x00	Identification word	1Byte	0x00 : Success Not 0: Failed
Data field	1 Byte	bit0: White light control bit (invalid) bit1: Red light control bit (invalid) bit2: Green light control bit	Data field length	2Byte	N
			Data field	N Byte	Data: not available when n = 0
Check word	1Byte		Check word	1Byte	

For example:

Card swiping and turn on the light: 55 AA 24 01 00 04 DE

Card swiping does not turn on the light: 55 AA 24 01 00 00 DA

Device response: 55 AA 24 00 00 00 DB

5.1.5. 0X25 BUZZER RESPONSE CONFIGURATION

Instruction: 0x25					
Instruction:					
PC->Reader(Send)			Reader->PC(Receive)		
Project	Byte	Instruction	Project	Byte	Instruction
Packet headers	2Byte	Default: 0x55 0xaa	Packet headers	1Byte	Default: 0x55 0xaa
Command word	1Byte	0x25	Command word	1Byte	0x25
Data field length	2Byte	0x01 0x00	Identification word	1Byte	0x00 : Success Not 0: Failed

Data field	1 Byte	0: Buzzer off 1: Buzzer on	Data field length	2Byte	N
			Data field	N Byte	Data: not available when n = 0
Check word	1Byte		Check word	1Byte	

For example:

Card swiping buzzer: 55 AA 25 01 00 01 DA

Card swiping does not ring buzzer: 55 AA 25 01 00 00 DB

Device response: 55 AA 25 00 00 00 DA

5.1.6. CARD NUMBER REPORTING SWITCH

Instruction: 0x53					
Instruction:					
PC->Reader(Send)			Reader->PC(Receive)		
Project	Byte	Instruction	Project	Byte	Instruction
Packet headers	2Byte	Default: 0x55 0xaa	Packet headers	1Byte	Default: 0x55 0xaa
Command word	1Byte	0x53	Command word	1Byte	0x53
Data field length	2 Byte		Identification word	1Byte	0x00: Success 0xFF: Failed Other values: Failed
Data field	1 Byte	0x00: Do not report card number 0x01: Do not report card number 0x02: Serial port common output card number 0x03: Serial protocol output card number	Data field length	2 Byte	
			Data field	N Byte	
Check word	1 Byte		Check word	1 Byte	

For example:

55 aa 53 01 00 00 AD Do not report card number

55 aa 53 01 00 01 AC Do not report card number

55 aa 53 01 00 02 AF Serial port common output card number

55 aa 53 01 00 03 AE Serial protocol output card number

5.1.7. 5.1.7.READ A PIECE OF DATA FROM M1 CARD

<table>
<tr><td colspan="3">Instruction: 0x51</td><td colspan="3">Read a piece of data from M1 card</td></tr>
<tr><td colspan="6">Instruction: The task start flag field is optional. When the flag bit is not included in instructions, it is executed according to the auto ID by default</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 colspan="3">Instruction</td><td>Project</td><td>Byte</td><td>Instruction</td></tr>
<tr><td>Packet headers</td><td>2Byte</td><td colspan="3">Default: 0x55 0xAA</td><td>Packet headers</td><td>1Byte</td><td>Default: 0x55 0xAA</td></tr>
<tr><td>Command word</td><td>1Byte</td><td colspan="3">0x51</td><td>Command word</td><td>1Byte</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>1Byte</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>1Byte</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">Data: not available when n = 0</td></tr>
<tr><td>Task start flag bit (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>1Byte</td><td></td></tr>
</table>

For example:

Use a (0x60) key for authentication and read the data of the second block of sector 6 (the absolute block number is 0x19).The authentication key is FF FF FF FF FF FF, and the 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 success

Reader-->PC :55 AA 51 FF 00 00 51 Failed or no card

5.1.8. WRITE A PIECE OF DATA FROM M1 CARD

<table>
<tr><td colspan="3">Instruction: 0x52</td><td colspan="4">Write data to a block of M1 card</td></tr>
<tr><td colspan="7">Instruction:The task start flag field is optional. When the flag bit is not included in instructions, it is executed according to the auto ID by default.</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">Instruction</td><td>Project</td><td>Byte</td><td>Instruction</td></tr>
<tr><td>Packet headers</td><td>2 Byte</td><td colspan="3">Default: 0x55 0xAA</td><td>Packet headers</td><td>1Byte</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>1Byte</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>1Byte</td><td>0x00 : Success
Not 0 : Failed</td></tr>
<tr><td rowspan="6">Data field</td><td rowspan="6">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">2Byte</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="4">Data field</td><td rowspan="4">NByte</td><td rowspan="4">Data:not available when n = 0</td></tr>
<tr><td>Data</td><td>16 Byte</td><td></td></tr>
<tr><td>Task flag bit(optional)</td><td>1 Byte</td><td>0x00 -> AUTO
0x01 -> START
0X02 -> FINISH</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>Check word</td><td>1 Byte</td><td colspan="3"></td><td>Check word</td><td>1Byte</td><td></td></tr>
</table>

For example:

Use the B (0x61) key for authentication and write data to the second block of sector 6 (the absolute block number is 0x19).The authentication key is FF FF FF FF FF FF,and the 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 CD No flag bit

Reader-->PC :55 AA 52 00 00 00 AD Write data success

Reader-->PC :55 AA 52 FF 00 00 52 Failed or no card

5.1.9. READ MULTIPLE BLOCKS IN M1 CARD SECTOR

Instruction: 0xA0	Read multiple blocks in M1 card sector

Instruction: 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	Instruction			Project	Byte	Instruction
Packet headers	2Byte	Default: 0x55 0xAA			Packet headers	1Byte	Default: 0x55 0xAA
Command word	1Byte	0xA0			Command word	1Byte	0x51
Data field length	2Byte	N			Identification word	1Byte	0x00 : Success Not 0 : Failed
Data field	11 Byte	Task flag bit	1 Byte	0x00 -> AUTO 0x01 -> START 0x02 -> FINISH	Data field length	2Byte	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	NByte	Data:not available when n = 0
		Number of blocks	1Byte	S50 -> 0x01~0x04 S70 -> 0x01~0x04 or 0x01~0x10			
		Secret key	6 Byte				

Check word	1Byte		Check word	1Byte	

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 6F Read data success

Reader-->PC :55 AA A0 FF 00 00 A0 Failed or no card

5.1.10. WRITE MULTIPLE BLOCKS IN M1 CARD SECTOR

Instruction: 0xA1	Write multiple blocks of data

Instruction: 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.

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	Read 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	Instruction			Project	Byte	Instruction
Packet headers	2 Byte	Default: 0x55 0xAA			Packet headers	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	Data:not available when n = 0

		Number of blocks	1Byte	S50 -> 0x01~0x04 S70 -> 0x01~0x04 or 0x01~0x10			
		Secret key	6 Byte				
		Data	N Byte	N = 16 * Number of blocks			
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 11 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

Reader-->PC : 55 AA A1 00 00 00 5E Write data success

Reader-->PC :55 AA A1 FF 00 00 A1 Failed or no card

5.1.11. 0XA6 SEND APDU INSTRUCTION

<table>
<tr><td colspan="3">Instruction: 0xA6</td><td colspan="6"></td></tr>
<tr><td colspan="9">Instruction: 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="5">Reader->PC(Receive)</td></tr>
<tr><td>Project</td><td>Byte</td><td colspan="3">Instruction</td><td>Project</td><td>Byte</td><td colspan="2">Instruction</td></tr>
<tr><td>Packet headers</td><td>2 Byte</td><td colspan="3">Default: 0x55 0xAA</td><td>Packet headers</td><td>1 Byte</td><td colspan="2">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 colspan="2">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 colspan="2">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 colspan="2" 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 colspan="2">Data:not available when n = 0</td></tr>
<tr><td>Check word</td><td>1 Byte</td><td colspan="3"></td><td>Check word</td><td>1 Byte</td><td colspan="2"></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

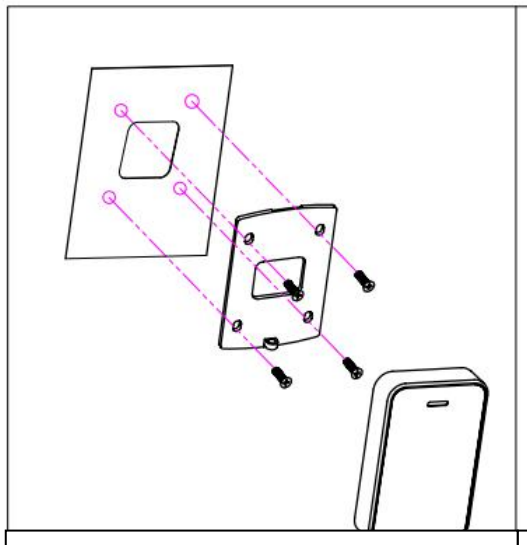
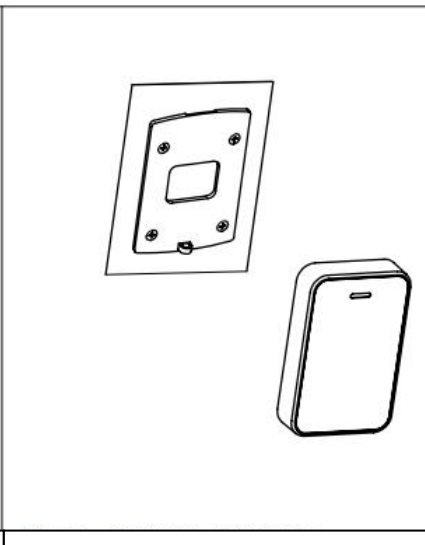
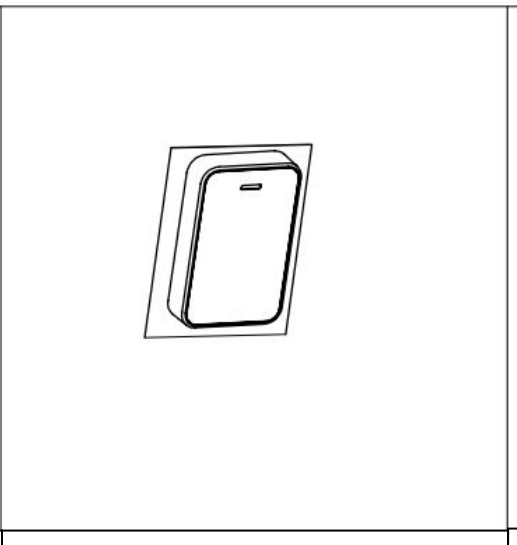
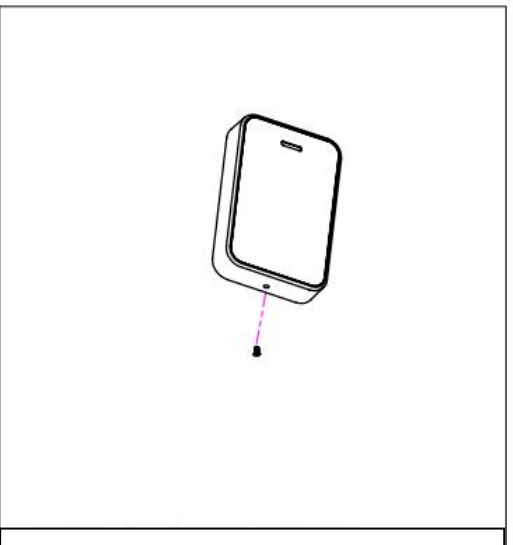
Reader-->PC : 55 AA A6 00 02 00 90 00 CB

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

6. Installation method

			
Step 1:Open screw holes and outlet holes on the holder.	Step 2:Lock the metal support with four M4 cross countersunk head screws.	Step 3:Fix the product to the metal bracket.	Step 4:Lock the product with a M3*6 cross countersunk head screw.

7. Contact

Company name: Beijing Vguang Internet Technology Co., Ltd.

Address: China Meteorological Science and Technology Park, No.2, Zhenxing Road, Changping District, Beijing, China.

Hot line：400-810-2019