**ICM522-C5-RS232**

Contents

[1. Product features 2](#_Toc63258174)

[2. Product introduction 2](#_Toc63258175)

[3. Product photo 2](#_Toc63258176)

[4. Model 2](#_Toc63258177)

[5. Specification 3](#_Toc63258178)

[6. Pin function 3](#_Toc63258179)

[6.1. Terminal: 1.25mm pitch vertical patch header 3](#_Toc63258180)

[7. Power data communication protocol ground 4](#_Toc63258181)

[7.1 UART interface： 4](#_Toc63258182)

[7.2 Send data format： 4](#_Toc63258183)

[7.3 Return data format： 4](#_Toc63258184)

[8.Command list 5](#_Toc63258185)

[8.1 Set the module to enter sleep mode 5](#_Toc63258186)

[8.2 Set antenna and card detection mode 5](#_Toc63258187)

[8.3 Set active output card ID mode 6](#_Toc63258188)

[8.4 Set LED status 7](#_Toc63258189)

[8.5 Set the buzzer on time 7](#_Toc63258190)

[8.6 Set the serial port baud rate 8](#_Toc63258191)

[8.7 ISO14443 TYPE A search card 9](#_Toc63258192)

[8.8 Mifare one card reading block 10](#_Toc63258193)

[8.9 Mifare one card writing block 10](#_Toc63258194)

[8.10 Mifare one card initialization wallet 11](#_Toc63258195)

[8.11 Mifare one card reader wallet 12](#_Toc63258196)

[8.12 Mifare one Card charge value wallet 13](#_Toc63258197)

[8.13 Mifare one card deduction wallet 13](#_Toc63258198)

[【Receive Data:】FE 02 09 0B 14](#_Toc63258199)

[8.14 Mifare one card backup wallet 14](#_Toc63258200)

[8.15 Mifare Ultralight (M0) card page reading 15](#_Toc63258201)

[8.16 Mifare Ultralight (M0) card write page 16](#_Toc63258202)

[8.17 Mifare one/Ultralight card sleep 16](#_Toc63258203)

[8.18 ISO14443 TYPE A CPU card reset 17](#_Toc63258204)

[8.19 ISO14443 TYPE A CPU card Send COS instruction 17](#_Toc63258205)

[9. Size parameters 18](#_Toc63258206)

[10. Appendix 1: Command summary table 19](#_Toc63258207)

# Product features

* Using highly integrated read-write chip as the characteristics of radio frequency base station products
* Support ISO14443 TYPE A standard, Mifare standard card, the module integrates instructions such as automatic card searching, reading, writing, initializing e-wallet, adding value, reducing value, checking balance, etc. The user uses the command set to simply operate the card;
* Support mobile wallet NFC door key
* Integrated antenna design
* Ultra-low static power consumption：≤30uA
* Support serial port protocol (UART RS232): serial port bit data (8), parity bit (none), stop bit (1), baud rate can be set, default: 9600
* Ultra-small DIP，45mm×20mm
* The module software has strong extension functions, and personalized modules can be customized according to user requirements
* **A1190/A2180/A3180-C3566 Port :ttys2,9600 Baud rate**

# Product introduction

The ICM522-C5-RS232 read-write module adopts

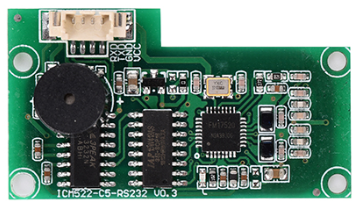
13.56MHZ non-contact radio frequency technology, and an embedded highly integrated read-write chip is used as a radio frequency base station. The user only needs to send commands through simple instructions to achieve complete operation of the card. This series of read-write modules supports

MF1: S50, S70, FM11RF08;

MF0: Ultralight, Ultralight C, NTAG203, NTAG213;

CPU: FM1208 and compatible cards. ICM522-C5-RS232 module has strong software and hardware expansion functions, and personalized modules can be customized according to user requiremen

# Product photo

****

# Model

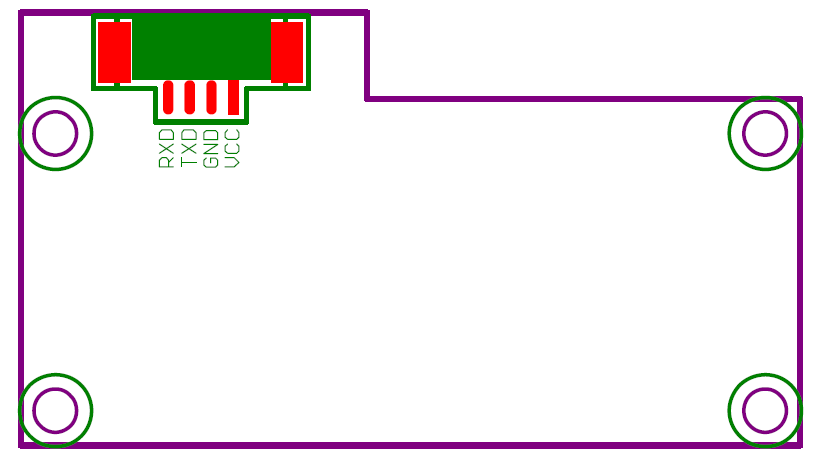
|  |  |  |
| --- | --- | --- |
| **Model** | **Difference** | Support card type |
| ICM522-C5-CPU-RS232 | UART RS232 | MF0：Ultralight、Ultralight C、NTAG203、NTAG213  M F1：S50、S70、FM11RF08  CPU：FM1208  and compatible cards |

# Specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | **Parameter** | | | | **Sstatus** |
| **MINI.** | **TYPE.** | **MAX.** | **Unit** |
| Power |  |  |  |  |  |
| Select power supply voltage 5.0V | 4.7 | 5.0 | 6.0 | V |  |
| Select power supply voltage 3.3V | 3.0 | 3.3 | 3.6 | V |  |
| Peak reading current | - | 35 | 60 | mA |  |
| Average quiescent current | - | - | 30 | uA |  |
| Reading distance | - | 5 | - | CM | Standard card test |
| Frequency |  |  |  |  |  |
| Frequency range | - | 13.56 | - | MHz |  |

# 6. Pin function

## 6.1. Terminal: 1.25mm pitch vertical patch header



|  |  |  |  |
| --- | --- | --- | --- |
| **PIN** | **Signal name** | **Direction** | **Description** |
| J1\_1 | RXD | I/O | RXD serial data input |
| J1\_2 | TXD | I/O | TXD serial data output |
| J1\_3 | GND | G | PWR GND |
| J1\_4 | VCC | Pwr/I | DC3.3V OR 5.0V Power input（select one） |

# 7. Power data communication protocol ground

## 7.1 UART interface：

Start bit：1

Data bit：8

Parity bit：None

Stop bit：1

Baud rate：Default 9600

 Timing diagram：

## 7.2 Send data format：

**Data packet content:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address (2Byte) | Length(1Byte) | Command(1Byte) | Data(nByte) | Check(1Byte) |

Module address**：**It is fixed to 0x0000 for modules used alone；

For a single-pair network version module, it is 0x0001~0xFFFE;

0xFFFF is broadcast。

**Length：**length(1Byte) + Command(1Byte) + Data(nByte)

Command: The meaning of this command

Data:valid data

Check: length (1Byte) + command (1Byte) + data (nByte) XOR and

## 7.3 Return data format：

**Data packet content:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header (1Byte) | Length(1Byte) | Success/failure value (1Byte) | Data(nByte) | Check(1Byte) |

Command header: 0xFE

Length: length (1Byte) + command (1Byte) + data (nByte)

Success/Failure value: return command execution success; 0xE0---0xFF execution error

Data: if there is data return, it is valid data

Check: length (1Byte) + command (1Byte) + data (nByte) XOR and

# 8.Command list

## 8.1 Set the module to enter sleep mode

Function description: Used to set the module to enter sleep mode, reduce power consumption ≤30uA, and send other commands than this command during sleep to wake up.

**Send data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | Check |
| 0x00,0x00 | 0x02 | 0x01 | null | 0x03 |

Note: When send wakes up the Command, it needs to send 2 Commands. Because the module takes time to start up, the 1st Command is only used for waking up and may not be parsed. It needs to send 2 Commands.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x01 | null | 0x03 |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xE0 | null | 0xE2 |

Send and return correct example：

【Send Data:】00 00 02 01 03 ; Set the module to enter sleep mode

【Receive Data:】FE 02 01 03

## 8.2 Set antenna and card detection mode

**Function description：**It is used to set the switch and detection card mode of the module's RF antenna. Automatic card detection means that the user only needs to send a read/write command to operate the card data block, no need to send the card search command, which simplifies the process. In manual mode, send the card search command first, and then send the command to operate the card data block such as read/write, which is more complicated. Turning off the antenna can reduce the power consumption of the module. The module default is: turn on the antenna, turn on the automatic card detection

**Send Data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | Check |
| 0x00,0x00 | 0x03 | 0x02 | 1-byte parameter：  BIT0 Antenna status →   BIT0=0：OFF  BIT0=1：ON BIT1 Automatic card detection →   BIT1=0：OFF  BIT1=1：ON | 0”x”XX |

Note：Only the automatic card detection mode module can actively output the card number, see 9.3 for details

This parameter will not be saved after power failure.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x02 | null | 0x00 |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xE1 | null | 0xE3 |

Send and return correct example：

【Send Data:】00 00 03 02 03 02 ;Turn on the antenna and turn on the automatic card detection

【Send Data:】00 00 03 02 01 00 ;Turn on the antenna and turn off the automatic card detection

【Send Data:】00 00 03 02 00 01 ;Turn off the antenna and turn off the automatic card detection

【Receive Data:】FE 02 02 00

## 8.3 Set active output card ID mode

**Function description：**When a card enters the radio frequency area, the buzzer will beep once, and the card ID (4-7 bytes) will be output actively, with the low byte first. Turn on the active output card ID mode, no need to send command to get the card ID, which is simple and convenient. Module default is: Actively output card ID

**Send Datasequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | Check |
| 0x00,0x00 | 0x03 | 0x0C | 1-byte parameter：  00：Turn on automatic card number reading 01：Turn off automatic card number reading | 0”x”XX |

Note：Only the automatic detection card mode module can actively output the card number, see 9.2 for details

This parameter is saved when power off.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x0C | null | 0x0E |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xEB | null | 0xE9 |

**Actively output card ID Data sequence：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Command header | Length | Successful  (Command value) | Data | | Check |
| Card type | 4 or 7 byte sequence number |
| 0xFE | 0x0B(MF0) | 0x03 | 44 00 | xx xx xx xx xx xx xx | 0”x”XX |
| 0xFE | 0x08(MF1 S50) | 0x03 | 04 00 | xx xx xx xx | 0”x”XX |
| 0xFE | 0x08(MF1 S70) | 0x03 | 02 00 | xx xx xx xx | 0”x”XX |
| 0xFE | 0x08(CPU FM1208) | 0x03 | 04 00 | xx xx xx xx | 0”x”XX |

Send and return correct example：

【Send Data:】00 00 03 0C 00 0F ; Open the active output card ID mode

【Send Data:】00 00 03 0C 01 0E ; Turn off the active output card ID mode

【Receive Data:】FE 02 0C 0E

## 8.4 Set LED status

**Function description:** Set the LED switch status of the module. Module default is: off

**Send Datasequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | Check |
| 0x00,0x00 | 0x03 | 0x0D | 1-byte parameter：  00：Turn off the LED 01：Turn on the LED | 0”x”XX |

Note: This parameter will not be saved after power off.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x0D | null | 0x0F |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xEC | null | 0xEE |

Send and return correct example：

【Send Data:】00 00 03 0D 01 0F ;Turn on the LED

【Send Data:】00 00 03 0D 00 0E ;Turn off the LED

【Receive Data:】FE 02 0D 0F

## 8.5 Set the buzzer on time

**Function description：**Set the module buzzer on time 0-255mS (00-FF). The module default value is: 255

**Send Datasequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | Check |
| 0x00,0x00 | 0x03 | 0x0E | 1-byte parameter：0x00-0xFF | 0”x”XX |

Note: This parameter will not be saved after power off.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful(Command value) | Data | Check |
| 0xFE | 0x02 | 0x0E | null | 0x0C |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure(Error value) | Data | Check |
| 0xFE | 0x02 | 0xED | null | 0xEF |

Send and return correct example：

【Send Data:】00 00 03 0E 00 0D ; Set the buzzer on time 0mS

【Send Data:】00 00 03 0E FF F2 ; Set the buzzer on time to 255mS

【Receive Data:】FE 02 0E 0C

## 8.6 Set the serial port baud rate

**Function description：**Set the module serial communication baud rate. Module default baud rate: 19200

**Send Datasequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | Check |
| 0x00,0x00 | 0x03 | 0x0F | 1-byte parameter： 01：9600  02：14400  03：19200  04：28800  05：38400  06：57600  07：115200 | 0”x”XX |

Note: This parameter will not be saved after power off.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x0F | null | 0x0D |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xEE | null | 0xEC |

Send and return correct example：

【Send Data:】00 00 03 0F 01 0D ;Set the serial port baud rate: 9600

【Send Data:】00 00 03 0F 02 0E ;Set the serial port baud rate: 14400

【Send Data:】00 00 03 0F 03 0F ;Set the serial port baud rate: 19200

【Send Data:】00 00 03 0F 04 08 ;Set the serial port baud rate: 28800

【Send Data:】00 00 03 0F 05 09 ;Set the serial port baud rate: 38400

【Send Data:】00 00 03 0F 06 0A ;Set the serial port baud rate: 57600

【Send Data:】00 00 03 0F 07 0B ;Set the serial port baud rate: 115200

【Receive Data:】FE 02 0F 0D

## 8.7 ISO14443 TYPE A search card

**Function description：**ISO14443 TYPE A find card to get card type and ID. Contains cards：**ISO14443** **TYPE A：**MF0: Ultralight、Ultralight C、NTAG203、NTAG213; MF1: S50、S70、FM11RF08; CPU: FM1208。

**Send Datasequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | Check |
| 0x00,0x00 | 0x03 | 0x03 | 1-byte parameter：  00：Find all cards in the antenna area 01：Find a card that is not sleeping | 0”x”XX |

**Correctly return data sequence：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | | Check |
| Card type | 4 or 7 byte sequence number |
| 0xFE | 0x0B(MF0) | 0x03 | 44 00 | xx xx xx xx xx xx xx | 0”x”XX |
| 0xFE | 0x08(MF1 S50) | 0x03 | 04 00 | xx xx xx xx | 0”x”XX |
| 0xFE | 0x08(MF1 S70) | 0x03 | 02 00 | xx xx xx xx | 0”x”XX |
| 0xFE | 0x08(CPU FM1208) | 0x03 | 04 00 | xx xx xx xx | 0”x”XX |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xE2 | null | 0xE0 |

Send and return correct example：

【Send Data:】00 00 03 03 00 00 ; Find all cards in the antenna area

【Send Data:】00 00 03 03 01 01 ; Find a card that is not sleeping

【Receive Data:】FE 0B 03 44 00 04 1A 70 8A 12 49 81 72 ;MF0 NTAG213

【Receive Data:】FE 08 03 04 00 50 F2 12 57 E8 ; MF1 S50

【Receive Data:】FE 08 03 02 00 EB 86 6D 38 31 ; MF1 S70

【Receive Data:】FE 08 03 04 00 5D A2 F2 9A 98 ; CPU FM1208

## 8.8 Mifare one card reading block

**Function description：**Mifare one: S50、S70、FM11RF08 card reading block

**Send Datasequence：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | | | Check |
| Key ID (1Byte) | Block number  (1Byte) | Key  (6Byte) |
| 0x00,0x00 | 0x0A | 0x04 | BIT0 ＝0：A Key  ＝1：B Key BIT1 ＝0：Use the 6-byte Key in the instruction | **S50:**0～63  **S70:**0～255 | Factory default:  FF FF FF FF FF FF | 0”x”XX |

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful  (Command value) | Data | Check |
| 16 bytes of data read |
| 0xFE | 0x12 | 0x04 | xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx | 0”x”XX |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xE3 | null | 0xE1 |

Send and return correct example：

【Send Data:】00 00 0A 04 00 01 FF FF FF FF FF FF 0F ; Read block 1Data

【Receive Data:】FE 12 04 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 16 ; The blue part is the read block 1Data

## 8.9 Mifare one card writing block

**Function description：**Mifare one: S50、S70、FM11RF08 **card writing block**。

**Send Datasequence：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | | | | Check |
| Key ID  (1Byte) | Block number  (1Byte) | Key  (6Byte) | 16 bytes of data written into the block |
| 0x00,0x00 | 0x1A | 0x05 | BIT0 ＝0：A Key  ＝1：B Key BIT1 ＝0：Use the 6-byte Key in the instruction | **S50:**0～63  **S70:**0～255 | The factory default:  FF FF FF FF FF FF | xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx | 0”x”XX |

Note：The third block of each sector is the Key block. Writing this block is equivalent to changing the Key. Philips/NXP has an important update to the S50 chip in 2001: When BKey can be read, BKey becomes invalid. If you want to use BKey, modify the original Key control bit DataFF 07 80 69 to 7F 07 88 69; after modifying this control bit: Modification of Key A, B (write sector block 3), you must use BKey; sector block 0 -2, both A and BKey can be read and written.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x05 | null | 0x07 |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xE4 | null | 0xE6 |

Send and return correct example：

【Send Data:】00 00 1A 05 00 01 FFF FF FF FF FF FF 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 1E ; Write block 1 data, the blue part is the data written in block 1

【Send Data:】00 00 1A 05 00 03 FFF FF FF FF FF FF 01 02 03 04 05 06 7F 07 88 69 01 02 03 04 05 06 85 ; Write block 3 to modify 0 sector A, BKey are 01 02 03 04 05 06, the key control bit is 7F 07 88 69

【Receive Data:】FE 02 05 07

## 8.10 Mifare one card initialization wallet

**Function description：**Mifare one: S50, S70, FM11RF08 cards set a block as a wallet and initialize the wallet value。

**Send Datasequence：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | | | | Check |
| Key ID  (1Byte) | Block number  (1Byte) | Key  (6Byte) | Write 4 bytes of wallet initial value (low byte first) |
| 0x00,0x00 | 0x0E | 0x06 | BIT0 ＝0：A Key  ＝1：B Key BIT1 ＝0：Use the 6-byte Key in the instruction | **S50:**0～63  **S70:**0～255 | The factory default:  FF FF FF FF FF FF | xx xx xx xx | 0”x”XX |

Note：The third block of each sector is the Key block and cannot be used for the wallet block. The low byte of the initial value of the 4-byte wallet is first.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x06 | null | 0x04 |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xE5 | null | 0xE7 |

Send and return correct example：

【Send Data:】00 00 0E 06 00 05 FFF FF FF FF FF FF 00 00 00 00 0D ; Initialize wallet block 5, the blue value is 0, the low byte is first

【Receive Data:】FE 02 06 04

## 8.11 Mifare one card reader wallet

**Function description：**Mifare one: S50、S70、FM11RF08 card reading or query wallet block。

**Send Datasequence：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | | | Check |
| Key ID  (1Byte) | Block number  (1Byte) | Key  (6Byte) |
| 0x00,0x00 | 0x0A | 0x07 | BIT0 ＝0：A Key  ＝1：B Key BIT1 ＝0：Use the 6-byte Key in the instruction | **S50:**0～63  **S70:**0～255 | The factory default:  FF FF FF FF FF FF | 0”x”XX |

Note：The third block of each sector is the Key block and cannot be used for the wallet block. This block must be initialized as a wallet in advance.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful  (Command value) | Data | Check |
| 4-byte wallet value read (low byte first) |
| 0xFE | 0x06 | 0x07 | xx xx xx xx | 0”x”XX |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xE6 | null | 0xE4 |

Send and return correct example：

【Send Data:】00 00 0A 07 00 05 FF FF FF FF FF FF 08 ; Read block 5 wallet value

【Receive Data:】FE 06 07 00 00 00 00 01 ; The blue part is the read block 5 wallet value, with low byte first

## 8.12 Mifare one Card charge value wallet

**Function description：**Mifare one: S50、S70、FM11RF08 card charge value wallet block.

**Send Datasequence：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | | | | Check |
| Key ID  (1Byte) | Block number  (1Byte) | Key  (6Byte) | Charge into wallet value 4 bytes (low byte first) |
| 0x00,0x00 | 0x0E | 0x08 | BIT0 ＝0：A Key  ＝1：B Key BIT1 ＝0：Use the 6-byte Key in the instruction | **S50:**0～63  **S70:**0～255 | The factory default:  FF FF FF FF FF FF | xx xx xx xx | 0”x”XX |

Note：The third block of each sector is the Key block and cannot be used for the wallet block. 4-byte wallet value low byte first.

**Correctly return Data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x08 | null | 0x0A |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xE7 | null | 0xE5 |

Send and return correct example：

【Send Data:】00 00 0E 08 00 05 FFF FF FF FF FF FF 02 00 00 00 01 ; The value of wallet block 5 is charged to 2, the blue value is 2, and the low byte is first

【Receive Data:】FE 02 08 0A

## 8.13 Mifare one card deduction wallet

**Function description：**Mifare one: S50、S70、FM11RF08 card deduction wallet block。

**Send Datasequence：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | | | | Check |
| Key ID  (1Byte) | Block number  (1Byte) | Key  (6Byte) | Deduction wallet value 4 bytes (low byte first) |
| 0x00,0x00 | 0x0E | 0x09 | BIT0 ＝0：A Key  ＝1：B Key BIT1 ＝0：Use the 6-byte Key in the instruction | **S50:**0～63  **S70:**0～255 | The factory default:  FF FF FF FF FF FF | xx xx xx xx | 0”x”XX |

Note：The third block of each sector is the Key block and cannot be used for the wallet block. 4-byte wallet value low byte first.。

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x09 | null | 0x0B |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure(Error value) | Data | Check |
| 0xFE | 0x02 | 0xE8 | null | 0xEA |

Send and return correct example：

【Send Data:】00 00 0E 09 00 05 FFF FF FF FF FF FF 01 00 00 00 03 ; The value of wallet block 5 is deducted 1, the blue value is 1, the low byte is first

## 【Receive Data:】FE 02 09 0B

## 8.14 Mifare one card backup wallet

**Function description：**Mifare one: S50、S70、FM11RF08 card backup wallet block。

**Send data sequence：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | | | | Check |
| Key ID  (1Byte) | Current Wallet Block number (1Byte) | Backup wallet Block number (1Byte) | Key  (6Byte) |
| 0x00,0x00 | 0x0B | 0x0A | BIT0 ＝0：A Key  ＝1：B Key BIT1 ＝0：Use the 6-byte Key in the instruction | **S50:**0～63  **S70:**0～255 | **S50:**0～63  **S70:**0～255 | The factory default:  FF FF FF FF FF FF | 0”x”XX |

Note：The third block of each sector is the Key block and cannot be used for the wallet block. The backup wallet spans sectors. The current wallet Block numberKey and the backup wallet Block numberKey must be the same. It is recommended to use the same sector for backup.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x0A | null | 0x08 |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xE9 | null | 0xEB |

Send and return correct example：

【Send Data:】00 00 0B 0A 00 05 06 FFF FF FF FF FF FF 02 ; Backup wallet block 5 to block 6

【Receive Data:】FE 02 0A 08

## 8.15 Mifare Ultralight (M0) card page reading

**Function description：**Ultralight、Ultralight C、NTAG203、NTAG213 card page reading。

**Send Datasequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | Check |
| Starting page number (1Byte) |
| 0x00,0x00 | 0x03 | 0x14 | **Ultralight:**0-15  **Ultralight C:**0-47  **NTAG203:**0-41  **NTAG213:**0-44 | 0”x”XX |

Note：Continuously read 4 pages of 16-byte data starting from the starting page number, 4 bytes of data per page。

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful(Command value) | Data | Check |
| 4 consecutive pages of 16-byte data read |
| 0xFE | 0x12 | 0x14 | xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx | 0”x”XX |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xE3 | null | 0xE1 |

Send and return correct example：

【Send Data:】00 00 03 14 04 13 ; Read the first 16 bytes of page 4

【Receive Data:】FE 12 14 11 11 11 11 22 22 22 22 00 00 00 00 00 00 00 00 06 ; The red and blue part is the read Data

## 8.16 Mifare Ultralight (M0) card write page

**Function description：**Ultralight、Ultralight C、NTAG203、NTAG213 card write page。

**Send data sequence：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | | Check |
| Page number (1Byte) | 4-byte Data written to the page |
| 0x00,0x00 | 0x07 | 0x15 | **Ultralight:**0-15  **Ultralight C:**0-47  **NTAG203:**0-41  **NTAG213:**0-44 | xx xx xx xx | 0”x”XX |

Note：4 bytes Data per page.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x15 | null | 0x17 |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xE4 | null | 0xE6 |

Send and return correct example：

【Send Data:】00 00 07 15 04 11 11 11 11 16 ; Write the data on page 4, the blue part is the 4 bytes of data written

【Receive Data:】FE 02 15 17

## 8.17 Mifare one/Ultralight card sleep

**Function description：**Mifare one/Ultralight card sleep。

**Send data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | Check |
| 0x00,0x00 | 0x02 | 0x0B | null | 0x09 |

Note：This Command is valid only when the automatic card detection mode is turned off (set to manual mode), see 9.2 for details

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful (Command value) | Data | Check |
| 0xFE | 0x02 | 0x0B | null | 0x09 |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xEA | null | 0xE8 |

Send and return correct example：

【Send Data:】00 00 02 0B 09

【Receive Data:】FE 02 0B 09

## 8.18 ISO14443 TYPE A CPU card reset

**Function description：**It is used to perform the reset operation of the CPU card meeting ISO14443 TYPE A, and the card reset information is returned Correctly.

Note: The CPU card operation module will close the automatic card search, prohibit the module from automatically outputting the card number, and the CPU card operation is completed Send Command 00 00 03 02 03 02 to open the automatic card search; or use the manual card search method Send Command 00 00 03 03 00 00 to obtain the card number, Avoid operating Failure on other cards.

**Send Datasequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | Check |
| 0x00,0x00 | 0x03 | 0x20 | 0x26 means looking for non-sleeping cards within the antenna range  0x52 means to find all cards within the antenna range | 0”x”XX |

**Correctly return Datasequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful  (Command value) | Data | Check |
| 0xFE | 0x16 | 0x20 | 4 bytes CSN+16 bytes information | 0”x”XX |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xF0 | null | 0xF2 |

Send and return correct example：

【Send Data:】00 00 03 20 26 05 ; CPU card reset, find all cards in the antenna area

【Send Data:】00 00 03 20 52 71 ; CPU card reset, find the card not in sleep state

【Receive Data:】FE 16 20 5D A2 F2 9A 10 78 80 90 02 20 90 00 00 00 00 00 5D A2 F2 9A FC ; The color part is the information returned

## 8.19 ISO14443 TYPE A CPU card Send COS instruction

**Function description：**Used for Send ISO14443 TYPE A CPU card COS command.

Note：The CPU card operation module will close the automatic card search, prohibit the module from automatically outputting the card number, and the CPU card operation is completed Send Command 00 00 03 02 03 02 to open the automatic card search; or use the manual card search method Send Command 00 00 03 03 00 00 to obtain the card number, Avoid operating Failure on other cards.

**Send Datasequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module address | Length | Command | Data | Check |
| 0x00,0x00 | 0”x”XX | 0x21 | COS command | 0”x”XX |

Note：Send CPU card reset instruction first, this instruction is valid.

**Correctly return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Successful  (Command value) | Data | Check |
| 0xFE | 0”x”XX | 0x21 | Return Data for COS instruction | 0”x”XX |

**Error return data sequence：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Command header | Length | Failure (Error value) | Data | Check |
| 0xFE | 0x02 | 0xF1 | null | 0xF3 |

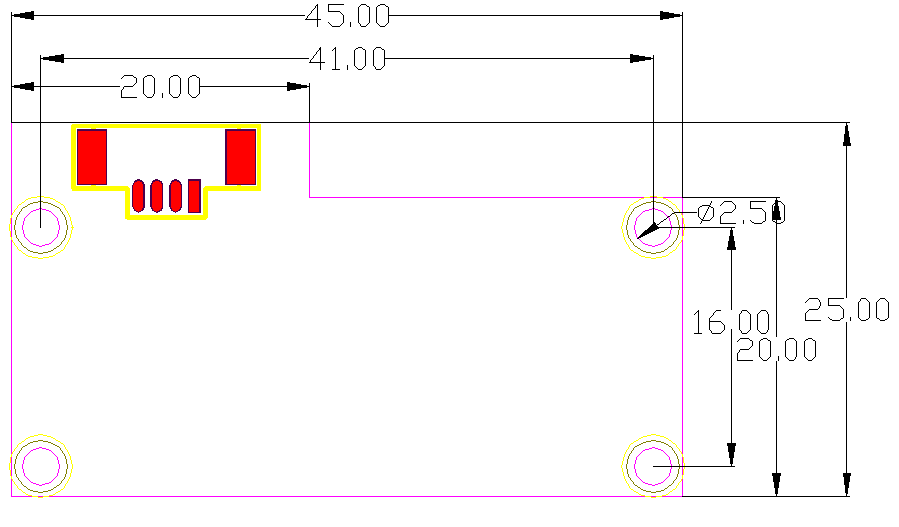
Send and return correct example：

【Send Data:】00 00 07 21 00 84 00 00 04 A6 ; Take a 4-byte random number instruction (the blue part). Send CPU card reset instruction first, this instruction is valid.

【Receive Data:】FE 08 21 EE 6E E2 F3 90 00 28 ; Red is a 4-byte random number; blue 90 00 card return indicates successful

# 9. Size parameters

Unit：mm



# 10. Appendix 1: Command summary table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO. | Command | Description | Executecorrectly return | Execute Error return |
| Module settings | | | | |
| 1 | 0x01 | Set the module to enter sleep mode | 0x01 | 0xE0 |
| 2 | 0x02 | Set antenna and card detection mode | 0x02 | 0xE1 |
| 3 | 0x0C | Set active output card ID mode | 0x0C | 0xEB |
| 4 | 0x0D | Set LED status | 0x0D | 0xEC |
| 5 | 0x0E | Set the buzzer on time | 0x0E | 0xED |
| 6 | 0x0F | Set the serial port baud rate | 0x0F | 0xEE |
| Card operation | | | | |
| 7 | 0x03 | ISO14443 TYPE A search card | 0x03 | 0xE2 |
| 8 | 0x04 | Mifare one card reading block | 0x04 | 0xE3 |
| 9 | 0x05 | Mifare one card write block | 0x05 | 0xE4 |
| 10 | 0x06 | Mifare one card initialization wallet | 0x06 | 0xE5 |
| 11 | 0x07 | Mifare one card reader wallet | 0x07 | 0xE6 |
| 12 | 0x08 | Mifare one card recharge value wallet | 0x08 | 0xE7 |
| 13 | 0x09 | Mifare one card deduction wallet | 0x09 | 0xE8 |
| 14 | 0x0A | Mifare one card backup wallet | 0x0A | 0xE9 |
| 15 | 0x14 | Mifare Ultralight (M0) card page reading | 0x14 | 0xE3 |
| 16 | 0x15 | Mifare Ultralight (M0) card write page | 0x15 | 0xE4 |
| 17 | 0x0B | Mifare one/Ultralight card sleep | 0x0B | 0xEA |
| 18 | 0x20 | ISO14443 TYPE A CPU card reset | 0x20 | 0xF0 |
| 19 | 0x21 | ISO14443 TYPE A CPU card Send COS instruction | 0x21 | 0xF1 |