MM Version Reader Control Protocol

User Manual

|  |  |  |
| --- | --- | --- |
| Version Control | | |
| **Date** | **Version** | **Content** |
| 2019-04-28  2019-11-16  2020-06-17 | V1.0 V1.1 V1.2 | Initial version Update some commands  Update Multi antenna commands |

# Table of Contents

[Table of Contents 1](#_bookmark0)

1. [Introduction 3](#_bookmark1)
2. [Data’s type and the basic format of protocols 4](#_bookmark2)
   1. [Data’s Type 4](#_bookmark3)
   2. [Basic format of protocol 4](#_bookmark4)
   3. [Data Format 5](#_bookmark5)
3. [Code Table 6](#_bookmark6)
4. [Communication Protocol 8](#_bookmark7)
   1. [Read Type C UII 9](#_bookmark8)
   2. [Read Type C Tag Data 10](#_bookmark9)
   3. [Write Type C Tag Data 11](#_bookmark10)
   4. [Lock Type C Tag 11](#_bookmark11)
   5. [Kill Type C Tag 13](#_bookmark12)
   6. [Encrypted Type C Tag 14](#_bookmark13)
   7. [Get Access EPC match status 15](#_bookmark14)
   8. [Set Access EPC match status 15](#_bookmark15)
   9. [Get Tx Power Level 16](#_bookmark16)
   10. [Set Tx Power Level 17](#_bookmark17)
   11. [Get Region 17](#_bookmark18)
   12. [Set Region 18](#_bookmark19)
   13. [Get current RF Channel 19](#_bookmark20)
   14. [Set current RF Channel 19](#_bookmark21)
   15. [Get Frequency Hopping Table \* 20](#_bookmark22)
   16. [Set Frequency Hopping Table \* 20](#_bookmark23)
   17. [Get Modulation 21](#_bookmark24)
   18. [Set Modulation 22](#_bookmark25)
   19. [Get Frequency Hopping status 23](#_bookmark26)
   20. [Set Frequency Hopping status 23](#_bookmark27)
   21. [Get Base Parameters 24](#_bookmark28)
   22. [Set Base Parameters 26](#_bookmark29)
   23. [Get Antenna configuration \* 27](#_bookmark30)
   24. [Set Antenna configuration \* 28](#_bookmark31)
   25. [Get Encryption method for Tag 29](#_bookmark32)
   26. [Set Encryption method for Tag 30](#_bookmark33)
   27. [Get Protocol Address 30](#_bookmark34)
   28. [Set Protocol Address 31](#_bookmark35)
   29. [Get UART Baudrate \* 31](#_bookmark36)
   30. [Set UART Baudrate \* 31](#_bookmark37)
   31. [Get Output Mode \* 31](#_bookmark38)
   32. [Set Output Mode \* 33](#_bookmark39)
   33. [Reset System 34](#_bookmark40)
   34. [Update Registry 34](#_bookmark41)
   35. [Erase Registry 35](#_bookmark42)
   36. [Get GPIO Mode \* 35](#_bookmark43)
   37. [Set GPIO Mode \* 35](#_bookmark44)

## Introduction

###### Communications protocol definition

Support RS232/ RS485;

The mode of information transmission is asynchronous ，date bits: 8 ，stop bits: 1 ，no checksum。 Rate of data transmission: 57.6kb/s;

The monitoring unit (SU) and device control module (SM) communication mainly from the way, the monitoring unit for the host computer, slave computer monitoring module. SU call SM and issue the command, SM receives the command returns response information. SU 500ms is not receiving a SM response or receiving response information error, think of the communication process to fail.

Supervision Unit (SU): like PC or control device; Supervisory Module (SM): Reader;

Note: Communication data is HEX; **Denotation Method xxH;**

## Data’s type and the basic format of protocols

### Data’s Type

Two types:

* Command: SU to SM;
* Response: SM to SU;

### Basic format of protocol

Table 2-1 basic format of protocols

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| byte | 1 | 2 | 1 | 1 | 1 | LENGTH | 1 |
| format | SOI | ADR | CID1 | CID2 | LENGTH | INFO | CHKSUM |

Table 2-2 basic format of notes

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Symbol | significance | Remarks |
| 1 | SOI | START OF INFORMATION | Command(7CH)  Response(CCH) |
| 2 | ADR | Equip address （ 1 ～ 65534 ） ,(65535 public address,0 reserve  address) | FFFFH |
| 3 | CID1 | Control identification code (data type description) |  |
| 4 | CID2 | Command: control identification code (action type description)  Response: RTN（Return code Table 2-3） |  |
| 5 | LENGTH | INFO Data Length |  |
| 6 | INFO | Command: Command information  Response: Response data information |  |
| 7 | CHKSUM | The checksum code |  |

Table 2-3 Return code RTN

|  |  |  |  |
| --- | --- | --- | --- |
| No. | RTN Value(HEX) | significance | Remarks |
| 1 | 00H | Succeed |  |
| 2 | 01H | Fail |  |
|  | 02H | Response massage for Command |  |
| 3 | 05H | Auto send to SU |  |

### Data Format

CHKSUM data format:

###### CHKSUM Introduction

The calculation of CHKSUM is in addition to CHKSUM, other characters in 16 hex code values of cumulative sum, the result modulo 256 remainder taking anti - plus 1.

For example：Receive or send data is：“CC 02 01 B1 22 04 BB 12 02 03 88”. The last byte “88” is CHKSUM.

Calculate as follows： ‘CC’＋‘02’＋‘01’＋…＋‘22’＋‘04’＋‘BB’＋‘12’ ＋‘02’＋‘03’

= CCH + 02H + 01H + … + 22H + 04H + BBH + 12H + 02H + 03H

= 0278H

0278H mode 256 and the remainder is 78H, 78H anti plus 1 is 88H.

###### CHKSUM Calculate refers:

unsigned char Checksum ( unsigned char \*uBuff, unsigned char uBuffLen)

{

unsigned char i, uSum =0; for(i=0; i<uBuffLen; i++)

{

uSum = uSum + uBuff[i];

}

uSum = (~uSum) + 1; return uSum;

}

# Code Table

CID1、CID2 Code Distribution and Classification as follows :

Table 3-1 Command code Classification (SENIOR CID1)

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Content | CID1 | Remark |
| 1 | Read Type C UII | 20H |  |
| 2 | Read Type C Tag Data | 21H |  |
| 3 | Write Type C Tag Data | 22H |  |
| 4 | Lock Type C Tag | 26H |  |
| 5 | Kill/Recom Type C Tag | 28H |  |
| 6 | Encrypted Type C Tag | 2AH |  |
| 7 | Get Access EPC MATCH | 2CH |  |
| 8 | Set Access EPC MATCH | 2DH |  |
| 9 | Get Tx Power Level | 50H |  |
| 10 | Set Tx Power Level | 51H |  |
| 11 | Get Region | 52H |  |
| 12 | Set Region | 53H |  |
| 13 | Get current RF Channel | 54H |  |
| 14 | Set current RF Channel | 55H |  |
| 15 | Get Frequency Hopping Table \* | 56H |  |
| 16 | Set Frequency Hopping Table \* | 57H |  |
| 17 | Get Modulation \* | 58H |  |
| 18 | Set Modulation \* | 59H |  |
| 19 | Get Frequency Hopping status | 5AH |  |
| 20 | Set Frequency Hopping status | 5BH |  |
| 21 | Base Parameter | 81H |  |
| 22 | Antenna configuration \* | 83H |  |
| 23 | Encryption method for Tag | 84H |  |
| 24 | Protocol Address | 85H |  |
| 25 | UART Baud rate \* | 86H |  |
| 26 | Output Mode \* | 87H |  |
| 27 | Reset System | D0H |  |
| 28 | Update Registry | D2H |  |
| 29 | Erase Registry | D3H |  |
| 30 | Get GPIO Mode \* | D6H |  |
| 31 | Set GPIO Mode \* | D7H |  |
| 32 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Table 3-2 Command action Classification (CID2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. |  | Content | CID2 | Remarks |
|  | 1 | Senior command | 00H |  |
|  | 2 | Set command | 31H |  |
|  | 3 | Get command | 32H |  |

# Communication Protocol

For the use of this protocol in the protocol code as follows.

Table 4-1 protocol code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Content | CID1 | CID2 | Remarks |
| 1 | Read Type C UII | 20H | 00H |  |
| 2 | Read Type C Tag Data | 21H | 00H |  |
| 3 | Write Type C Tag Data | 22H | 00H |  |
| 4 | Lock Type C Tag | 26H | 00H |  |
| 5 | Kill/Recom Type C Tag | 28H | 00H |  |
| 6 | Encrypted Type C Tag | 2AH | 00H |  |
| 7 | Get Access EPC MATCH | 2CH | 00H |  |
| 8 | Set Access EPC MATCH | 2DH | 00H |  |
| 9 | Get Tx Power Level | 50H | 00H |  |
| 10 | Set Tx Power Level | 51H | 00H |  |
| 11 | Get Region | 52H | 00H |  |
| 12 | Set Region | 53H | 00H |  |
| 13 | Get current RF Channel | 54H | 00H |  |
| 14 | Set current RF Channel | 55H | 00H |  |
| 15 | Get Frequency Hopping Table \* | 56H | 00H |  |
| 16 | Set Frequency Hopping Table \* | 57H | 00H |  |
| 17 | Get Modulation \* | 58H | 00H |  |
| 18 | Set Modulation \* | 59H | 00H |  |
| 19 | Get Frequency Hopping status | 5AH | 00H |  |
| 20 | Set Frequency Hopping status | 5BH | 00H |  |
| 21 | Get Base Parameters | 81H | 32H |  |
| 22 | Set Base Parameters | 81H | 31H |  |
| 23 | Get Antenna configuration \* | 83H | 32H |  |
| 24 | Set Antenna configuration \* | 83H | 31H |  |
| 25 | Get Encryption method for Tag | 84H | 32H |  |
| 26 | Set Encryption method for Tag | 84H | 31H |  |
| 27 | Get Protocol Address | 85H | 32H |  |
| 28 | Set Protocol Address | 85H | 31H |  |
| 29 | Get UART Baudrate \* | 86H | 32H |  |
| 30 | Set UART Baudrate \* | 86H | 31H |  |
| 31 | Get Output Mode \* | 87H | 32H |  |
| 32 | Set Output Mode \* | 87H | 31H |  |
| 33 | Reset System | D0H | 00H |  |
| 34 | Update Registry | D2H | 00H |  |
| 35 | Erase Registry | D3H | 00H |  |
| 36 | Get GPIO Mode \* | D6H | 00H |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 37 | Set GPIO Mode \* | D7H | 00H |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Note: with \* command representation is optional command; the reader does not have this feature, if have this feature, should be in accordance with the execution of this agreement. (Hereinafter appearing \* place, meaning as described above, not detailed below.)

### Read Type C UII

#### When the working mode is set to active, this command does not need to be sent, and the reader will automatically read and response,RTN is 05H;

* + 1. Command

CID1: 20H

CID2: 00H

INFO: - None.

Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 20 | 00 | 00 | 0xNN |

* + 1. Tag Response(When there are multiple tags, the answer returns multiple)

CID1: 20H

RTN: 02H

INFO: - ANT (8-bit): Ant No.(Def 0x00)

* EPC (variable): Target tag’s PC+EPCS
* RSSI (8-bit): RSSI

Example: **ANT=0x00, PC = 0x3000, EPC = 0xE2003411B802011383258566, RSSI=0xC9**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **ANT** |
| CC | FF | FF | 20 | 02 | 10 | 00 |
| **PC (MSB)** | **PC (LSB)** | **EPC (MSB)** | **--** | **--** | **--** | **--** |
| 30 | 00 | E2 | 00 | 34 | 11 | B8 |
| **--** | **--** | **--** | **--** | **--** | **--** | **EPC (LSB)** |
| 02 | 01 | 13 | 83 | 25 | 85 | 66 |
| **RSSI** | **CHECKSUM** |  |  |  |  |  |
| C9 | 0xNN |  |  |  |  |  |

* + 1. Response

CID1: 20H

RTN: 00H

INFO: - ANT (8-bit): Ant No.(Def 0x00)

* STC (8-bit): Send Tag Count
* RTC (8-bit): Read Tag Count

Example: **ANT=0x00, STC = 0x27, RTC=0x27**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **ANT** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CC | FF | FF | 20 | 02 | 03 | 00 |
| **STC** | **RCT** | **CHECKSUM** |  |  |  |  |
| 27 | 27 | 0xNN |  |  |  |  |

### Read Type C Tag Data

##### This command should be preceded by a match EPC status command (see 4.8)

* + 1. Command

CID1: 21H

CID2: 00H

INFO: - AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filed to 0x00000000.

* MB (8-bit): Target memory bank; RFU (0x00), EPC (0x01), TID (0x02), User (0x03)
* SA (8-bit): Starting Address word pointer (Word)
* DL (8-bit): Data Length (Word Count). Example: **Access Password = 0x00000000,**

**Target memory bank = EPC, Start Address = 0x02, Length = 2 word**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **AP(MSB)** |
| 7C | FF | FF | 21 | 00 | 07 | 00 |
| **--** | **--** | **AP(LSB)** | **MB** | **SA** | **DL** | **CHECKSUM** |
| 00 | 00 | 00 | 01 | 02 | 02 | 0xNN |

* + 1. Response

CID1: 21H

RTN: 00H

INFO: - ANT (8-bit): Ant No.(Def 0x00)

* EPC (variable): Target tag’s PC+EPC
* DT(variable): Tag memory contents Example 如: **ANT=0x00,**

**PC = 0x3000,**

**EPC = 0xE2003411B802011383258566, DT=0x E2003411**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **ANT** |
| CC | FF | FF | 21 | 00 | 13 | 00 |
| **PC (MSB)** | **PC (LSB)** | **EPC (MSB)** | **--** | **--** | **--** | **--** |
| 30 | 00 | E2 | 00 | 34 | 11 | B8 |
| **--** | **--** | **--** | **--** | **--** | **--** | **EPC (LSB)** |
| 02 | 01 | 13 | 83 | 25 | 85 | 66 |
| **DT (MSB)** | **--** | -- | **DT (LSB)** | **CHECKSUM** |  |  |
| E2 | 00 | 34 | 11 | 0xNN |  |  |

### Write Type C Tag Data

##### This command should be preceded by a match EPC status command (see 4.8).

* + 1. Command

CID1: 22H

CID2: 00H

INFO: - AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filed to 0x00000000.

* MB (8-bit): Target memory bank; RFU (0x00), EPC (0x01), TID (0x02), User (0x03)
* SA (8-bit): Starting Address word pointer (Word)
* DL (8-bit): Data Length (Word Count).
* DT (variable): Data to write.

Example: **Access Password = 0x00000000, Target memory bank = EPC,**

**Start Address = 0x02, Data Length = 2 word,**

**Data to write = 0x12345678**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **AP(MSB)** |
| 7C | FF | FF | 22 | 00 | 0B | 00 |
| **--** | **--** | **AP(LSB)** | **MB** | **SA** | **DL** | **DT(MSB)** |
| 00 | 00 | 00 | 01 | 02 | 02 | 12 |
| **--** | **--** | **DT(LSB)** | **CHECKSUM** |  |  |  |
| 34 | 56 | 78 | 0xNN |  |  |  |

* + 1. Response

CID1: 22H

RTN: 00H

INFO: - ANT (8-bit): Ant No.(Def 0x00)

- EPC (variable): Target tag’s PC+EPC Example: **ANT=0x00,**

**PC = 0x3000,**

**EPC = 0xE2003411B802011383258566**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **ANT** |
| CC | FF | FF | 22 | 00 | 0F | 00 |
| **PC (MSB)** | **PC (LSB)** | **EPC (MSB)** | **--** | **--** | **--** | **--** |
| 30 | 00 | E2 | 00 | 34 | 11 | B8 |
| **--** | **--** | **--** | **--** | **--** | **--** | **EPC (LSB)** |
| 02 | 01 | 13 | 83 | 25 | 85 | 66 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

### Lock Type C Tag

##### This command should be preceded by a match EPC status command (see 4.8).

* + 1. Command

CID1: 26H

CID2: 00H

INFO: - AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filed to 0x00000000.

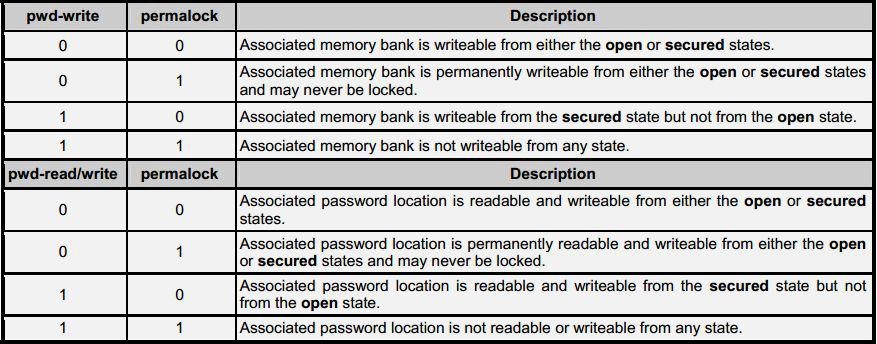
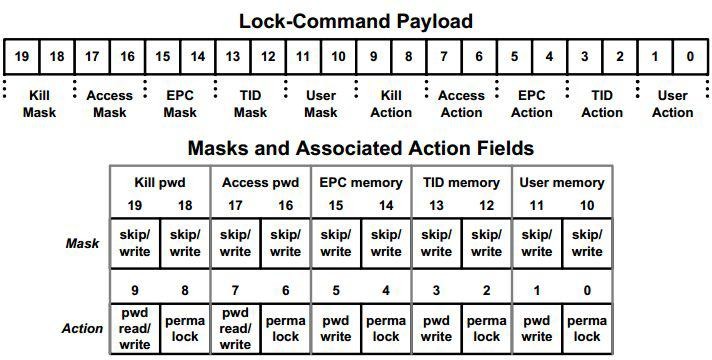
- LD (24-bit): Lock Data.

*The high 4 bits of lock operation parameter LD are reserved bits, and the remaining 20 bits are lock operation payload, including mask and action, 10 bits from high to low. For details, please refer to section 6.3.2.11.3.5 of EPC Gen2 protocol version 1.2.0.*

*Mask is a mask. Only actions with mask bit 1 are valid. Actions in each data area have 2 bits, 00-11, which correspond to opening, permanent opening, locking and permanent locking.*

*For example, if the kill mask is 2bits 00, the kill action will not take effect regardless of the kill action. When the kill mask is 2bits 10 and the kill action is 2bits 10, it means that the kill password is locked (not perma lock). Only through a valid access password can it be read and written.*

*The meaning of each bit of mask and action is shown in the table below.*

**

Example: If you need lock Access Password，then:

**Access Password = 0x0000FFFF, LD = 0x020080**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **AP(MSB)** |
| 7C | FF | FF | 26 | 00 | 07 | 00 |
| **--** | **--** | **AP(LSB)** | **LD(MSB)** | **--** | **LD(LSB)** | **CHECKSUM** |
| 00 | FF | FF | 02 | 00 | 80 | 0xNN |

* + 1. Response

CID1: 26H

RTN: 00H

INFO: - ANT (8-bit): Ant No.(Def 0x00)

* EPC (variable): Target tag’s PC+EPC
* DT(variable): Tag memory contents Example: **ANT=0x00,**

**PC = 0x3000,**

**EPC = 0xE2003411B802011383258566**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **ANT** |
| CC | FF | FF | 26 | 00 | 0F | 00 |
| **PC (MSB)** | **PC (LSB)** | **EPC (MSB)** | **--** | **--** | **--** | **--** |
| 30 | 00 | E2 | 00 | 34 | 11 | B8 |
| **--** | **--** | **--** | **--** | **--** | **--** | **EPC (LSB)** |
| 02 | 01 | 13 | 83 | 25 | 85 | 66 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

### Kill Type C Tag

##### This command should be preceded by a match EPC status command (see 4.8).

* + 1. Command

CID1: 28H

CID2: 00H

INFO: - KP (32-bit): Kill Password. If KP filed set to 0x00000000, ‘Kill Type C Tag’ command do not work. The target tag ignores it.

- Recom (8-bit): Recommissioning bits. Example: **Kill Password =0x87654321, Recom = 0x00**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **KP(MSB)** |
| 7C | FF | FF | 28 | 00 | 05 | 87 |
| **--** | **--** | **KP(LSB)** | **Recom** | **CHECKSUM** |  |  |
| 65 | 43 | 21 | 00 | 0xNN |  |  |

* + 1. Response

CID1: 28H

RTN: 00H

INFO: - ANT (8-bit): Ant No.(Def 0x00)

- EPC (variable): Target tag’s PC+EPC Example: **ANT=0x00,**

**PC = 0x3000,**

**EPC = 0xE2003411B802011383258566**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **ANT** |
| CC | FF | FF | 28 | 00 | 0F | 00 |
| **PC (MSB)** | **PC (LSB)** | **EPC (MSB)** | **--** | **--** | **--** | **--** |
| 30 | 00 | E2 | 00 | 34 | 11 | B8 |
| **--** | **--** | **--** | **--** | **--** | **--** | **EPC (LSB)** |
| 02 | 01 | 13 | 83 | 25 | 85 | 66 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

### Encrypted Type C Tag

(Limited encryption mode, only valid for the company's production equipment)

***Before this instruction, set the label encryption method of the device (see 4.26).***

***Otherwise, the encryption command is invalid.***

* + 1. Command

CID1: 2AH

CID2: 00H

INFO: - AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filed to 0x00000000.

Example: **Access Password = 0x00000000**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **AP(MSB)** |
| 7C | FF | FF | 28 | 00 | 04 | 00 |
| **--** | **--** | **AP(LSB)** | **CHECKSUM** |  |  |  |
| 00 | 00 | 00 | 0xNN |  |  |  |

* + 1. Response

CID1: 2AH

RTN: 00H

INFO: - ANT (8-bit): Ant No.(Def 0x00)

- EPC (variable): Target tag’s PC+EPC Example: **ANT=0x00,**

**PC = 0x3000,**

**EPC = 0xE2003411B802011383258566**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **ANT** |
| CC | FF | FF | 2A | 00 | 0F | 00 |
| **PC (MSB)** | **PC (LSB)** | **EPC (MSB)** | **--** | **--** | **--** | **--** |
| 30 | 00 | E2 | 00 | 34 | 11 | B8 |
| **--** | **--** | **--** | **--** | **--** | **--** | **EPC (LSB)** |
| 02 | 01 | 13 | 83 | 25 | 85 | 66 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

### Get Access EPC match status

* + 1. Command

CID1: 2CH

CID2: 00H

INFO: -None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 2C | 00 | 00 | 0xNN |

* + 1. Response

CID1: 2CH

RTN: 00H

INFO: - MODE (8-bit): Match Mode

0x00,Mismatch 0x01,select tag action

0x02,select tag action,def use this mode

* LEN (8-bit): Mach EPC length
* EPC (variable): Target tag’s EPC data Example1: **MODE=0x00,**

**LEN=0x00,**

**EPC = null,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **MODE** |
| CC | FF | FF | 2C | 00 | 02 | 00 |
| **LEN** | **CHECKSUM** |  |  |  |  |  |
| 00 | 0xNN |  |  |  |  |  |

Example2: **MODE=0x02,**

**LEN=0x0C,**

**EPC = 0xE2003411B802011383258566,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **MODE** |
| CC | FF | FF | 2C | 00 | 0E | 02 |
| **LEN** | **EPC (MSB)** | **--** | **--** | **--** | **--** | **--** |
| 0C | E2 | 00 | 34 | 11 | B8 | 02 |
| **--** | **--** | **--** | **--** | **--** | **EPC (LSB)** | **CHECKSUM** |
| 01 | 13 | 83 | 25 | 85 | 66 | 0xNN |

### Set Access EPC match status

* + 1. Command

CID1: 2DH

CID2: 00H

INFO: - MODE (8-bit): match mode

0x00,Cancel current match 0x01,select tag action

0x02,select tag action,def use this mode

* LEN (8-bit): Mach EPC length
* EPC (variable): Target tag’s EPC data

Example1: Cancel match **MODE=0x00, LEN=0x00,**

**EPC = null,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **MODE** |
| 7C | FF | FF | 2D | 00 | 02 | 00 |
| **LEN** | **CHECKSUM** |  |  |  |  |  |
| 00 | 0xNN |  |  |  |  |  |

Example2: Select tag action **MODE=0x02, LEN=0x0C,**

**EPC = 0xE2003411B802011383258566,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **MODE** |
| 7C | FF | FF | 2D | 00 | 0E | 02 |
| **LEN** | **EPC (MSB)** | **--** | **--** | **--** | **--** | **--** |
| 0C | E2 | 00 | 34 | 11 | B8 | 02 |
| **--** | **--** | **--** | **--** | **--** | **EPC (LSB)** | **CHECKSUM** |
| 01 | 13 | 83 | 25 | 85 | 66 | 0xNN |

* + 1. Response

CID1: 2DH

RTN: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FF | FF | 2D | 00 | 00 | 0xNN |

### Get Tx Power Level

* + 1. Command

CID1: 50H

CID2: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 50 | 00 | 00 | 0xNN |

* + 1. Response

CID1: 50H

RTN: 00H

INFO: - PWR (8-bit): Tx Power Level(15~26 dBm) Example: **PWR=0x1A,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **PWR** |
| CC | FF | FF | 50 | 00 | 01 | 1A |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

### Set Tx Power Level

The update registry is valid. Otherwise, it is a temporary update.

* + 1. Command

CID1: 51H

CID2: 00H

INFO: - PWR (8-bit): Tx Power Level(15~26 dBm) Example: **PWR=0x1A,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **PWR** |
| 7C | FF | FF | 51 | 00 | 01 | 1A |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

* + 1. Response

CID1: 51H

RTN: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FF | FF | 51 | 00 | 00 | 0xNN |

### Get Region

* + 1. Command

CID1: 52H

CID2: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 52 | 00 | 00 | 0xNN |

* + 1. Response

CID1: 52H

RTN: 00H

INFO: - Region (8-bit): RF Region

0x01 : China 900M(920.125~924.875MHz),

0x02 : China 800M(840.125~ 844.875MHz),

0x03 : US (902.250~ 926.750MHz),

0x04 : Europe (865.100~ 867.900MHz),

0x05 : Korea (917.100~ 926.900MHz),

Example: **Region=0x01,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **Region** |
| CC | FF | FF | 52 | 00 | 01 | 01 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

### Set Region

The update registry is valid. Otherwise, it is a temporary update.

* + 1. Command

CID1: 53H

CID2: 00H

INFO: - Region (8-bit): RF Region

0x01 : China 900M(920.125~924.875MHz),

0x02 : China 800M(840.125~ 844.875MHz),

0x03 : US (902.250~ 926.750MHz),

0x04 : Europe (865.100~ 867.900MHz),

0x05 : Korea (917.100~ 926.900MHz),

Example: **Region=0x01,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **Region** |
| 7C | FF | FF | 53 | 00 | 01 | 01 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

* + 1. Response

CID1: 53H

RTN: 00H INFO:

* + - * None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FF | FF | 53 | 00 | 00 | 0xNN |

### Get current RF Channel

* + 1. Command

CID1: 54H

CID2: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 54 | 00 | 00 | 0xNN |

* + 1. Response

CID1: 54H

RTN: 00H

INFO: - CH (8-bit): RF Channel

|  |  |  |
| --- | --- | --- |
| China 900M | (0~19), | (Freq\_CH-920.125M)/0.25M |
| China 800M | (0~19), | (Freq\_CH-840.125M)/0.25M |
| US | (0~49), | (Freq\_CH-902.250M)/0.50M |
| Europe | (0~14), | (Freq\_CH-865.100M)/0.20M |
| Korea | (0~49), | (Freq\_CH-917.100M)/0.20M |

Example: Current working channel 920.250MHz, than **CH=0x01,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CH** |
| CC | FF | FF | 54 | 00 | 01 | 01 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

### Set current RF Channel

The update registry is valid. Otherwise, it is a temporary update.

* + 1. Command

CID1: 55H

CID2: 00H

INFO: - CH (8-bit): RF Channel

|  |  |  |
| --- | --- | --- |
| China 900M | (0~19), | (Freq\_CH-920.125M)/0.25M |
| China 800M | (0~19), | (Freq\_CH-840.125M)/0.25M |
| US | (0~49), | (Freq\_CH-902.250M)/0.50M |
| Europe | (0~14), | (Freq\_CH-865.100M)/0.20M |
| Korea | (0~49), | (Freq\_CH-917.100M)/0.20M |

Example: Set Current working channel 920.250MHz,than **CH=0x01,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CH** |
| 7C | FF | FF | 55 | 00 | 01 | 01 |
| **CHECKSUM** |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0xNN |  |  |  |  |  |  |

* + 1. Response

CID1: 55H

RTN: 00H INFO:

* + - * None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FF | FF | 55 | 00 | 00 | 0xNN |

### Get Frequency Hopping Table \*

* + 1. Command

CID1: 56H

CID2: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 56 | 00 | 00 | 0xNN |

* + 1. Response

CID1: 56H

RTN: 00H

INFO: - TS(8-bit): table length (0~50)

- CN (variable): RF channel value

Example: Table Size = 6, channel numbers = 47, 19, 20, 23, 46, 16, then

**TS=0x06,CN=0x2F,0x13,0x14,0x17,0x2E,0x10**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **TS** |
| CC | FF | FF | 56 | 00 | 07 | 06 |
| **CN0** | **CN1** | **CN2** | **CN3** | **CN4** | **CN5** | **CHECKSUM** |
| 2F | 13 | 14 | 17 | 2E | 10 | 0xNN |

### Set Frequency Hopping Table \*

The update registry is valid. Otherwise, it is a temporary update.

* + 1. Command

CID1: 57H

CID2: 00H

INFO: - TS(8-bit): table length (0~50)

- CN (variable): RF channel value

Example: Table Size = 6, channel numbers = 47, 19, 20, 23, 46, 16, than

**TS=0x06,CN=0x2F,0x13,0x14,0x17,0x2E,0x10**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **TS** |
| 7C | FF | FF | 57 | 00 | 07 | 06 |
| **CN0** | **CN1** | **CN2** | **CN3** | **CN4** | **CN5** | **CHECKSUM** |
| 2F | 13 | 14 | 17 | 2E | 10 | 0xNN |

* + 1. Response

CID1: 57H

RTN: 00H INFO:

* + - * None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FF | FF | 57 | 00 | 00 | 0xNN |

### Get Modulation

#### GetCurrent modulation mode. Demodulator parameters include mixer gain, if amplifier gain and signal demodulation threshold.

* + 1. Command

CID1: 58H

CID2: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 58 | 00 | 00 | 0xNN |

* + 1. Response

CID1: 58H

RTN: 00H

INFO: - MODE(8-bit): 0x00-high sensitivity,0x01-High-density,0xFF-Custom

- MG (8-bit): Mixer gain

Mixer Gain table

|  |  |
| --- | --- |
| Type | Mixer\_G(dB) |
| 0x00 | 0 |
| 0x01 | 3 |
| 0x02 | 6 |
| 0x03 | 9 |
| 0x04 | 12 |
| 0x05 | 15 |

- IFG (8-bit): If amplifier gain

IF AMP Gain table

|  |  |
| --- | --- |
| Type | IF\_G(dB) |
| 0x00 | 12 |
| 0x01 | 18 |
| 0x02 | 21 |
| 0x03 | 24 |
| 0x04 | 27 |
| 0x05 | 30 |
| 0x06 | 36 |
| 0x07 | 40 |

- THRD (16-bit): Signal demodulation threshold

The smaller the signal demodulation threshold is, the lower the RSSI of the demodulable tag is, but the more unstable it is, the lower the threshold is, the less demodulable it is; on the contrary, the larger the threshold is, the greater the RSSI of the demodulable tag is, the closer the distance is, the more stable it is. 0x01b0 is the recommended minimum)

Example:

**MODE=0x01,MG=0x01,IFG=0x02,THRD=0x0042**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **MODE** |
| CC | FF | FF | 58 | 00 | 05 | 01 |
| **MG** | **IFG** | **THRD(MSB)** | **THRD(LSB)** | **CHECKSUM** |  |  |
| 01 | 02 | 00 | 42 | 0xNN |  |  |

### Set Modulation

#### Set Current modulation mode. Demodulator parameters include mixer gain, if amplifier gain and signal demodulation threshold.

The update registry is valid. Otherwise, it is a temporary update.

* + 1. Command

CID1: 59H

CID2: 00H

INFO: - MODE(8-bit): 0x00-high sensitivity,0x01-High-density,0xFF-Custom

* MG (8-bit): Mixer gain
* IFG (8-bit): If amplifier gain
* THRD (16-bit): Signal demodulation threshold Example1: Set High-density, than

**MODE=0x01,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **MODE** |
| 7C | FF | FF | 59 | 00 | 01 | 01 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

Example2: Set Custom mode, Mixer gain 9dB,If amplifier gain 36dB,Signal demodulation threshold 0x01B0,than

**MODE=0xFF, MG=0x03,IFG=0x06,THRD=0x01B0**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **MODE** |
| 7C | FF | FF | 59 | 00 | 05 | FF |
| **MG** | **IFG** | **THRD(MSB)** | **THRD(LSB)** | **CHECKSUM** |  |  |
| 03 | 06 | 01 | B0 | 0xNN |  |  |

* + 1. Response

CID1: 59H

RTN: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FF | FF | 59 | 00 | 00 | 0xNN |

### Get Frequency Hopping status

#### Get FH state. Auto FH mode or fixed frequency mode.

* + 1. Command

CID1: 5AH

CID2: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 5A | 00 | 00 | 0xNN |

* + 1. Response

CID1: 5AH

RTN: 00H

INFO: - FM (8-bit): Frequency Mode: 0x00 - fixed frequency, 0x01 - Auto FH

Example: FM**=0x01,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **FM** |
| CC | FF | FF | 5A | 00 | 01 | 01 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

### Set Frequency Hopping status

#### Set FH state. Auto FH mode or fixed frequency mode.

The update registry is valid. Otherwise, it is a temporary update.

* + 1. Command

CID1: 5BH

CID2: 00H

INFO: - FM (8-bit): Frequency Mode: 0x00 - fixed frequency, 0x01 - Auto FH

Example: FM**=0x01,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **FM** |
| 7C | FF | FF | 5B | 00 | 01 | 01 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

* + 1. Response

CID1: 5BH

RTN: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FF | FF | 5B | 00 | 00 | 0xNN |

### Get Base Parameters

* + 1. Command

CID1: 81H

CID2: 32H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 81 | 32 | 00 | 0xNN |

* + 1. Response

CID1: 81H

RTN: 00H

INFO: - OM (8-bit): Output mode, (when the working mode is active, the reader data active output interface)

0x00 - 232(BLE/SPP)

0x01 - 485(USB/HID/WIFI/TCPIP/PDA)

0x02 - WG26

0x03 - WG34

0x04 - WG66

0x05 - WG98 \*(not used temporarily)

* WM (8-bit): Working mode, 0x00- Command,

0x01- Active, 0x02- Passive.

* RT (8-bit): Read Type, 0x02-EPC,

0x03-EPC+OTHER DATA. (Valid by active mode)

* RI (8-bit): reading interval, 2~200,unit is 10ms
* RD (8-bit): Read delay,delayed reading after command interaction,0~255,unit is second. (Valid by active mode)
* WG (32-bit): Including (data offset, output period, pulse width, pulse period). (Valid by WG) Offset (8-bit): (0~14) Byte, Def (0x02)

Interval (8-bit): (0~255) \*10ms, Def (0x1E) Width (8-bit): (0~255) \*10us, Def (0x0A) Period (8-bit): (0~255) \*100us, Def (0x0F)

* SI (16-bit): Same ID output interval,(Valid by active mode)
* BZ (8-bit): buzzer enabled; Disabled (0x00) Enabled (0x01),
* UD (112-bit): Additional data for additional send Tags; (Valid by Read Type = 0x03)

AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filed to 0x00000000;

MB (8-bit): Target tag block selection;

0x00 RFU, 0x01 EPC, 0x02 TID, 0x03 User

SA (8-bit): Target tag data address offset (Word) DL (8-bit): Target tag data length (Word Count). CT (8-bit): Target tag data capture method.

0x00- EPC+TID,

0x01- TID,

0x02- TID+EPC,

0x03- EPC+TIDKEY.

EL(8-bit): Target tag EPC value length. KL(8-bit): Target label KEYS value length. KS(32-bit): Target label KEYS value.

* REV (8-bit): reserve

Note: the red font in information is an extension function, which is not enabled yet. Fill in 0x00 or the default value.

Example:

**OM=0x00, WM=0x01, RT=0x02, RI=0x28, RD=0x0A,**

**Offset = 0x02, Interval=0x1E, Width=0x0A, Period=0x0F, SI=0x0001, BZ=0x01,**

**AP =0x00000000, MB=0x02, SA=0x00, DL=0x06, CT =0x00, EL=0x00, KL=0x00,KS =0x00000000, REV=0x00,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **OM** |
| CC | FF | FF | 81 | 00 | 1B | 01 |
| **WM** | **RT** | **RI** | **RD** | **Offset** | **Interval** | **Width** |
| 01 | 02 | 28 | 0A | 02 | 1E | 0A |
| **Period** | **SI(MSB)** | **SI(LSB)** | **BZ** | **AP(MSB)** | **--** | **--** |
| 0F | 00 | 01 | 01 | 00 | 00 | 00 |
| **AP(LSB)** | **MB** | **SA** | **DL** | **CT** | **EL** | **KL** |
| 00 | 02 | 00 | 06 | 00 | 00 | 00 |
| **KS(MSB)** | **--** | **--** | **KS(LSB)** | **REV** | **CHECKSUM** |  |
| 00 | 00 | 00 | 00 | 00 | 0xNN |  |

### Set Base Parameters

* + 1. Command

CID1: 81H

CID2: 31H

INFO: - OM (8-bit): Output mode, (when the working mode is active, the reader data active output interface)

0x00 - 232(BLE/SPP)

0x01 - 485(USB/HID/WIFI/TCPIP/PDA)

0x02 - WG26

0x03 - WG34

0x04 - WG66

0x05 - WG98 \*(not used temporarily)

* WM (8-bit): Working mode, 0x00- Command,

0x01- Active, 0x02- Passive.

* RT (8-bit): Read Type, 0x02-EPC,

0x03-EPC+OTHER DATA. (Valid by active mode)

* RI (8-bit): reading interval, 2~200,unit is 10ms
* RD (8-bit): Read delay,delayed reading after command interaction,0~255,unit is second. (Valid by active mode)
* WG (32-bit): Including (data offset, output period, pulse width, pulse period). (Valid by WG) Offset (8-bit): (0~14) Byte, Def (0x02)

Interval (8-bit): (0~255) \*10ms, Def (0x1E) Width (8-bit): (0~255) \*10us, Def (0x0A) Period (8-bit): (0~255) \*100us, Def (0x0F)

* SI (16-bit): Same ID output interval,(Valid by active mode)
* BZ (8-bit): buzzer enabled; Disabled (0x00) Enabled (0x01),
* UD (112-bit): Additional data for additional send Tags; (Valid by Read Type = 0x03)

AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filed to 0x00000000;

MB (8-bit): Target tag block selection;

0x00 RFU, 0x01 EPC, 0x02 TID, 0x03 User

SA (8-bit): Target tag data address offset (Word) DL (8-bit): Target tag data length (Word Count). CT (8-bit): Target tag data capture method.

0x00- EPC+TID,

0x01- TID,

0x02- TID+EPC,

0x03- EPC+TIDKEY.

EL(8-bit): Target tag EPC value length.

KL(8-bit): Target label KEYS value length. KS(32-bit): Target label KEYS value.

* REV (8-bit): reserve

Note: the red font in information is an extension function, which is not enabled yet. Fill in 0x00 or the default value.

Example:

**OM=0x00, WM=0x01, RT=0x04, RI=0x28, RD=0x0A,**

**Offset = 0x02, Interval=0x1E, Width=0x0A, Period=0x0F, SI=0x0001, BZ=0x01,**

**AP =0x00000000, MB=0x02, SA=0x00, DL=0x06, CT =0x00, EL=0x00, KL=0x00,KS =0x00000000, REV=0x00,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **OM** |
| 7C | FF | FF | 81 | 31 | 1B | 00 |
| **WM** | **RT** | **RI** | **RD** | **Offset** | **Interval** | **Width** |
| 01 | 04 | 28 | 0A | 02 | 1E | 0A |
| **Period** | **SI(MSB)** | **SI(LSB)** | **BZ** | **AP(MSB)** | **--** | **--** |
| 0F | 00 | 01 | 01 | 00 | 00 | 00 |
| **AP(LSB)** | **MB** | **SA** | **DL** | **CT** | **EL** | **KL** |
| 00 | 02 | 00 | 06 | 00 | 00 | 00 |
| **KS(MSB)** | **--** | **--** | **KS(LSB)** | **REV** | **CHECKSUM** |  |
| 00 | 00 | 00 | 00 | 00 | 0xNN |  |

* + 1. Response

CID1: 5BH

RTN: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FF | FF | 5B | 00 | 00 | 0xNN |

### Get Antenna configuration \*

#### Antenna configuration - multi antenna reader active.

* + 1. Command

CID1: 83H

CID2: 32H

INFO: -None

Example :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 83 | 32 | 00 | 0xNN |

* + 1. Response

CID1: 83H

RTN: 00H

INFO: - CA (8-bit): current ant (1~16)

- EA (16-bit): enabled ant , bit set , for example value is 0x0009,Indicates enable antenna 1 and antenna 4;

Example 1: **CA=0x01,**

**EA=0x0009,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CA** |
| CC | FF | FF | 83 | 00 | 03 | 01 |
| **EA(MSB)** | **EA(LSB)** | **CHECKSUM** |  |  |  |  |
| 00 | 09 | 0xNN |  |  |  |  |

### Set Antenna configuration \*

#### Antenna configuration - multi antenna reader active.

* + 1. Command

CID1: 83H

CID2: 31H

INFO: - MA (8-bit):

-0x00: Set enabled ant,do not save configuration

-0xFF: Set enabled ant,save configuration

-0x01~0x10: change current ant (Ignore EA)

- EA (16-bit): enabled ant , bit set , for example value is 0x0009,Indicates enable antenna 1 and antenna 4;

Example 1: enabled antenna 1,4,do not save configuration

**CA=0x01, EA=0x0009,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **MA** |
| 7C | FF | FF | 83 | 31 | 03 | 00 |
| **EA(MSB)** | **EA(LSB)** | **CHECKSUM** |  |  |  |  |
| 00 | 09 | 0xNN |  |  |  |  |

Example 2: enabled antenna 1,2,3,4, save configuration

**CA=0xff, EA=0x000f,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **MA** |
| 7C | FF | FF | 83 | 31 | 03 | FF |
| **EA(MSB)** | **EA(LSB)** | **CHECKSUM** |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 00 | 0F | 0xNN |  |  |  |  |

Example 3: Change ant to 1

**CA=0x01,**

**EA=null,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **MA** |
| 7C | FF | FF | 83 | 31 | 01 | 01 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

* + 1. Response

CID1: 83H

RTN: 00H

INFO: - None Example :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FE | FF | 83 | 00 | 00 | 0xNN |

### Get Encryption method for Tag

#### Get tag encryption, pairing encryption or CRC verification encryption.

* + 1. Command

CID1: 84H

CID2: 32H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 84 | 32 | 00 | 0xNN |

* + 1. Response

CID1: 84H

RTN: 00H

INFO: - TYPE(8-bit): 0x00-No encryption,0x01-pairing,0x02-CRC

* PM (8-bit): Password high byte,(Pairing encryption is only valid for PM);
* PL (8-bit): Password low byte,(CRC encryption mode PM + PL is effective); Example: **TYPE=0x01, PM=0x01, PL=0x00,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **TYPE** |
| CC | FE | FF | 84 | 00 | 03 | 01 |
| **PM** | **PL** | **CHECKSUM** |  |  |  |  |
| 01 | 00 | 0xNN |  |  |  |  |

### Set Encryption method for Tag

#### Set tag encryption, pairing encryption or CRC verification encryption.

**After set the tag encryption mode, the tags needs to be encrypted (see 4.6), otherwise the card will not be actively identified.**

* + 1. Command

CID1: 84H

CID2: 31H

INFO: - TYPE(8-bit): 0x00-No encryption,0x01-pairing,0x02-CRC

* PM (8-bit): Password high byte,(Pairing encryption is only valid for PM);
* PL (8-bit): Password low byte,(CRC encryption mode PM + PL is effective); Example: **TYPE=0x01, PM=0x01, PL=0x00,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **TYPE** |
| 7C | FF | FF | 84 | 31 | 03 | 01 |
| **PM** | **PL** | **CHECKSUM** |  |  |  |  |
| 01 | 00 | 0xNN |  |  |  |  |

* + 1. Response

CID1: 84H

RTN: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FE | FF | 84 | 00 | 00 | 0xNN |

### Get Protocol Address

#### Get Current communication address. (when there are multiple readers, this address can be used to distinguish Readers)

* + 1. Command

CID1: 85H

CID2: 32H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 85 | 32 | 00 | 0xNN |

* + 1. Response

CID1: 85H

RTN: 00H

INFO: - ADDR(16-bit): protocol address Example: **ADDR=0xFFFE,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **ADDR(MSB)** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CC | FE | FF | 85 | 00 | 02 | FF |
| **ADDR(LSB)** | **CHECKSUM** |  |  |  |  |  |
| FE | 0xNN |  |  |  |  |  |

### Set Protocol Address

* + 1. Command

CID1: 85H

CID2: 31H

INFO: - ADDR(16-bit): protocol address Example: **ADDR=0xFFFE,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **ADDR(MSB)** |
| 7C | FF | FF | 85 | 31 | 02 | FF |
| **ADDR(LSB)** | **CHECKSUM** |  |  |  |  |  |
| FE | 0xNN |  |  |  |  |  |

* + 1. Response

CID1: 85H

RTN: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FE | FF | 85 | 00 | 00 | 0xNN |

### Get UART Baudrate \*

Get UART Baudrate.

### Set UART Baudrate \*

Set UART Baudrate.

### Get Output Mode \*

#### Get the output mode of custom data format in the active reading mode. Working mode is active and effective.

* + 1. Command

CID1: 87H

CID2: 32H

INFO: - None

Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | 87 | 32 | 00 | 0xNN |

* + 1. Response

CID1: 87H

RTN: 00H

INFO: - EN(8-bit): 0x00-Disabled,0x01-Enabled

* TYPE(8-bit): Output Type. (0~4) 0x00- Decimal ,

0x01- Hex , 0x02-Wiegand, 0x03-ASCII,

0x04-BAILING

* SL(8-bit): Display the minimum data length. If the data length is less than this value, fill in 0.
* ENTER(8-bit): Whether the last position of the output data brings Carriage return character.
* ST(8-bit): Address offset value of data to be output (byte count).
* DL(8-bit): Length of data to be output (byte count).
* HL(8-bit): Whether to append fixed data length before output data.(0~20)
* HD(160-bit): Add fixed data value (fixed 20 bytes) in front of output data, fill in data according to HL value, default value is 0.
* EL(8-bit): Whether to append fixed data length after output data.(0~20)
* ED(160-bit): Add fixed data value (fixed 20 bytes) after output data, fill in data according to El value, default value is 0.

Example:

**EN=0x01, TYPE=0x00, SL=0x08, ENTER=0x01, ST=0x02, DL=0x03,**

**HL=0x00, HD=0x0000000000000000000000000000000000000000, EL=0x00, ED=0x0000000000000000000000000000000000000000,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **EN** |
| CC | FF | FF | 87 | 00 | 30 | 01 |
| **TYPE** | **SL** | **ENTER** | **ST** | **DL** | **HL** | **HD(MSB)** |
| 00 | 08 | 01 | 02 | 03 | 00 | 00 |
| **--** | **--** | **--** | **--** | **--** | **--** | **--** |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **--** | **--** | **--** | **--** | **--** | **--** | **--** |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **--** | **--** | **--** | **--** | **HD(LSB)** | **EL** | **ED(MSB)** |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **--** | **--** | **--** | **--** | **--** | **--** | **--** |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **--** | **--** | **--** | **--** | **--** | **--** | **--** |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **--** | **--** | **--** | **--** | **ED(LSB)** | **CHECKSUM** |  |
| 00 | 00 | 00 | 00 | 00 | 0xNN |  |

### Set Output Mode \*

#### Set the current custom data format output mode. Working mode is active and effective.

* + 1. Command

CID1: 87H

CID2: 31H

INFO: - EN(8-bit): 0x00-Disabled,0x01-Enabled

* TYPE(8-bit): Output Type. (0~4) 0x00- Decimal ,

0x01- Hex , 0x02-Wiegand, 0x03-ASCII,

0x04-BAILING

* SL(8-bit): Display the minimum data length. If the data length is less than this value, fill in 0.
* ENTER(8-bit): Whether the last position of the output data brings Carriage return character.
* ST(8-bit): Address offset value of data to be output (byte count).
* DL(8-bit): Length of data to be output (byte count).
* HL(8-bit): Whether to append fixed data length before output data.(0~20)
* HD(160-bit): Add fixed data value (fixed 20 bytes) in front of output data, fill in data according to HL value, default value is 0.
* EL(8-bit): Whether to append fixed data length after output data.(0~20)
* ED(160-bit): Add fixed data value (fixed 20 bytes) after output data, fill in data according to El value, default value is 0.

Example:

**EN=0x01, TYPE=0x01, SL=0x08, ENTER=0x01, ST=0x02, DL=0x03,**

**HL=0x01, HD=0x0200000000000000000000000000000000000000, EL=0x01, ED=0x0300000000000000000000000000000000000000,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **EN** |
| 7C | FF | FF | 87 | 31 | 30 | 01 |
| **TYPE** | **SL** | **ENTER** | **ST** | **DL** | **HL** | **HD(MSB)** |
| 01 | 08 | 01 | 02 | 03 | 01 | 02 |
| **--** | **--** | **--** | **--** | **--** | **--** | **--** |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **--** | **--** | **--** | **--** | **--** | **--** | **--** |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **--** | **--** | **--** | **--** | **HD(LSB)** | **EL** | **ED(MSB)** |
| 00 | 00 | 00 | 00 | 00 | 01 | 03 |
| **--** | **--** | **--** | **--** | **--** | **--** | **--** |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| **--** | **--** | **--** | **--** | **--** | **--** | **--** |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **--** | **--** | **--** | **--** | **ED(LSB)** | **CHECKSUM** |  |
| 00 | 00 | 00 | 00 | 00 | 0xNN |  |

* + 1. Response

CID1: 87H

RTN: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FE | FF | 87 | 00 | 00 | 0xNN |

### Reset System

#### Reset System.

* + 1. Command

CID1: D0H

CID2: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **CHECKSUM** |
| 7C | FF | FF | D0 | 00 | 00 | 0xNN |

* + 1. Response

CID1: D0H

RTN: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FF | FF | D0 | 00 | 00 | 0xNN |

### Update Registry

#### System restart is valid.

* + 1. Command

CID1: D2H

CID2: 00H

INFO: - ARG (8-bit): Store (0x01)

Example: ARG**=0x01,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **ARG** |
| 7C | FF | FF | D2 | 00 | 01 | 01 |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

* + 1. Response

CID1: D2H

RTN: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FF | FF | D2 | 00 | 00 | 0xNN |

### Erase Registry

#### System restart is valid.

* + 1. Command

CID1: D3H

CID2: 00H

INFO: - ARG (8-bit): Erase (0xFF)

Example: ARG**=0xFF,**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **CID2** | **LENGTH** | **ARG** |
| 7C | FF | FF | D3 | 00 | 01 | FF |
| **CHECKSUM** |  |  |  |  |  |  |
| 0xNN |  |  |  |  |  |  |

* + 1. Response

CID1: D3H

RTN: 00H

INFO: - None Example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **HEAD** | **ADDR(LSB)** | **ADDR(MSB)** | **CID1** | **RTN** | **LENGTH** | **CHECKSUM** |
| CC | FF | FF | D3 | 00 | 00 | 0xNN |

### Get GPIO Mode \*

Get GPIO Mode.

### Set GPIO Mode \*

Set GPIO Mode.