Tel: 0086-755-82706913
Fax: 0086-755-82706900
Mail: sales@sysiot.cn
Http://www.sysiot.cn

Add: Room 716, Yiben E-Commerce and Industrial Park, Chaguang Road Nanshan District, Shenzhen, China
Post Code：518055

# SYS-IoT UHF Reader & Module API Communication Protocol V2.8.0

2018-03-19

SHEZHEN SYS IoT CO., LTD.

Tel: 0086-755-82706913
Fax: 0086-755-82706900
Mail: sales@sysiot.cn
Http://www.sysiot.cn
Add: Room 716, Yiben E-Commerce and Industrial Park, Chaguang Road Nanshan District, Shenzhen, China
Post Code：518055

CATALOG

## ISO-18000 READER MESSAGE FORMAT

# 1. Format definition

- Command frame: Host → Reader

| FH | FH | Address | Length | Command | Command | Parameter | Check | Check |
|---|---|---|---|---|---|---|---|---|
| MS | LS | RA | LEN | CMDH | CMDL | Parameter( 0~32) | CRCH | CRCL |

- **Frame Header(FH/FL)**: Adopt double byte frame format, to facilitate quick search frame head，FH(0xAA) and FL(0xAA)；
  FH: Frame High Byte; FL: Frame Low Byte。
- **Address(RA)**: Device address，Address range from 0x00 to 0xFA，And 0xFF is the broadcast address。The default address of Reader is 0x00。Among them, 0xFB ~ 0xFE reserved for alternate address. For the broadcast address, any reader frames responses for the command.
- **Length(LEN)**: In addition to the frame head and address domain, from the length field to check the length of the domain data, including the length of the field；LEN values range from 5 to 37.
- **Command(CMDH/CMDL)**: The command divide into main command(CMDH) and sub-command(CMDL)；Command domain defines the function of the frame to control the reader or operation.
- **Parameter(Parameter)**: Defines the control parameters of command frame. The length of the parameter domain from 0 to 32 bytes；Parameter byte is 0, then the command frame parameters for no command frame.
- **Check(CRC)**: CRC check domain from the frame head to check parameter domain. The CRC - CCITT standards generate polynomial $X^{16} + X^{12} + X^5 + X^0$ See in detail *Appendix A* .

- Response frame: Reader → Host

| FH | FH | Address | Length | Command | Command | Status | Data | Check | Check |
|---|---|---|---|---|---|---|---|---|---|
| MS | LS | RA | LEN | CMDH | CMDL | status | data(0~ 128) | CRCH | CRCL |

- **Frame Header(FH/FL)**: Adopt double byte frame format, to facilitate quick search frame head，FH(0xAA) and FL(0xAA).
  FH: Frame High Byte; FL: Frame Low Byte.
- **Address(RA)**: Device address，Address range from 0x00 to 0xFA，And 0xFF is the broadcast address。The default address of Reader is 0x00。Among them, 0xFB ~ 0xFE reserved for alternate address. For the broadcast address, any reader frames responses for the command.
  For the RA is the broadcast address 0xFF command frame，reader's response frame device address is local set of reader's device address.
- **Length(LEN)**: In addition to the frame head and address domain, from the length field to check the length of the domain data, including the length of the field；LEN values range from 6 to 134.
- **Command(CMDH/CMDL)**: The command divide into main command(CMDH) and sub-command(CMDL)；Command domain defines the function of the frame to control the reader or operation. Response frame back to the host command, in order to inform the host, the response frame is response to which a control function of the command frame or operation.
- **Status(status)**: After the reader receives the host command frame, to obey the orders of the host, and returned to execute commands or results；To let the host know, the execution is the result of the success or failure. Please refer to the back of the description about status.

- **Data(data)**: Defines the reader to obey the orders of the host, the data returned results. Values range from 0 to 128 bytes. Data is 0, the response frame for countless according to the reply.
- **Check(CRC)**: CRC check domain from the frame head to check parameter domain. The CRC - CCITT standards generate polynomial $X^{16} + X^{12} + X^5 + X^0$ See in detail *Appendix A*

# 2. The Overview of the command

**The control commands of the reader and writer are divided into 3 categories**: system control commands, radio frequency parameters commands and Tag access commands. Among them, the system control commands, defined the module information, the device address and reset device, and so on; Radio frequency parameter command, defining the radio frequency RF control parameters of the reader module; The Tag access command defines the operation commands for ISO18000-6C (hereinafter referred to as ISO6C) tags, including reading Tags, writing labels, locking Tags, and so on.

## 2.1 Command List

| SN | command | code | macro definition | instructions |
|---|---|---|---|---|
| System control | | | | |
| 1 | Module information | 0x03 | CMD_GET_MODULE_INFO | Get module information such as hardware version, software version, and manufacturer information |
| 2 | Device Address | 0x05 | CMD_READ_ADDR | 设置和查询模块设备地址 |
| 3 | GPIO Control | 0x0A | CMD_IO_CONTROL | 设置 GPIO 口的控制方向和电平 |
| 4 | Reset | 0x0F | CMD_TEST_RESET | Reset reader |
| radio frequency ( RF) parameters | | | | |
| 5 | Setting RF chip parameters | 0x12 | CMD_SET_MODEM_PARA | Setting RF chip parameters |
| 6 | Read RF chip parameters | 0x13 | CMD_READ_MODEM_PARA | Read RF chip parameters |
| 7 | 模块环境模式 | 0x14 | CMD_SET_READER_ENV_MODE | 设置读写器环境模式：高灵敏度模式和密集读写器模式 |
| 8 | 阻塞信号测试 | 0x15 | CMD_SCAN_JAMMER | 测试射频输入端阻塞信号 |
| 9 | 测试信道 RSSI | 0x16 | CMD_SCAN_RSSI | 测试射频输入端 RSSI 信号大小，用于检测当前环境下有无读写器在工作 |
| 10 | Setting frequency region | 0x30 | CMD_SET_REGION | Setting up frequency work region |
| 11 | Get frequency region | 0x31 | CMD_GET_REGION | Get frequency working region |
| 12 | Set work frequency channel | 0x32 | CMD_SET_RF_CHANNEL | Set work frequency channel |
| 13 | Get work frequency channel | 0x33 | CMD_GET_RF_CHANNEL | Get work frequency channel |
| 14 | Set automatic frequency hopping | 0x37 | CMD_SET_FHSS | Setting automatic frequency hopping mode |
| 15 | Set RF emission power | 0x3B | CMD_SET_POWER | Setting the RF emission power |
| 16 | Query RF emission power | 0x3C | CMD_GET_POWER | Query the RF emission power |
| 17 | 设置连续载波 | 0x3D | CMD_SET_CW | 开启或关闭连续载波 |
| 18 | Set Antenna parameters | 0x3F | CMD_ANT | Set the working parameters of the antenna |
| Tag access | | | | |

| | | | | |
|---|---|---|---|---|
| 19 | Get Inventory mode | 0x40 | CMD_GET_INV_MODE | Get inventory mode |
| 20 | Set Inventory mode | 0x41 | CMD_SET_INV_MODE | Set inventory mode |
| 21 | Get Select parameters | 0x42 | CMD_GET_SELECT_PARA | Get Select parameters |
| 22 | Set Select parameters | 0x43 | CMD_SET_SELECT_PARA | Set Select parameters |
| 23 | Get Query parameters | 0x44 | CMD_GET_QUERY_PARA | Get Query parameters |
| 24 | Set Query parameters | 0x45 | CMD_SET_QUERY_PARA | Set Query parameters |
| 25 | Stop multi-Tags inventory | 0xC0 | CMD_STOP_MULTI | Stop multiple Tags inventory by multiple antennas |
| 26 | Start multi-Tags inventory | 0xC1 | CMD_MULTI_ID | Start multiple Tags inventory by multiple antennas |
| 27 | Write data to Tag | 0xC2 | CMD_WRITE_DATA | Write data to storage area of Tag |
| 28 | Lock Tag | 0xC3 | CMD_LOCK_UNLOCK | Lock Tag data storage area |
| 29 | Kill Tag | 0xC4 | CMD_KILL | Kill Tag |
| 30 | Single Tag inventory | 0xC8 | CMD_SINGLE_ID | Single Tag inventory |
| 31 | Read data from Tag | 0xC9 | CMD_READ_DATA | Read data from storage area of Tag |
| 32 | Start Read Sensor Tags | 0xCA | CMD_SENSORTAG_READ | Stop Read Sensor Tags |
| 33 | Stop Read Sensor Tags | 0xCB | CMD_SENSORTAG_STOP | Stop Read Sensor Tags |

# 3. State definition

The reader executes the command from the host, and returns the execution state to the host。

```
FAIL_OK = 0x00                          // Response OK, No error.
FAILE_RESPONSE_TIMEOUT =0x01            // No response and timeout.
FAIL_INVALID_PARA = 0x0E                // Input Parameter is not right.
FAIL_INVENTORY_TAG_TIMEOUT = 0x15       //Inventory tag timeout
FAIL_INVALID_CMD = 0x17                 // Command undefined
FAIL_FHSS_FAIL = 0x20                   // Hopping Frequency is Error.
FAIL_ANT_NOT_AVAILABLE = 0x21           // Antenna port is not available
FAIL_ACCESS_PWD_ERROR = 0x16            // Access Password is Error.
FAIL_READ_MEMORY_NO_TAG = 0x09          //No tag red while reading a memory on a tag
FAIL_READ_ERROR_CODE_BASE = 0xA0
FAIL_WRITE_MEMORY_NO_TAG = 0x10
FAIL_WRITE_ERROR_CODE_BASE = 0xB0
FAIL_LOCK_NO_TAG = 0x13
FAIL_LOCK_ERROR_CODE_BASE = 0xC0,
FAIL_KILL_NO_TAG = 0x12,
FAIL_KILL_ERROR_CODE_BASE = 0xD0
FAIL_WATCHDOG_OVERFLOW = 0x05
FAIL_SUBCMD_UNDEF =0xF1                 //API Sub-Command undefined
FAIL_MAINCMD_UNDEF =0xF2                //API Main Command undefined
FAIL_UNDEF =0xFF                        //Fail Information Undefined
```

Tag error code

```
ERROR_CODE_INSUFFICIENT_POWER = 11;
```

```
ERROR_CODE_MEM_LOCKED = 4;
ERROR_CODE_MEM_OVERRUN = 3;
ERROR_CODE_NON_SPEC_ERROR = 15;
ERROR_CODE_OTHER_ERROR = 0;
```

# 4. Command Details

Note: all defined multi byte data are MSB before and after LSB. *B represents Byte.

## 4.1 System Control Commands

### 4.1.1 Get Module information (0x03)

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x03 | 0x00 | 0x00- hardware version<br>0x01- software version | Get module information such as hardware version, software version, and manufacturer information. |
| 0x03 | 0x01 | | Get the main firmware version information. |

*B represents Byte

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x03 | 0x00 | FAIL_OK<br>FAIL_INVALID_PARA | Return to hardware / software version information, ASCII code format |
| 0x03 | 0x01 | FAIL_OK | Return to the main firmware version information, HEX code format |

>Host:AA AA FF 06 03 00 00 84 C3
>Reader: AA AA FF 17 03 00 00 00 4D 30 39 30 33 20 33 30 64 42 6D 20 56 32 2E 38 41 27
>Host: AA AA FF 06 03 00 01 94 E2
>Reader: AA AA FF 0D 03 00 00 01 56 53 32 2E 32 32 57 CF
>Host: AA AA FF 05 03 01 E0 D1
>Reader: AA AA FF 0A 03 01 00 92 02 01 0A EB 95

### 4.1.2 Address of Read Module (0x05)

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x05 | 0x00 | New Address(1B) | Setting new address of the device |
| 0x05 | 0x01 | XX(1B) (Meaningless) | Read the current address of the device |

*B represents Byte

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|

| CMDH | CMDL | Status | Data |
|---|---|---|---|
| 0x05 | 0x00 | FAIL_OK<br>FAIL_INVALID_PARA | new address of the reader module |
| 0x05 | 0x01 | FAIL_OK | the current address of the reader module (1B) |

Note:If you do not know the device address of the current reader module, you can use the host to connect the reader module separately, and then use the broadcast address 0xFF to query (read).

Query the current address of the reader module:

>Host:AA AA FF 06 05 01 FF 1B A2 //

>Reader: AA AA FF 07 05 01 00 01 B8 D5 // the current address of the device is 0x01;

### 4.1.3 GPIO Port Control (0x0A)

Set the control direction and level of GPIO port, or read the input state of GPIO.

The reader module has four GPIO ports. The default state of the power on is output mode, and the output is low level.

| **Main** | **Sub** | **Parameter** | | | |
|---|---|---|---|---|---|
| CMDH | CMDL | Parameter | | | Functional Specifications |
| 0x0A | 0x00 | IOM(1B) | IOP(1B) | IOC(1B) | Set the control direction and level of GPIO port |

| | Parameter | Byte | Specifications |
|---|---|---|---|
| 1 | IOM | 1B | Operation mode selection:<br>0x00: Set IO direction;<br>0x01: Set IO level;<br>0x02: Read the IO level.<br>The pins to be operated are specified in the IOP parameter. |
| 2 | IOP | 1B | The parameter range is 0x01~0x04, corresponding to the port GPIO1~GPIO4 to operate. |
| 3 | IOC | 1B | The parameter value is 0x00 or 0x01.<br>IOM / IOC / —<br>0x00 / 0x00 / IO Configured as input mode<br>0x00 / 0x01 / IO Configured as output mode<br>0x01 / 0x00 / Set IO output to low level.<br>0x01 / 0x01 / Set IO output to high level.<br>This parameter is meaningless when IOM is 0x02. |

Response frame

| **Main** | **Sub** | **Status** | **Data** | | |
|---|---|---|---|---|---|
| CMDH | CMDL | Status | Data | | |
| 0x0A | 0x00 | FAIL_OK；<br>FAIL_INVALID_PARA； | IOM(1B) | IOP(1B) | IOC(1B) |

| | Parameter | Byte | Specifications |
|---|---|---|---|
| 1 | IOM | 1B | Operation mode selection:<br>0x00: Set IO direction; |

<table>
<tr><td></td><td></td><td></td><td>0x01: Set IO level;<br>0x02: Read the IO level.<br>The pins to be operated are specified in the IOP parameter.</td></tr>
<tr><td>2</td><td>IOP</td><td>1B</td><td>The parameter range is 0x01~0x04, corresponding to the port GPIO1~GPIO4 to operate.</td></tr>
<tr><td>3</td><td>IOC</td><td>1B</td><td>The parameter value is 0x00 or 0x01.
<table>
<tr><td>IOM</td><td>IOC</td><td></td></tr>
<tr><td>0x00</td><td>0x00</td><td>IO Configured as input mode</td></tr>
<tr><td>0x00</td><td>0x01</td><td>IO Configured as output mode</td></tr>
<tr><td>0x01</td><td>0x00</td><td>Set IO output to low level.</td></tr>
<tr><td>0x01</td><td>0x01</td><td>Set IO output to high level.</td></tr>
<tr><td>0x02</td><td>0x00</td><td>The corresponding port is low level.</td></tr>
<tr><td>0x02</td><td>0x01</td><td>The corresponding port is high level.</td></tr>
</table></td></tr>
</table>

>Host: AA AA FF 08 0A 00 00 03 01 53 DE
>Reader: AA AA FF 09 0A 00 00 00 03 01 23 E4
>Host: AA AA FF 08 0A 00 01 03 01 64 EE
>Reader: AA AA FF 09 0A 00 00 01 03 01 14 D4
>Host: AA AA FF 08 0A 00 02 03 00 2D 9F
>Reader: AA AA FF 09 0A 00 00 02 03 01 4D 84

### 4.1.4 Reset Reader Module (0x0F)

Reset reader module

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x0F | 0x00 | | Reset reader module |

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x0F | 0x00 | FAIL_OK | 无 |

>Host:AA AA FF 05 0F 00 B5 9D
>Reader: AA AA FF 06 0F 00 00 F1 A2

### 4.1.4 Setting RF Chip Parameters (0x12)

Set up the current reader RF chip demodulator parameters. The demodulator parameters include Mixer gain, medium frequency amplifier (IF AMP) gain and signal demodulation threshold. (Usually, users do not need to change this parameter).

| Main | Sub | Parameter |
|---|---|---|

| CMDH | CMDL | Mixer_G | IF_G | Thrd(MSB) | Thrd(LSB) | |
|---|---|---|---|---|---|---|
| 0x12 | 0x00 | 1B | 1B | 1B | 1B | 设置射频芯片解调器参数 |

*B represents Byte

Mixer_G: 0x03 (Mixer gain is 9dB)

| |
|---|
| 000: Gain = 0dB |
| 001: Gain = 3dB |
| 010: Gain = 6dB |
| 011: Gain = 9dB |
| 100: Gain = 12dB |
| 101: Gain = 15dB |
| 110: Gain = 16dB |
| 111: Reserved |

IF_G: 0x06 (medium frequency amplifier (IF AMP) gain is 36dB)

| |
|---|
| 000: Gain = 12dB |
| 001: Gain = 18dB |
| 010: Gain = 21dB |
| 011: Gain = 24dB |
| 100: Gain = 27dB |
| 101: Gain = 30dB |
| 110: Gain = 36dB |
| 111: Gain = 40dB |

Thrd: 0x01B0 (Signal demodulation threshold)

Note: the smaller the signal demodulation threshold is, the closer the label distance can be demodulated, the lower the return RSSI value, and the more unstable it is, the lower the certain value cannot be demodulated at all. On the contrary, the larger the threshold is, the larger RSSI of the demodulated label return signal is, and the closer the label distance is, the more stable it is.

0x01B0 is the recommended minimum value.

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x12 | 0x00 | FAIL_OK | 0x00 |

>Host: AA AA FF 09 12 00 04 06 01 20 1E 52

>Reader: AA AA FF 07 12 00 00 00 D5 4E

#### 4.1.5 Read RF Chip Parameters (0x13)

Read the current reader RF chip demodulator parameters. The demodulator parameters include Mixer gain, medium frequency amplifier (IF AMP) gain and signal demodulation threshold. (Usually, users do not need to change this parameter).

| Main | Sub | Parameter |
|---|---|---|

| CMDH | CMDL | Parameter | Functional Specifications |
|---|---|---|---|
| 0x13 | 0x00 | | Read RF chip demodulator parameters of the current reader |

Response frame

| **Main** | **Sub** | **Status** | **Data** | |
|---|---|---|---|---|
| CMDH | CMDL | Status | Data | |
| 0x13 | 0x00 | FAIL_OK | Mixer_G | IF_G |
| | | | 1B | 1B |
| **Data** | | | | |
| Thrd(MSB) | Thrd(LSB) | | | |
| 1B | 1B | | | |

>Host: AA AA FF 05 13 00 F3 83
>Reader: AA AA FF 0A 13 00 00 04 06 01 20 21 A0

### 4.1.6 Setting Environment Mode (0x14)

Setting Environment mode of the reader module.

| **Main** | **Sub** | **Parameter** | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x14 | 0x00 | 0x00- high sensitivity mode<br>0x01- multiple reader mode | Setting Environment mode of the reader module |

Response frame

| **Main** | **Sub** | **Status** | **Data** |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x14 | 0x00 | FAIL_OK | 0x00 |

>Host: AA AA FF 06 14 00 00 42 30
>Reader: AA AA FF 07 14 00 00 00 F2 D7
>Host: AA AA FF 06 14 00 01 52 11
>Reader: AA AA FF 07 14 00 00 00 F2 D7

#### 4.1.7 Test Scan Jammer (0x15)

The blocking signal of the radio frequency input endpoint (Scan Jammer) is tested to detect the blocking signal size of each channel of the reader antenna in the current area.

| **Main** | **Sub** | **Parameter** | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x15 | 0x00 | | Test Scan Jammer |

Response frame

| Main | Sub | Status | Data | | |
|---|---|---|---|---|---|
| CMDH | CMDL | Status | Data | | |
| 0x15 | 0x00 | FAIL_OK | CH_L | CH_H | JMR |
| | | | 1B | 1B | (CH_H-CH_L+1) |

If the test RF input blocking signal (Scan Jammer) is correctly executed over a total of 20 channels at 900MHz in China, the response frame is:
Test start channel CH_L: such as 0x00 (test start channel Index is 0).
Test end channel CH_H: such as 0x13 (end of test channel Index is 19).
The channel blocking signal is JMR: 0xF2F1F0EFECEAE8EAECEEF0F1F5F5F5F6F5F5F5F5 (where 0xF2 is -14dBm).

>Host: AA AA FF 05 15 00 59 25
>Reader: AA AA FF 1C 15 00 00 00 13 06 06 06 07 07 07 07 07 07 07 07 07 06 06 06 06 06 06 06 06 07 58

### 4.1.8 Test RSSI RSSI (0x16)

Test the size of the RF input signal (RSSI) to detect whether the reader is working in the current environment.

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x16 | 0x00 | | Test RSSI |

Response frame

| Main | Sub | Status | Data | | |
|---|---|---|---|---|---|
| CMDH | CMDL | Status | Data | | |
| 0x16 | 0x00 | FAIL_OK | CH_L | CH_H | JMR |
| | | | 1B | 1B | (CH_H-CH_L+1) |

If the test RF input blocking signal (RSSI) is correctly executed over a total of 20 channels at 900MHz in China, the response frame is:
Test start channel CH_L: such as 0x00 (test start channel Index is 0).
Test end channel CH_H:, such as 0x13 (end of test channel Index is 19).
The channel signal RSSI: 0xF2F1F0EFECEAE8EAECEEF0F1F5F5F5F6F5F5F5F5 (where 0xF2 is -14dBm).
The value of signal strength is complementary code. The calculation method is RSSI = - (~V+1).
Rssi (DBM) = - (~0xF2+1) = - (0x0D + 1) =-14 (dBm)

>Host: AA AA FF 05 16 00 0C 76
>Reader: AA AA FF 1C 16 00 00 00 13 BA BA BA BA BA BA BA BA BA BA BA BA BA BA BA BA BA BA BA BA BA BA D5 2B

### 4.1.9 Setting Frequency Region (0x30)

Setting up frequency work region.

| Main | Sub | Parameter |
|---|---|---|

| CMDH | CMDL | Parameter | Functional Specifications |
|---|---|---|---|
| 0x30 | 0x00 | 1B | Setting up frequency work region |

| number | Region | sort |
|---|---|---|
| 1 | China 2 920.125~924.875MHz | 01 |
| 2 | China 1 840.125~844.875MHz | 04 |
| 3 | FCC 902~928MHz | 02 |
| 4 | Europe 865.1-867.9MHz | 03 |
| 5 | Korea 917.1~922.9MHz | 06 |
| | | |

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x30 | 0x00 | FAIL_OK | 0x00 |

>Host: AA AA FF 06 30 00 01 08 17
>Reader: AA AA FF 07 30 00 00 00 0F 68

#### 4.1.10 Get Frequency Region (0x31)

Get frequency working region

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x31 | 0x00 | | Get frequency working region |

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x13 | 0x00 | FAIL_OK | 1B |

*B represents Byte

>Host: AA AA FF 05 31 00 93 07
>Reader: AA AA FF 07 31 00 00 01 69 FD

#### 4.1.11 Set Work Frequency Channel (0x32)

Set work frequency channel

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x32 | 0x00 | Channel Index 1B | Set work frequency channel |

Channel Index: Work Frequency Channel Index value.

| Country | Freq_CH |
|---|---|
| China1 | 840.125MHz~844.875MHz |
| China2 | 920.125MHz~924.875MHz |
| FCC | 902.25MHz~927.75MHz |
| Europe | 865.1MHz~867.9MHz |
| Korea | 917.1MHz~923.3MHz |

Freq_CH as channel frequency

China1 channel parameter calculation formula:

CH_Index = (Freq_CH-840.125M)/0.25M

Freq_CH = CH_Index*0.25M + 840.125M

China2 channel parameter calculation formula:

CH_Index = (Freq_CH-920.125M)/0.25M

Freq_CH = CH_Index*0.25M + 920.125M

FCC channel parameter calculation formula:

CH_Index = (Freq_CH-902.25M)/0.5M

Freq_CH = CH_Index*0.5M + 902.25M

Europe channel parameter calculation formula:

CH_Index = (Freq_CH-865.1M)/0.2M

Freq_CH = CH_Index*0.2M + 865.1M

Korea channel parameter calculation formula:

CH_Index = (Freq_CH-917.1M)/0.2M

Freq_CH = CH_Index*0.2M + 917.1M

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x32 | 0x00 | FAIL_OK | 0x00 |

>Host: AA AA FF 06 32 00 00 76 56

>Reader: AA AA FF 07 32 00 00 00 E2 00

>Host: AA AA FF 06 32 00 08 F7 5E

>Reader: AA AA FF 07 32 00 00 00 E2 00

### 4.1.12 Get Work Frequency Channel (0x33)

Set work frequency channel

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x33 | 0x00 | | Set work frequency channel |

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x33 | 0x00 | FAIL_OK | Channel Index 1B |

Channel Index: Work Frequency Channel Index value.

| Country | Freq_CH |
|---|---|
| China1 | 840.125MHz~844.875MHz |
| China2 | 920.125MHz~924.875MHz |
| FCC | 902.25MHz~927.75MHz |
| Europe | 865.1MHz~867.9MHz |
| Korea | 917.1MHz~923.3MHz |

Freq_CH as channel frequency

China1 channel parameter calculation formula:

CH_Index = (Freq_CH-840.125M)/0.25M

Freq_CH = CH_Index*0.25M + 840.125M

China2 channel parameter calculation formula:

CH_Index = (Freq_CH-920.125M)/0.25M

Freq_CH = CH_Index*0.25M + 920.125M

FCC channel parameter calculation formula:

CH_Index = (Freq_CH-902.25M)/0.5M

Freq_CH = CH_Index*0.5M + 902.25M

Europe channel parameter calculation formula:

CH_Index = (Freq_CH-865.1M)/0.2M

Freq_CH = CH_Index*0.2M + 865.1M

Korea channel parameter calculation formula:

CH_Index = (Freq_CH-917.1M)/0.2M

Freq_CH = CH_Index*0.2M + 917.1M

>Host: AA AA FF 05 33 00 F5 65

>Reader: AA AA FF 07 33 00 00 08 15 BC

### 4.1.13 Set Automatic Frequency Hopping Mode (0x37)

Setting automatic frequency hopping mode is turn on or turn off, when the read-writer module is power on, it is on the automatic frequency hopping mode.

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x37 | 0x00 | 0x00 – turn on automatic mode<br>0xFF – turn off automatic mode | Set automatic frequency hopping mode is turn on or turn off |

**FHSS** (Frequency-Hopping Spread Spectrum) is a kind of interference avoidance communication mode based on pseudorandom code for frequency hopping communication

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x37 | 0x00 | FAIL_OK | Work Frequency Channel Index value 1B |

>Host: AA AA FF 06 37 00 00 9D A6 //turn off automatic mode
>Reader: AA AA FF 07 37 00 00 00 5E 45
>Host: AA AA FF 06 37 00 FF 83 56 //turn on automatic mode
>Reader: AA AA FF 07 37 00 00 00 5E 45

### 4.1.14 Set RF Emission Power Capacity (0x3B)

Set RF emission power capacity

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x3B | 0x00 | Power capacity value (2B) | Set RF emission power capacity |

e.g. Power capacity value: 0x0BB8 (The current power capacity is decimal 3000, that is 30dBm)

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x3B | 0x00 | FAIL_OK | 0x00 |

>Host: AA AA FF 07 3B 00 0B B8 EB 5E //30dBm (3000=0x0BB8)
>Reader: AA AA FF 07 3B 00 00 00 11 77

### 4.1.15 Get RF Emission Power Capacity (0x3C)

Get RF emission power capacity

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x3C | 0x00 | | Get RF emission power capacity |

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x3C | 0x00 | FAIL_OK | Power capacity value 2B |

*2B represents 2Bytes

>Host: AA AA FF 05 3C 00 E5 5B
>Reader: AA AA FF 08 3C 00 00 0B B8 2D EE

### 4.1.16 Setting continuous carrier signal (0x3D)

Setting a continuous carrier signal is transmitted or the continuous carrier signal is switched off. Used for testing, and no label operation instruction.

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x3D | 0x00 | 0x00 - switched off<br>0xFF - switched on | switched on/off a continuous carrier signal |

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x3D | 0x00 | FAIL_OK | 0x00 |

>Host: AA AA FF 06 3D 00 FF 44 97 //switched on
>Reader: AA AA FF 07 3D 00 00 00 36 EE
>Host: AA AA FF 06 3D 00 00 5A 67 //switched off
>Reader: AA AA FF 07 3D 00 00 00 36 EE

### 4.1.17 Antenna parameter control (0x3F)

Setting or getting the working parameters of the antenna, the number of the antenna ports, if the antenna port is working, the power value of the antenna port, and the frequency of the automatic polling of the antenna.
Antenna parameter control, only valid for multi tags inventory command (0xC1)! When using the multi tag inventory command 0xC1, the antenna port is automatically switched and multi tags inventory according to the control command (0x3F).

#### 4.1.17.1 Set the working parameters of the antenna (0x3F01)

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x3F | 0x00 | | Set the working parameters of the antenna |

| Parameter | | | | | | | |
|---|---|---|---|---|---|---|---|
| ANTQNT | ANTEN | ANTPOL | ANTPWR1 | ANTPLTN1 | ..... | ANTPWRn | ANTPLTNn |
| 1B | 4B | 1B | 2B | 2B | .... | 2B | 2B |

- ANTQNT: Antenna Quantity，1，4，8，16，32,(Need reader / writer module hardware support) ANTQNT=n，（1,4,8...）。(ANTPWR1, ANTPLN1),…, (ANTPWRn, ANTPLNn);

- ANTEN: Antenna Port Bit Enable, Need reader / writer module hardware support to work to enable.
  ANTEN[0]: bit 7~Ant8, bit6~Ant7, ....., bit1~Ant2, bit0~Ant1;
  ANTEN[1]: bit 15~Ant16, bit14~Ant15, ....., bit9~Ant10, bit8~Ant9;
  ANTEN[2]: 0x00, (Reserved).
  ANTEN[3]: 0x00, (Reserved).

- ANTPOL: Antenna Polling Enable, Antenna port automatic polling; 0, close; 1, open. When the multi tag inventory is started, the antenna is automatically polled in order to be able to work (ANTEN) of the antenna port (only on the multi tag inventory).
- ANTPWR: The power value of the antenna port. For each antenna port, you can set different power values, such as 30dBm -- > 3000 -- > 0x0BB8; When multiple tags inventory, and when the automatic polling is switched to an antenna port, at first the reader module sets the power of the antenna port according to the parameter (ANTPWR), and then go on multi label inventory.

- ANTPLTN: The number of multiple label inventory (inventory EPC) for each antenna port. If the current antenna port inventory is completed, automatically switch to the next already enabled antenna port to carry out multi tag inventory operation.

When start the multi tag inventory, the reader inventory tags in turn at the enable antenna port polling ANTPLTN times, Non enabled antenna ports are not accessed. The reader poll to complete the specified number of times at one antenna port, automatically switch to the next antenna port, and continue to poll the number of times is specified.

Please refer to the "multi tag inventory (0xC1)", and InvNumber defines the number of polling all the antennas at the time of multi tag inventory. If InvNumber =0 is showing an infinite inventory; if InvNumber is in 1~65535, after the inventory tag completes the number of specified times, the reader automatically terminates the multi label inventory, and automatically respond to the 0xC0 command to notify the Host to end the inventory

One of InvNumber is the reader automatically polls all the enabling antenna ports once, In this process, each antenna poll tags the corresponding specified ANTPLTN times.

For example, ANTPOL=1, ANTEN[0]=0x0B (ANT1, ANT2, ANT4 for the work of the antenna port, ANT3, ANTPLTN1=5, ANTPLTN2=9 does not work), ANTPLTN4=15, start the multi tag inventory(0xC1 command), reader module in turn enable polling antenna port: ANT1 5 times, ANT2 9 times, ANT4 15 times, is complete an inventory, plus 1 InvNumber counter. In turn, when the count of the InvNumber counter reaches InvNumber, it ends the multi tags inventory and sends the end response frame 0xC0 to the host.

**Note: 1. ANT1 is the highest priority antenna port, and in the system the antenna port ANT1 must be able to use!**

**2. Antenna port must have an external impedance of 50 Omega, and an antenna in Bobbi (VSWR) less than 1.3 (working band)**

**3. The antenna port is not connected to the antenna, and the antenna port is enabled when the antenna parameters are set. When the multi label inventory command is started, it is still accessible to the unattached antenna but set to the enabling antenna port.，It will cause the RF signal power of the module to not be effectively transmitted and reflected back into the module，This long-term work will damage the RF power amplifier inside the module！**

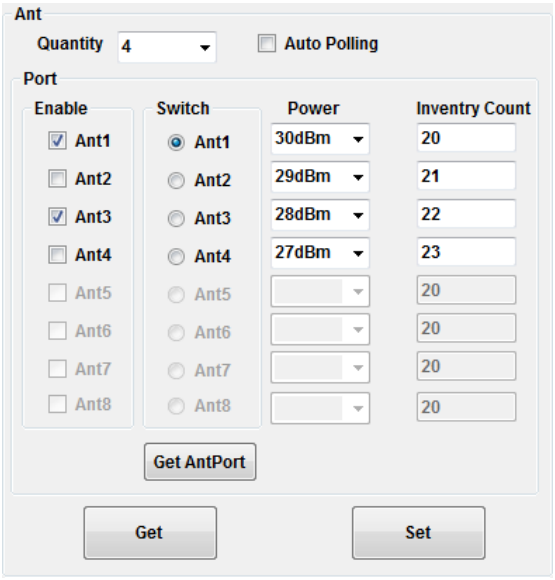
Response frame

| Main | Sub | Status |
|---|---|---|
| CMDH | CMDL | Status |
| 0x3F | 0x00 | FAIL_OK;<br>FAIL_ANT_NOT_AVAILABLE; |

| Data | | | | | | | |
|---|---|---|---|---|---|---|---|
| ANTQN | ANTEN | ANTPOL | ANTPW | ANTPLN1 | ..... | ANTPWR | ANTPL |

| T | | | R1 | | | n | Nn |
|---|---|---|---|---|---|---|---|
| 1B | 4B | 1B | 2B | 2B | .... | 2B | 2B |

>Host: AA AA FF 1B 3F 00 04 05 00 00 00 00 0B B8 00 14 0B 54 00 15 0A F0 00 16 0A 8C 00 17 2F 74
>Reader: AA AA FF 1C 3F 00 00 04 05 00 00 00 00 0B B8 00 14 0B 54 00 15 0A F0 00 16 0A 8C 00 17 73 FD




>Host: AA AA FF 1B 3F 00 04 05 00 00 00 01 0B B8 00 14 0B 54 00 15 0A F0 00 16 0A 8C 00 17 3F 96
>Reader: AA AA FF 1C 3F 00 00 04 05 00 00 00 01 0B B8 00 14 0B 54 00 15 0A F0 00 16 0A 8C 00 17 63 1F
>Host: AA AA FF 0F 3F 00 01 01 00 00 00 01 0B B8 00 14 D4 0F
>Reader: AA AA FF 10 3F 00 00 01 01 00 00 00 01 0B B8 00 14 49 56

### 4.1.17.2 Get the working parameters of the antenna (0x3F01)

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x3F | 0x01 | | Get the working parameters of the antenna |

Response frame

| Main | Sub | Status |
|---|---|---|
| CMDH | CMDL | Status |
| 0x3F | 0x01 | FAIL_OK;<br>FAIL_ANT_NOT_AVAILABLE; |

| Data | | | | | | | |
|---|---|---|---|---|---|---|---|
| ANTQNT | ANTEN | ANTPOL | ANTPWR1 | ANTPLN1 | ..... | ANTPWRn | ANTPLNn |
| 1B | 4B | 1B | 2B | 2B | .... | 2B | 2B |

Host: AA AA FF 05 3F 01 A0 29
Reader: AA AA FF 10 3F 01 00 01 01 00 00 00 00 0B B8 00 05 E2 62

### 4.1.17.3 Set the antenna port of the current work (0x3F02)

Switch the antenna port of the current work to the specified antenna port.

Note: if the antenna port does not work(Status is Disable), then it will switch is not successful。

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x3F | 0x02 | ANTNO 1B | Set the antenna port of the current work |

Response frame

| Main | Sub | Status | Data | |
|---|---|---|---|---|
| CMDH | CMDL | Status | Data | |
| 0x3F | 0x02<br>0x03 | FAIL_OK<br>FAIL_ANT_NOT_AVAILABLE | ANTNO(1B) | RFSW(1B) |

ANTNO: Antenna Port Number, 0-->ANT1, 1-->ANT2, 2-->ANT3 …
RFSW: RF Switch Type, Reserved.

#### 4.1.17.4 Get the antenna port of the current work (0x3F03)

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x3F | 0x03 | | Get the antenna port of the current work |

Response frame

| Main | Sub | Status | Data | |
|---|---|---|---|---|
| CMDH | CMDL | Status | Data | |
| 0x3F | 0x02<br>0x03 | FAIL_OK<br>FAIL_ANT_NOT_AVAILABLE | ANTNO(1B) | RFSW(1B) |

ANTNO: Antenna Port Number, 0-->ANT1, 1-->ANT2, 2-->ANT3...
RFSW: RF Switch Type, Reserved.
>Host: AA AA FF 06 3F 02 02 72 27 // Switch to the antenna port 3（Ant3）
>Reader: AA AA FF 08 3F 02 00 02 02 92 5D
>Host: AA AA FF 05 3F 03 80 6B // Get the antenna port of the current work
>Reader: AA AA FF 08 3F 03 00 02 02 E4 E9
>Host: AA AA FF 06 3F 02 03 62 06 // Switch to the antenna port 4（Ant4），ANT4 is Not enable.
>Reader: AA AA FF 06 3F 02 21 66 26

## 4.2 Tag Access Commands

### 4.2.1 Get Inventory mode (0x40)

Get inventory mode, in general, the user may not have to care or set the inventory mode.

| Main | Sub | Parameter |
|---|---|---|

| CMDH | CMDL | Parameter | Functional Specifications |
|---|---|---|---|
| 0x40 | 0x00 | | Get inventory mode |

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x40 | 0x00 | FAIL_OK | Inventory mode 1B |

Inventory mode description: whether the Tags Select command is required to be sent when Tags inventory.
Inventory mode value:
0x00: The Select command is sent in advance to select a specific tag before all the operation of the tag.
0x01: No Select command is not sent before the tag operation.
0x02: Only send Select command before tag operation except for inventory command, such as Read, Write, Lock, Kill before selecting specific tags through Select command.

>Host: AA AA FF 05 40 00 A8 6F
>Reader: AA AA FF 07 40 00 00 FF 53 ED

### 4.2.2 Set Inventory Mode (0x41)

Set inventory mode, in general, the user may not have to care or set the inventory mode.

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x41 | 0x00 | inventory mode 1B | Set inventory mode |

Response frame

| Main | Sub | Status | Data |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x41 | 0x00 | FAIL_OK | 0x00 |

>Host: AA AA FF 06 41 00 01 E7 2F
>Reader: AA AA FF 07 41 00 00 00 3B A9

### 4.2.3 Get Select Parameters (0x42)

Get Select command parameters.

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x42 | 0x00 | | Get Select command parameters |

Response frame

| Main | Sub | Status |
|---|---|---|

| CMDH | CMDL | Status | |
|---|---|---|---|
| 0x42 | 0x00 | FAIL_OK | |

| Data | | | | |
|---|---|---|---|---|
| SelParam | Ptr | MaskLen | Truncate | Mask (MSB~LSB) |
| 1B | 4B | 1B | 1B | (0~MaskLen/8) |

SelParam: 0x01 (Target: 3’b000, Action: 3’b000, MemBank: 2’b01)

Target: The inventory session mode of the tag (S0, S1, S2, S3, SL), the default selection S0;

Action: The tag inventory processing marks the flip action. Default selection “000”;

MemBank: The tag memory area.

| | | |
|---|---|---|
| Bank 11 | User | b11 = 0x03 |
| Bank 10 | TID | b10 = 0x02 |
| Bank 01 | EPC | b01 = 0x01 |
| Bank 00 | RESERVED | b00 = 0x00 |

Ptr: The start address of the tag access, the length of the bit unit, not word unit.

MaskLen: The length of the mask data, e.g. the length of the mask data 0x60(6 words，96bits)

Truncate: 0x00(0x00 Disable truncation，0x80 Enable truncation)

Mask: mask data, 0x30751FEB705C5904E3D50D70 （EPC）

Reference documents《EPC™ Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface Protocol for Communications at 860 MHz－960 MHz Version 2.0.0 Ratified》

If MemBank=0x01, select tags to require the start address Ptr=0x20 (bit as a unit, non-word) from the EPC storage area.

>Host: AA AA FF 05 42 00 CE 0D

>Reader: AA AA FF 0D 42 00 00 01 00 00 00 20 00 00 E3 63

SelParam=0x01 (Target: 3’ b000, Action: 3’ b000, MemBank: 2’ b01);

Ptr =0x00000020;

MaskLen=0x00;

Truncate=0x00;

Mask=null;

### 4.2.4 Set Select Parameter (0x43)

Set Select command parameters.

If inventory mode=0x02, in the case of multiple tags in the antenna field, before sending inventory tags operations, we first send Select command to select tags, then inventory tags(multiple tags operation), or locking/unlocking/destroying/reading/writing data (one tag selected).

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x43 | 0x00 | | Set Select parameters |
| Parameter | | | |

| SelParam | Ptr | MaskLen | Truncate | Mask (MSB~LSB) |
|---|---|---|---|---|
| 1B | 4B | 1B | 1B | (0~MaskLen/8) |

SelParam: 0x01 (Target: 3’b000, Action: 3’b000, MemBank: 2’b01)

Target: tag inventory session mode (S0, S1, S2, S3, SL)，Default is S0；

Action: tag inventory handling flip flags, Default is “000”.

MemBank: tag memory area

| | | |
|---|---|---|
| Bank 11 | User | b11 = 0x03 |
| Bank 10 | TID | b10 = 0x02 |
| Bank 01 | EPC | b01 = 0x01 |
| Bank 00 | RESERVED | b00 = 0x00 |

Ptr: The start address of the tag access, the length of the bit unit, not word unit.

MaskLen: The length of the mask data, e.g. the length of the mask data 0x60(6 words，96bits)

Truncate: truncation. 0x00 Disable truncation，0x80 Enable truncation

Mask: mask data, 0x30751FEB705C5904E3D50D70 （EPC）

Reference documents《EPC™ Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface Protocol for Communications at 860 MHz－960 MHz Version 2.0.0 Ratified》

Response frame

| **Main** | **Sub** | **Status** | **Data** |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x3A | 0x00 | FAIL_OK | 0x00 |

e.g.:

SelParam=0x01 (Target: 3’b000, Action: 3’b000, MemBank: 2’b01);

Ptr =0x00000020;

MaskLen=0x60;

Truncate=0x00;

Mask =0x E2 00 10 26 77 01 00 97 22 00 31 58；

>Host:AA AA 01 18 43 00 01 00 00 00 20 60 00 E2 00 10 26 77 01 00 97 22 00 31 58 44 65

>Reader: AA AA 01 07 43 00 00 00 C7 DE

#### 4.2.5 Get Query Parameter (0x44)

Get the Query command parameters.

| **Main** | **Sub** | **Parameter** | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0x44 | 0x00 | | Get Query parameter |

Response frame

| **Main** | **Sub** | **Status** |
|---|---|---|
| CMDH | CMDL | Status |

| 0x44 | 0x00 | FAIL_OK | |
|---|---|---|---|
| Data | | | |
| Para(MSB) | Para(LSB) | | |
| 1B | 1B | | |

Set the Query command parameters. The parameter is 2 bytes, which is made by the bit splicing of the following parameters:

DR(1 bit): DR=8(1’ b0), DR=64/3(1’ b1). **Only support DR=8**

M(2 bit): M=1(2’ b00), M=2(2’ b01), M=4(2’ b10), M=8(2’ b11). **Only support M=1.**

TRext(1 bit): No pilot tone(1’ b0), Use pilot tone(1’ b1). **Only support Use pilot tone(1’ b1)!**

Sel(2 bit): ALL(2’ b00/2’ b01), ~SL(2’ b10), SL(2’ b11)

Session(2 bit): S0(2’ b00), S1(2’ b01), S2(2’ b10), S3(2’ b11)

Target(1 bit): A(1’ b0), B(1’ b1)

Q(4 bit): 4’ b0000~4’ b1111. Q value: access the number of $2^Q$ tags. The greater the Q value, the longer cycle of the reader and writer tag inventory.

| **DR** | **M** | **TRext** | **Sel** | **Session** | **Target** | **Q** |
|---|---|---|---|---|---|---|
| 1 | 2 | 1 | 2 | 2 | 1 | 4 |
| 0: DR=8<br>1: DR=64/3 | 00: M=1<br>01: M=2<br>10: M=4<br>11: M=8 | 0: No pilot tone<br>1: Use pilot tone | 00: All<br>01: All<br>10: ~**SL**<br>11: **SL** | 00: S0<br>01: S1<br>10: S2<br>11: S3 | 0: *A*<br>1: *B* | 0–15 |

>Host:AA AA 01 05 44 00 59 BC

>Reader: AA AA 01 08 44 00 00 10 28 73 65

Reference documents《EPC™ Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface Protocol for Communications at 860 MHz – 960 MHz Version 2.0.0 Ratified》

### 4.2.6 Set Query Parameter (0x45)

Set the Query command parameters.

The parameter is 2 bytes, which is made by the bit splicing of the following parameters:

DR(1 bit): DR=8(1’ b0), DR=64/3(1’ b1). **Only support DR=8**

M(2 bit): M=1(2’ b00), M=2(2’ b01), M=4(2’ b10), M=8(2’ b11). **Only support M=1.**

TRext(1 bit): No pilot tone(1’ b0), Use pilot tone(1’ b1). **Only support Use pilot tone(1’ b1)!**

Sel(2 bit): ALL(2’ b00/2’ b01), ~SL(2’ b10), SL(2’ b11), chooses which Tags respond to the Query.

Session(2 bit): S0(2’ b00), S1(2’ b01), S2(2’ b10), S3(2’ b11)

Target(1 bit): A(1’ b0), B(1’ b1)

Q(4 bit): 4’ b0000~4’ b1111. Q value: access the number of $2^Q$ tags. The greater the Q value, the longer cycle of the reader and writer tag inventory.

| **DR** | **M** | **TRext** | **Sel** | **Session** | **Target** | **Q** |
|---|---|---|---|---|---|---|
| 1 | 2 | 1 | 2 | 2 | 1 | 4 |
| 0: DR=8<br>1: DR=64/3 | 00: M=1<br>01: M=2 | 0: No pilot tone<br>1: Use pilot tone | 00: All<br>01: All | 00: S0<br>01: S1 | 0: *A*<br>1: *B* | 0–15 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 10: M=4<br>11: M=8 | | 10: ~**SL**<br>11: **SL** | 10: S2<br>11: S3 | | |

| **Main** | **Sub** | **Parameter** | | |
|---|---|---|---|---|
| CMDH | CMDL | Parameter | | Functional Specifications |
| 0x45 | 0x00 | Para(MSB) 1B | Para(LSB) 1B | Set Query parameter |

Response frame

| **Main** | **Sub** | **Status** | **Data** |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0x45 | 0x00 | FAIL_OK | 0x00 |

>Host: AA AA 01 07 45 00 10 28 46 5E
>Reader: AA AA 01 07 45 00 00 00 E0 47

### 4.2.7 Stop multiple Tags inventory (0xC0)

In the process of initiating multiple inventory tags by the host, multiple Tags inventory operation by multiple antennas can be stopped immediately.

| **Main** | **Sub** | **Parameter** | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0xC0 | 0x00 | | Stop multiple inventory multiple Tags |

Response frame

| **Main** | **Sub** | **Status** | **Data** |
|---|---|---|---|
| CMDH | CMDL | Status | Data |
| 0xC0 | 0x00 | FAIL_OK | TagCounter (4B) |

>Host: AA AA FF 05 C0 00 B3 F7
>Reader: AA AA FF 0A C0 00 00 **00 00 00 C6** 6E C4 // Inventory tag is 0x000000C6=198 times.

### 4.2.8 Start multiple Tags inventory (0xC1)

Start multiple Tags and multiple antennas inventory, inventory Number Range of value 0-65535.

| **Main** | **Sub** | **Parameter** | | |
|---|---|---|---|---|
| CMDH | CMDL | Parameter | | Functional Specifications |
| 0xC1 | 0x00 | QV(1B) | InvNumber (2B) | Algorithm 0 |
| 0xC1 | 0x01 | | | Algorithm 1 |
| 0xC1 | 0x02 | | | Algorithm 2 |

- QV: Q Value. The scope of the amount of inventory to be labelled is defined: $2^Q$-1 tags. The proposed Q value ($2^Q$-1 tags) is more than 20% larger than the actual number of tags. But the greater the Q value, the longer the cycle required to inventory the tags, so the setting of the Q value should be considered.

- InvNumber: Inventory Number (times), Range of value 0-65535.
  (1) InvNumber =0, Limitless inventory tags；
  (2) InvNumber <>0, The reader / writer module inventory to the number of times specified by InvNumber, automatically terminates the inventory tag, and reports the 0xC0 command frame to the host to notify the host to end the inventory

Please refer to the antenna parameter control command (0x3F).

Response frame

| Main | Sub | Status |
|---|---|---|
| CMDH | CMDL | Status |
| 0xC1 | 0x00 | FAIL_OK;<br>FAIL_INVENTORY_TAG_TIMEOUT; |

| Data | | | | |
|---|---|---|---|---|
| RSSI | PC | EPC | StroredCRC | ANT |
| 1B | 2B | Length defined as PC | 2B | 1B |

- **RSSI**: The signal intensity of the tag returned to the reader.

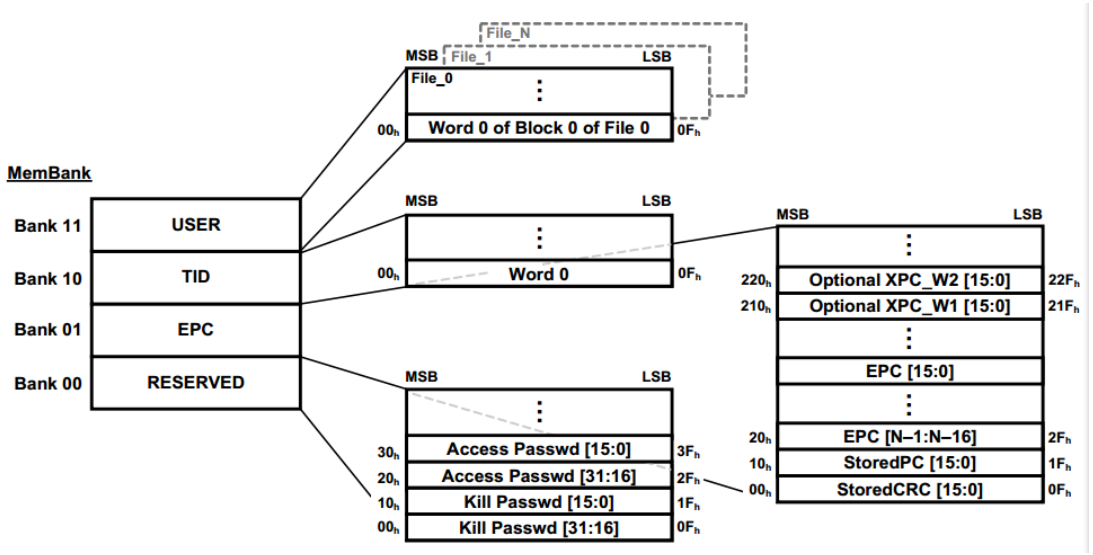
The RSSI value reflects the size of the input signal of the RF chip on the reader module (signal intensity), and does not contain the antenna gain and the attenuation of the directional coupler. The data is a symbolic number of sixteen (HEX), and the unit is dBm. Assuming that RSSI is 0xC9, the value of the RSSI value is -55dBm calculated by the complement value of the 0xC9.

- **PC**: The tag's protocol control character (Protocol Control), 2 bytes. Stored on an EPC MemBank with a starting address of 0x01, with a length of one word (2 bytes).

The PC value defines the length of the word length for the EPC, which is determined by the bits 15h - 11h of the PC value. For example, if PC=0x3000=0011000000000000b, the data of its 15~11 bit is: 00110b =0x06; that is 6 words (12 bytes).

Algorithm formula: Epc Length = (PC > 11) * 2 bytes

- **EPC**: The tag's EPC ID data is stored on the EPC MemBank with the starting address of 0x02, and its length is defined by the TAG protocol control character (PC).
- **StroredCRC**: The CRC16 verification of PC+EPC data, using CRC-CCITT standard to generate polynomials are X16 + X12 + X5 + X0. It is stored on the EPC MemBank with a starting address of 0x00, with a length of one word (2 bytes).
- **ANT**: The sequence number of the operating antenna corresponding to the current tag.
  Antenna number is: 0x00->ANT1, 0x01->ANT2, 0x02->ANT3, 0x03->ANT4 …




>Host: AA AA FF 08 C1 00 05 00 BC 44 4C //QV=5, InvNumber = 0x00BC=188
>Reader: AA AA FF 17 C1 00 00 CF 30 00 00 00 00 00 00 00 00 10 00 00 2D E3 C5 32 E8
>Reader: AA AA FF 17 C1 00 00 CF 30 00 00 00 00 00 00 00 00 10 00 00 2D E3 C5 32 E8
>Reader: AA AA FF 06 C1 00 15 8E C0
>Reader: AA AA FF 18 C1 00 00 BB 30 00 E2 00 41 06 22 18 00 64 19 80 47 1E 21 3D 00 BD 83
>Reader: AA AA FF 18 C1 00 00 BC 30 00 E2 00 30 09 28 11 01 46 11 20 A5 20 23 98 00 4D 56
>Reader: AA AA FF 18 C1 00 00 BC 30 00 E2 00 30 09 28 11 01 46 11 20 A5 20 23 98 00 4D 56
>Reader: AA AA FF 18 C1 00 00 C9 30 00 11 22 33 44 55 66 77 88 99 00 AA BB 01 0B 00 B1 7F
>Reader: AA AA FF 18 C1 00 00 BB 30 00 E2 00 41 06 22 18 00 64 19 80 47 1E 21 3D 00 BD 83
>Reader: AA AA FF 18 C1 00 00 B9 30 00 E2 00 41 06 22 18 00 64 19 80 47 1E 21 3D 00 4D C0
>Reader: AA AA FF 18 C1 00 00 C9 30 00 11 22 33 44 55 66 77 88 99 00 AA BB 01 0B 00 B1 7F
>Reader: AA AA FF 18 C1 00 00 BB 30 00 E2 00 30 09 28 11 01 46 11 20 A5 20 23 98 00 BD 83
.....
>Reader: AA AA FF 0A C0 00 00 00 00 07 A1 EB 12

### 4.2.9 Write Tag data storage area (0xC2)

For a single tag, write the data of the specified address and length into the tag data storage area.
Note:
1. The tag data area address offset (SA) and the length (DL) of the tag data to be written, their unit is word, which is 2 bytes (16 Bits).
2. The Select parameter should be set before this command to select the specified tag to write data of storage area in this tag.
3. If the Access Password is all zero, the Access instruction is not sent.
4. The length of the written data to the tag data store should be no more than 32 word, that is the 64 bytes(512 Bits)。

| Main | Sub | parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0xC2 | 0x00 | | Write tag data storage area |
| Parameter | | | |

| AP | MemBank | SA | DL | DT |
|---|---|---|---|---|
| 4B | 1B | 2B | 2B | Length defined as DL |

- **AP**: Access Password，If there is no access to the password, for all 0, that is, 0x00000000. MSB is before, LSB is behind.
- **MemBank** Tag data storage area，e.g. MemBank=0x03，User area。

| | | |
|---|---|---|
| Bank 11 | User | b11 = 0x03 |
| Bank 10 | TID | b10 = 0x02 |
| Bank 01 | EPC | b01 = 0x01 |
| Bank 00 | RESERVED | b00 = 0x00 |

- **SA**: Tag data area start address. e.g. SA=0x0000
- **DL**: Data length. e.g. DL=0x0002
- **DT**: Data to be written about. E.g. DT=0x12345678

Response frame

| **Main** | **Sub** | **Status** | |
|---|---|---|---|
| CMDH | CMDL | Status | |
| 0xC2 | 0x00 | FAIL_OK;<br>FAIL_WRITE_MEMORY_NO_TAG; | |
| Data | | | |
| UL | PC | EPC | AccessStatus |
| 1B | 2B | Length defined as PC | 1B |

- **UL**: the length of PC+EPC, e.g. UL: 0x0E
- **PC**: Tag's protocol control characters (Protocol Control).
  e.g. **PC**=0x3400
- **EPC**: The EPC ID data of the tag, Length defined as PC.
  e.g. EPC =0xE200102677010097220031 58
- **AccessStatus**: State after the execution of the command.
  e.g. AccessStatus=0x00(Execute the command successfully)

The steps to write the data to the tag:

**1. Read the EPC ID data of the tag:**

>Host: AA AA FF 05 C8 00 3A 5E

>Reader: AA AA FF 17 C8 00 00 D5 30 00 E2 00 10 26 77 01 00 97 22 00 31 58 C2 B0 02 37

**2. Select the specified tag:**

>Host: AA AA FF 18 43 00 01 00 00 00 20 60 00 E2 00 10 26 77 01 00 97 22 00 31 58 10 96

>Reader: AA AA FF 07 43 00 00 00 D6 C1

**3. Read data from the specified tag**

>Host: AA AA FF 0E C9 00 00 00 00 00 03 00 00 00 08 E7 4B

>Reader: AA AA FF 25 C9 00 00 0E 30 00 E2 00 10 26 77 01 00 97 22 00 31 58 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 40 B3

The original data: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

**4. Write data to the specified tag**

>Host: AA AA FF 1E C2 00 00 00 00 00 03 00 00 00 08 AA 11 22 33 44 55 66 77 88 99 00 AA BB CC DD EE 48 1E

>Reader: AA AA FF 16 C2 00 00 0E 30 00 E2 00 10 26 77 01 00 97 22 00 31 58 00 C4 6F

**5. Re-read data from the specified tag**

>Host: AA AA FF 0E C9 00 00 00 00 00 03 00 00 00 08 E7 4B

>Reader: AA AA FF 25 C9 00 00 0E 34 00 E2 00 10 26 77 01 00 97 22 00 31 58 AA 11 22 33 44 55 66 77 88 99 00 AA BB CC DD EE 95 F2

**6. Write data success.**

>Host: AA AA FF 1A C2 00 00 00 00 00 01 00 02 00 06 11 22 33 44 55 66 77 88 99 00 AA BB 40 58

>Reader: AA AA FF 06 C2 00 BB 83 B4 // Insufficient power to write data.

### 4.2.10 Lock Tag (0xC3)

Lock or Unlock data storage area of a single tag.

Note: the Set Select Parameter (0x43) should be run to select tag before Lock operation on the specified tag.

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0xC3 | 0x00 | | Lock Tag |
| Parameter | | | |
| AP | | LD | |
| 4B | | 3B | |

- **AP**: Access Password, if there is no access to the password, for all 0, that is, 0x00000000. MSB is before, LSB is behind.
- **LD**: Lock operation parameter.

The high 4 bit of the Lock operation parameter LD is a reservation bit, and the remaining 20 bits are the Lock operation Payload, including Mask and Action, each 10 bits from high to low.

Reference documents《EPC™ Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface Protocol for Communications at 860 MHz – 960 MHz Version 2.0.0 Ratified》， section 6.3.2.12.3.5 Lock (mandatory) 。

Only Mask bit=1, then Action is valid. The Action of each data area has 2 bits, 00~11, which in turn corresponds to open, permanently open, locked, and permanently locked.

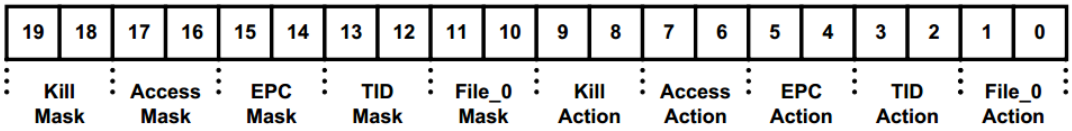
Only Action with a Mask bit of 1 is valid. The Action of each data area has 2 bits, 00~11, which in turn corresponds to open, permanently open, locked, and permanently locked.

| sort | Action | value |
|---|---|---|
| 1 | **Open (Unlock)** | 00 |
| 2 | **Perma-Open** | 01 |
| 3 | **Lock** | 10 |
| 4 | **Perma-Lock** | 11 |

For example, Kill Mask is 2bits 00, and no matter what Kill Action is, Kill Action will not take effect. When the Kill Mask is 10(2 bits) and Kill Action is 10(2 bits), show Kill Password Lock (not Permanent Lock) live, only through the effective Access Password to read and write.

The meaning of each of Mask and Action is shown in the following table.

**Lock-Command Payload**




**Masks and Associated Action Fields**

| | Kill pwd | | Access pwd | | EPC memory | | TID memory | | File_0 memory | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
| *Mask* | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write | skip/ write |
| | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| *Action* | pwd read/ write | perma lock | pwd read/ write | perma lock | pwd write | perma lock | pwd write | perma lock | pwd write | perma lock |

| Memory Bank | Lock Type | Read-Write Permission |
|---|---|---|
| EPC Bank<br>TID Bank (Read only)<br>User Bank | Unlock<br>Open | The memory of the tag can be read and written without access password。 |
| | Lock | The tag access password is all 0, and the label can be read and written without a password.<br>The tag access password is not 0, readable but inwritable, and need to enter the correct access password to read and write. |
| | Perma-unlock<br>Perma-Open | Without access password, you can read and write the tag, and it is permanently rewritten, and no other lock operations can be done.。 |
| | Perma-lock | The tag storage area is readable, but it is permanently unwritten, and no other locking operations can be done.。 |
| Reserver Bank<br>(Kill Password)<br>(Access Password) | Unlock<br>Open | The memory of the tag can be read and written without access password。 |
| | Lock | The tag access password is all 0, and the tag can be read and written without a password.<br>The tag access password is not 0, and the tag cannot be read or written. and needs to enter the correct access password to read and write. |
| | Perma-unlock<br>Perma-Open | Without access password, you can read and write the tag, and it is permanently rewritten, and no other lock operations can be done.。 |

| | | |
|---|---|---|
| | Perma-lock | Perpetual unreadable, unwritten, and no other locking operations |

Response frame

| Main | Sub | Status |
|---|---|---|
| CMDH | CMDL | Status |
| 0xC4 | 0x00 | FAIL_OK;<br>FAIL_ACCESS_PWD_ERROR;<br>FAIL_LOCK_NO_TAG;<br>FAIL_LOCK_ERROR_CODE_BASE; |

| Data | | | |
|---|---|---|---|
| UL | PC | EPC | AccessStatus |
| 1B | 2B | Length defined as PC | 1B |

- **UL**: the length of PC+EPC, e.g. UL: 0x0E
- **PC**: Tag's protocol control characters (Protocol Control).
  e.g. **PC**=0x3400
- **EPC**: The EPC ID data of the tag, Length defined as PC.
  e.g. EPC =0xE2001026770100972200 3158
- **AccessStatus**: State after the execution of the command.
  e.g. AccessStatus=0x00(Execute the command successfully)

*If there is no tag in the RF area or the specified EPC code is wrong, the error code 0x13 will be returned.
* If Access Password is not correct, it returns the error code 0x16 and the PC+EPC of the operating tag.
* If the operation tag returns the error code specified by EPC Gen2 (error codes), the response frame will return the new error code value (error code and 0xC0 are performed or operated). For example, if the tag TID area has been permanently locked, then set the TID area to the open state through the Lock directive. According to the EPC Gen2 protocol, the tag will return to error code 0x04 (Memory Locked). The response frame returns the error code 0xC4 and returns the PC+EPC of the operating tag.

//Suppose Access Password 0x01010101
>Host: AA AA FF 0C C3 00 01 01 01 01 02 00 80 A5 8D //lock Access Password
>Reader: AA AA FF 16 C3 00 00 0E 34 00 11 22 33 44 55 66 77 88 99 00 AA BB 00 D3 ED

>Host: AA AA FF 0C C3 00 01 01 01 01 00 00 00 5A 65 //Unlock Access Password
>Reader: AA AA FF 16 C3 00 00 0E 34 00 11 22 33 44 55 66 77 88 99 00 AA BB 00 D3 ED

>Host: AA AA FF 0C C3 00 00 00 00 00 00 08 02 D2 AA
>Reader: AA AA FF 06 C3 00 13 80 66 //Access password is error.

>Host: AA AA FF 0C C3 00 01 01 01 01 00 08 02 F3 8E //Lock User area.
>Reader: AA AA FF 16 C3 00 00 0E 34 00 11 22 33 44 55 66 77 88 99 00 AA BB 00 D3 ED
>Host: AA AA FF 0E C9 00 01 01 01 01 03 00 02 00 08 D8 1A //Read User area.
>Reader: AA AA FF 25 C9 00 00 0E 34 00 11 22 33 44 55 66 77 88 99 00 AA BB EE FF 11 22 33 44 55 66 77 88 99

00 00 00 00 00 B6 81
>Host: AA AA FF 1E C2 00 00 00 00 00 03 00 02 00 08 BB BB 11 22 33 44 55 66 77 88 99 AA AA AA AA AA AC BA // Access password is error 00 00 00 00 < > 01010101
>Reader: AA AA FF 06 C2 00 B4 72 5B //Write fail.
>Host: AA AA FF 1E C2 00 01 01 01 01 03 00 02 00 08 BB BB 11 22 33 44 55 66 77 88 99 AA AA AA AA AA D6 DE // Access password is ok 01010101
>Reader: AA AA FF 16 C2 00 00 0E 34 00 11 22 33 44 55 66 77 88 99 00 AA BB 00 0D F2 //Write ok.
When the User area is locked, you need the correct access password to modify (in addition to permanent locking) User area data。
>Host: AA AA FF 0C C3 00 01 01 01 01 00 08 02 F3 8E
>Reader: AA AA FF 06 C3 00 13 80 66

### 4.2.11 Kill Tag (0xC4)

Kill operation for single tag. The Select parameter should be set before this command to select the specified tag for the Kill operation.

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0xC4 | 0x00 | KP (4B) | Kill single tag |

- **KP**: Kill Password, Stored in MemBank=Reserved, the address is 0 and 2 word lengths.

Response frame

| Main | Sub | Status | |
|---|---|---|---|
| CMDH | CMDL | Status | |
| 0xC4 | 0x00 | FAIL_OK;<br>FAIL_ACCESS_PWD_ERROR;<br>FAIL_LOCK_NO_TAG;<br>FAIL_LOCK_ERROR_CODE_BASE; | |
| Data | | | |
| UL | PC | EPC | AccessStatus |
| 1B | 2B | Length defined as PC | 1B |

- **UL**: the length of PC+EPC, e.g. UL: 0x0E
- **PC**: Tag's protocol control characters (Protocol Control).
  e.g. **PC**=0x3400
- **EPC**: The EPC ID data of the tag, Length defined as PC.
  e.g. EPC =0xE2001026770100972200 3158
- **AccessStatus**: State after the execution of the command.
  e.g. AccessStatus=0x00(Execute the command successfully)

// Sending incorrect kill passwords
>Host: AA AA FF 09 C4 00 00 00 00 00 58 05 // KP=0x01020304，
>Reader: AA AA FF 06 C4 00 12 15 D7

// Sending correct kill passwords

>Host: AA AA FF 09 C4 00 01 02 03 04 55 06 // KP=0x01020304

>Reader: AA AA FF 16 C4 00 00 0E 34 00 11 22 33 44 55 66 77 88 99 00 AA BB 00 E9 F2

// **Kill tag success, the tag is no longer read & write access, such as no longer to be read or write data, a tag has been destroyed.**

### 4.2.12 Single Tag Inventory (0xC8)

Completing once inventory operation in accordance with the EPC Class1 Gen2 protocol.

The Select operation is not included in this command. RF Emission Power Capacity automatically opens and closes before and after the execution of the single tag inventory command.

When a single inventory tags, the number of inventory tags is configured by another Query command, and the firmware has an initial Q value.

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0xC8 | 0x00 | | Single tag inventory |

Response frame

| Main | Sub | Status |
|---|---|---|
| CMDH | CMDL | Status |
| 0xC4 | 0x00 | FAIL_OK;<br>FAIL_ACCESS_PWD_ERROR;<br>FAIL_LOCK_NO_TAG;<br>FAIL_LOCK_ERROR_CODE_BASE; |

| Data | | | | |
|---|---|---|---|---|
| RSSI | PC | EPC | StroredCRC | ANT |
| 1B | 2B | Length defined as PC | 2B | 1B |

- **RSSI**: The signal intensity of the tag returned to the reader.

The RSSI value reflects the size of the input signal of the RF chip on the reader module (signal intensity), and does not contain the antenna gain and the attenuation of the directional coupler. The data is a symbolic number of sixteen (HEX), and the unit is dBm. Assuming that RSSI is 0xC9, the value of the RSSI value is -55dBm calculated by the complement value of the 0xC9.

- **PC**: The tag's protocol control character (Protocol Control), 2 bytes. stored on a EPC MemBank with a starting address of 0x01, with a length of one word (2 bytes).

The PC value defines the length of the word length for the EPC, which is determined by the bits 15h - 11h of the PC value. For example, if PC=0x3000=0011000000000000b, the data of its 15~11 bit is: 00110b =0x06; that is 6 words (12 bytes).

- **EPC**: The tag's EPC ID data is stored on the EPC MemBank with the starting address of 0x02, and its length is defined by the TAG protocol control character (PC).
- **StroredCRC**: The CRC16 verification of PC+EPC data, using CRC-CCITT standard to generate polynomials are X16 + X12 + X5 + X0. It is stored on the EPC MemBank with a starting address of 0x00, with a length of one word (2 bytes).
- **ANT**: The sequence number of the operating antenna corresponding to the current tag.

Antenna number is: 0x00->ANT1, 0x01->ANT2, 0x02->ANT3, 0x03->ANT4 …

>Host: AA AA FF 05 C8 00 3A 5E // Single tag inventory, return three tags of EPC data once。
>Reader: AA AA FF 18 C8 00 00 BD 30 00 E2 00 30 09 28 11 01 46 11 20 A5 20 23 98 00 A7 E7
>Reader: AA AA FF 18 C8 00 00 C6 30 00 11 22 33 44 55 66 77 88 99 00 AA BB 01 0B 00 AA 45
>Reader: AA AA FF 18 C8 00 00 B8 30 00 E2 00 41 06 22 18 00 64 19 80 47 1E 21 3D 00 A7 71

### 4.2.13 Read Tag data storage area (0xC9)

For a single tag, read the data of the specified address and length from the tag data storage area.
Note:
1. The tag data area address offset (SA) and the length (DL) of the tag data to be read, their unit is word, which is 2 bytes (16 Bits).
2. The Select parameter should be set before this command. To select the specified tag to read data of storage area in this tag.
3. If the Access Password is all zero, the Access instruction is not sent.

| **Main** | **Sub** | **Parameter** | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0xC9 | 0x00 | | Read Tag data storage area |

| **Parameter** | | | | |
|---|---|---|---|---|
| AP | MemBank | SA | DL | |
| 4B | 1B | 2B | 2B | |

- **AP**: Access Password，If there is no access to the password, for all 0, that is, 0x00000000. MSB is before, LSB is behind.
- **MemBank**: data storage area in the Tag，e.g. MemBank=0x03，User area。

| | | |
|---|---|---|
| Bank 11 | User | b11 = 0x03 |
| Bank 10 | TID | b10 = 0x02 |
| Bank 01 | EPC | b01 = 0x01 |
| Bank 00 | RESERVED | b00 = 0x00 |

- **SA**: Tag data area start address. e.g. SA=0x0000
- **DL**: Data length. e.g. DL=0x0002

Response frame

| **Main** | **Sub** | **Status** |
|---|---|---|
| CMDH | CMDL | Status |
| 0xC9 | 0x00 | FAIL_OK；<br>FAIL_READ_MEMORY_NO_TAG；<br>FAIL_READ_ERROR_CODE_BASE； |

| **Data** | | | |
|---|---|---|---|
| UL | PC | EPC | DT |
| 1B | 2B | Length defined as PC | 1B |

- **UL**: the length of PC+EPC, e.g. UL: 0x0E

- **PC**: Tag's protocol control characters (Protocol Control).
  e.g. **PC**=0x3400
- **EPC**: The EPC ID data of the tag, Length defined as PC.
  e.g. EPC =0xE200102677010097220031 58
- **AccessStatus**: State after the execution of the command.
  e.g. AccessStatus=0x00(Execute the command successfully)

>Host: AA AA FF 18 43 00 01 00 00 00 20 60 00 E2 00 60 00 62 0D 00 21 11 50 9E D3 32 E1 //Select
>Reader: AA AA FF 07 43 00 00 00 D6 C1
>Host: AA AA FF 0E C9 00 00 00 00 00 03 00 00 00 08 E7 4B //Read
>Reader: AA AA FF 26 C9 00 00 0E 34 00 E2 00 60 00 62 0D 00 21 11 50 9E D3 11 22 33 44 55 66 77 88 99 00 AA BB CC DD EE FF 00 D3 93

#### 4.2.14 Read the sensor tag (0xCA)

Read the sensor value (temperature and water) of the specified sensor tag (EPC ID)

| Main | Sub | Parameter | |
|---|---|---|---|
| CMDH | CMDL | Parameter | Functional Specifications |
| 0xCA | 0x10 | | Read the sensor value |

| Parameter | | | | |
|---|---|---|---|---|
| Sensor Tag Type | Read Tag Number | Access Password | EPC Len | EPC ID |
| 1B | 1B | 4B | 1B | EPC Len |

- **Sensor Tag Type**: Sensor tag type, 0x01 RFM temperature tag, 0x02 RFM water tag.
- **Read Tag Number**: in this command, the number of times the tag will be read.
- **Access Password**: Access password, if the tag does not set the password, all is 0x00.
- **EPC Len**: The length of EPC ID data in the specified tag.
- **EPC ID**: The EPC ID data in the specified tag.

Response frame

| Main | Sub | Status |
|---|---|---|
| CMDH | CMDL | Status |
| 0xCA | 0x10 | FAIL_OK<br>FAIL_INVENTORY_TAG_TIMEOUT |

| Data | | | | | | |
|---|---|---|---|---|---|---|
| UL | PC | EPC ID | ANT | Sensor MSB | Sensor LSB | Sensor RSSI |
| 1B | 2B | UL-2 | 1B | 1B | 1B | 1B |

- **UL**: the length of PC+EPC
- **PC**: Tag's protocol control characters (Protocol Control), the length is 2 bytes.
- **EPC**: The EPC ID data of the tag, Length defined as PC.
- **ANT**: The sequence number of the operating antenna corresponding to the current tag.
- Sensor MSB: The value of the sensor, high byte
- Sensor LSB: The value of the sensor, low byte

- Sensor RSSI: The signal intensity of the tag returned to the reader.

>Host:AA AA 01 18 CA 10 01 02 00 00 00 00 0C 00 00 00 00 00 00 00 00 10 00 00 2D F7 FF
>Reader: AA AA 01 19 CA 10 00 0E 30 00 00 00 00 00 00 00 00 00 10 00 00 2D FF 00 FB 1F FE 53
>Reader: AA AA 01 19 CA 10 00 0E 30 00 00 00 00 00 00 00 00 00 10 00 00 2D FF 00 FB 1F FE 53

Temperature value: 0x00FB = 251， show 251.0/10.0=25.1℃；
Temperature value: 0x0352 = 850， show 850.0/10.0=85.0℃；

>Host: AA AA FF 18 CA 10 01 1E 00 00 00 00 0C 00 00 00 00 00 00 00 00 10 00 00 2D 50 3B
>Reader;
AA AA FF 19 CA 10 00 0E 30 00 00 00 00 00 00 00 00 00 10 00 00 2D 00 FF D1 14 CB A4
AA AA FF 19 CA 10 00 0E 30 00 00 00 00 00 00 00 00 00 10 00 00 2D 00 FF CB 0D A4 04
AA AA FF 19 CA 10 00 0E 30 00 00 00 00 00 00 00 00 00 10 00 00 2D 00 FF CE 12 B8 2F
AA AA FF 19 CA 10 00 0E 30 00 00 00 00 00 00 00 00 00 10 00 00 2D 00 FF CF 12 8B 1E
AA AA FF 19 CA 10 00 0E 30 00 00 00 00 00 00 00 00 00 10 00 00 2D 00 FF CA 0B F7 F3
AA AA FF 19 CA 10 00 0E 30 00 00 00 00 00 00 00 00 00 10 00 00 2D 00 FF CE 0C 4B D0
AA AA FF 19 CA 10 00 0E 30 00 00 00 00 00 00 00 00 00 10 00 00 2D 00 FF CC 0B 5D 55

**The formula for calculating the temperature value is a complement method**:
T = （~V+1）/10.0
Temperature value: 0xFFD1 --> -4.7℃
Temperature value: 0xFFCC --> -5.2℃
Temperature value: 0xFFF4 = -12， show -12℃；

Note: before reading the temperature tag, if you don't know the EPC ID of the tag, first use the command (0xC8) to read all the EPC ID of the all sensor tags, then use command (0xCA) to read the sensor's value of the specified tag (EPC ID).

Tel: 0086-755-82706913
Fax: 0086-755-82706900
Mail: sales@sysiot.cn
Http://www.sysiot.cn

Add: Room 716, Yiben E-Commerce and Industrial Park, Chaguang Road Nanshan District, Shenzhen, China
Post Code：518055

# Appendix A : CRC check

CRC polynomial $X^{16} + X^{12} + X^5 + X^0$

**Look-up table method:**

```
unsigned int code CRC_Table[256]={
0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7,
0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef,
0x1231, 0x0210, 0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de,
0x2462, 0x3443, 0x0420, 0x1401, 0x64e6, 0x74c7, 0x44a4, 0x5485,
0xa56a, 0xb54b, 0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4,
0xb75b, 0xa77a, 0x9719, 0x8738, 0xf7df, 0xe7fe, 0xd79d, 0xc7bc,
0x48c4, 0x58e5, 0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b,
0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12,
0xdbfd, 0xcbdc, 0xfbbf, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41,
0xedae, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49,
0x7e97, 0x6eb6, 0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78,
0x9188, 0x81a9, 0xb1ca, 0xa1eb, 0xd10c, 0xc12d, 0xf14e, 0xe16f,
0x1080, 0x00a1, 0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e,
0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256,
0xb5ea, 0xa5cb, 0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c,
0x26d3, 0x36f2, 0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab,
0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3,
0xcb7d, 0xdb5c, 0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92,
0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaa, 0xad8b, 0x9de8, 0x8dc9,
0x7c26, 0x6c07, 0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8,
0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0
};
unsigned int CRC16Check(unsigned char *ptr, unsigned char DataLen)
{
    unsigned int CRC;
    unsigned char data DataReg;
    CRC=0xffff;
    while(DataLen--!=0)
     {
```

```
        DataReg=(unsigned char) (CRC/256);
        CRC<<=8;
        CRC^=CRC_Table[DataReg^*ptr];
        ptr++;
    }
    return CRC;
}
```

**Shifting method:**

```
unsigned int    WINAPI    Calculate_CRC(unsigned char *ptr, unsigned char len)
{
    unsigned int xorval;
    unsigned char i,j;
    unsigned int CRCacc = 0xffff;

    for(j = 0; j < len; j++)
    {
        for (i=0; i<8; i++)
        {
            xorval = ((CRCacc>>8) ^ (ptr[j] << i)) & 0x0080;
            CRCacc = (CRCacc << 1) & 0xfffe;
            if (xorval)
            CRCacc ^= 0x1021;
        }
    }
    return CRCacc;
}
```