**GAORFID UHF**

**RFID application-layer communications protocol**

**SKU# 216002**

(Version V2.0.8)

# 

Table of Contents

[1. Outline 5](#_Toc533752955)

[2. Data transmission frame format 5](#_Toc533752956)

[2.1. Header and trailer 5](#_Toc533752957)

[2.2. Frame length 5](#_Toc533752958)

[2.3. CMD type list 5](#_Toc533752959)

[2.4. data 9](#_Toc533752960)

[2.5. BCC code 9](#_Toc533752961)

[3. A communications data frame described 9](#_Toc533752962)

[3.1. Device version 9](#_Toc533752963)

[3.1.1. Acquisition hardware version number 9](#_Toc533752964)

[3.1.2. Answer acquisition hardware version number 10](#_Toc533752965)

[3.1.3. Obtain the firmware version number 10](#_Toc533752966)

[3.1.4. Firmware version number of responses 11](#_Toc533752967)

[3.1.5. Acquisition acquisition equipmentID 11](#_Toc533752968)

[3.1.6. Get DeviceIDReply 12](#_Toc533752969)

[3.2. Device parameter settings 12](#_Toc533752970)

[3.2.1. Sets the transmit power 12](#_Toc533752971)

[3.2.2. Sets the transmit power response 13](#_Toc533752972)

[3.2.3. Get the current transmit power 13](#_Toc533752973)

[3.2.4. Get the current transmit power response 14](#_Toc533752974)

[3.2.5. Fixed frequency settings 14](#_Toc533752975)

[3.2.6. Fixed-frequency response settings 15](#_Toc533752976)

[3.2.7. Get current device setting state fixed frequency 15](#_Toc533752977)

[3.2.8. Get the current fixed-frequency setting device status response 16](#_Toc533752978)

[3.2.9. Set upGen2parameter 16](#_Toc533752979)

[3.2.10. Set upGen2Parameter Reply 19](#_Toc533752980)

[3.2.11. Get the currentGen2parameter settings 20](#_Toc533752981)

[3.2.12. Get the currentGen2Parameter setting response 20](#_Toc533752982)

[3.2.13. CWSet up twenty one](#_Toc533752983)

[3.2.14. CWSetting the answer twenty two](#_Toc533752984)

[3.2.15. Get the current equipmentCWSet up twenty two](#_Toc533752985)

[3.2.16. Get the current equipmentCWSetting the answer twenty three](#_Toc533752986)

[3.2.17. Antenna Set twenty three](#_Toc533752987)

[3.2.18. Setting the answer antenna twenty four](#_Toc533752988)

[3.2.19. Get the current antenna device settings twenty four](#_Toc533752989)

[3.2.20. Get the current antenna set the answering device 25](#_Toc533752990)

[3.2.21. Band locale 25](#_Toc533752991)

[3.2.22. Band locale response 26](#_Toc533752992)

[3.2.23. Gets band area 26](#_Toc533752993)

[3.2.24. Gets the locale band response 27](#_Toc533752994)

[3.2.25. Provided inventory data area 27](#_Toc533752995)

[3.2.26. Set inventory data area response 28](#_Toc533752996)

[3.2.27. Get Device current temperature 28](#_Toc533752997)

[3.2.28. Get the current temperature-responsive device 29](#_Toc533752998)

[3.2.29. Setting the temperature protection value 29](#_Toc533752999)

[3.2.30. Setting the temperature protection value of the response 30](#_Toc533753000)

[3.2.31. Gets temperature protection value 30](#_Toc533753001)

[3.2.32. Gets temperature protection value response 30](#_Toc533753002)

[3.2.33. An antenna working hours 31](#_Toc533753003)

[3.2.34. An antenna working time response 31](#_Toc533753004)

[3.2.35. Gets antenna working hours 32](#_Toc533753005)

[3.2.36. Obtain an antenna response time 32](#_Toc533753006)

[3.2.37. Recommended settingsRFLink combination 33](#_Toc533753007)

[3.2.38. Recommended settingsRFLink combined response 33](#_Toc533753008)

[3.2.39. Get recommendedRFCombination of links provided 34](#_Toc533753009)

[3.2.40. Get recommendedRFLink combination setting response 34](#_Toc533753010)

[3.2.41. Set upFastIDFeatures 35](#_Toc533753011)

[3.2.42. Set upFastIDAnswer the 35](#_Toc533753012)

[3.2.43. ObtainFastIDFunctional status 35](#_Toc533753013)

[3.2.44. ObtainFastIDFunctional status response 36](#_Toc533753014)

[3.2.45. Set upTagfocus Features 36](#_Toc533753015)

[3.2.46. Set upTagFocus Answer the 37](#_Toc533753016)

[3.2.47. ObtainTagFocus Functional status 37](#_Toc533753017)

[3.2.48. ObtainTagFocus Functional status response 37](#_Toc533753018)

[3.2.49. Software reset 38](#_Toc533753019)

[3.2.50. Software reset response 38](#_Toc533753020)

[3.2.51. Look for the label filter settings 38](#_Toc533753021)

[3.2.52. Find answer label filter settings 40](#_Toc533753022)

[3.2.53. EPC + TIDorEPC + USER Mode setting 40](#_Toc533753023)

[3.2.54. EPC + TIDorEPC + TID + USER Setting the answer mode 40](#_Toc533753024)

[3.2.55. ReadEPC + TIDorEPC + TID + USER Mode status 41](#_Toc533753025)

[3.2.56. ReadEPC + TIDorEPC + TID + USER Answer mode status 41](#_Toc533753026)

[3.2.57. reset 42](#_Toc533753027)

[3.2.58. Factory Reset response 42](#_Toc533753028)

[3.3. Label operation 43](#_Toc533753029)

[3.3.1. Single inventory tags 43](#_Toc533753030)

[3.3.2. Single label inventory response 44](#_Toc533753031)

[3.3.3. Continuous inventory labels 44](#_Toc533753032)

[3.3.4. Continuous label inventory response 45](#_Toc533753033)

[3.3.5. Stop continuous inventory labels 46](#_Toc533753034)

[3.3.6. Stop the continuous label inventory response 46](#_Toc533753035)

[3.3.7. Read the label data area 46](#_Toc533753036)

[3.3.8. Read label data area response 47](#_Toc533753037)

[3.3.9. Write tag data area 48](#_Toc533753038)

[3.3.10. Write data area response 49](#_Toc533753039)

[3.3.11. Locklabel 50](#_Toc533753040)

[3.3.12. LockTag replies 51](#_Toc533753041)

[3.3.13. Killlabel 51](#_Toc533753042)

[3.3.14. KillTag replies 52](#_Toc533753043)

[3.3.15. Block Writedata 52](#_Toc533753044)

[3.3.16. Block WriteResponse data 54](#_Toc533753045)

[3.3.17. Block Erasedata 54](#_Toc533753046)

[3.3.18. Block EraseResponse data 55](#_Toc533753047)

[3.3.19. Set upQTparameter 56](#_Toc533753048)

[3.3.20. Set upQTParameter Reply 57](#_Toc533753049)

[3.3.21. ObtainQTparameter 57](#_Toc533753050)

[3.3.22. ObtainQTParameter Reply 58](#_Toc533753051)

[3.3.23. QTlabelRead 58](#_Toc533753052)

[3.3.24. QTlabelRead answer 59](#_Toc533753053)

[3.3.25. QTlabelWrite 60](#_Toc533753054)

[3.3.26. QTlabelWrite response 61](#_Toc533753055)

[3.3.27. Block Permalockoperating 62](#_Toc533753056)

[3.3.28. Block PermalockOperating answer 63](#_Toc533753057)

# Outline

UHF RFID is an application layer protocol UHF reader module and the external communications protocol. By this protocol, the data communication between the external device and the reader module.

1. UHFModule and PC using asynchronous serial interface (UART, TTL level)Data communication, the baud rate 115200 default data bits, 1 stop bit, no parity bit, no hardware flow control. Data are transmitted according to a fixed frame format.
2. UHF reader may be used in addition to the serial communication, also support TCP / IP communications.

# Data transmission frame format

Data frame header, frame length, the CMD type, data, and frame check code tail components. As shown in Table 1.

Transmission data frame format

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Header | Frame length | Type CMD | data | BCC code | End of frame |
| 2 bytes | 2 bytes | 1 byte | N bytes | 1 byte | 2 bytes |

## Header and trailer

On behalf of the packet header beginning, a total of 2 bytes, a fixed value 0xC8,0x8COr 0xA5,0x5A, end of frame showing the end of the packet, total2 bytes, a fixed value 0x0d, 0x0a (return followed by line feed).

## Frame length

A length of the data frame length, is the length, the end of the entire frame comprising frame header data. The specific terms

Calculation formula is:

Length = frame header (2 bytes) + frame length (2 bytes) + the CMD Type (1 byte) + data (N bytes) + the BCC code (1 byte) + tail frame (2 bytes).

## CMD type list

CMD is the command type to distinguish between different types of control commands, UHF or reader module performs the corresponding operation according to the command type.

CMD type list

|  |  |
| --- | --- |
| Command Function | Type CMD |
| Acquisition hardware version number | 0x00 |
| Answer acquisition hardware version number | 0x01 |
| Obtain the firmware version number | 0x02 |
| Firmware version number of responses | 0x03 |
| Acquisition module ID | 0x04 |
| ID response acquisition module | 0x05 |
| Reserve | 0x06 ~ 0x0f |
| Sets the transmit power | 0x10 |
| Sets the transmit power response | 0x11 |
| Get the current transmit power | 0x12 |
| Get the current transmit power response | 0x13 |
| Hopping Set | 0x14 |
| Setting the answer hopping | 0x15 |
| Gets the current device status hopping set | 0x16 |
| Gets the current device settings hopping state response | 0x17 |
| Setting parameters Gen2 | 0x20 |
| Gen2 parameter setting response | 0x21 |
| Gets the current parameter settings Gen2 | 0x22 |
| Gets the current Gen2 parameter setting response | 0x23 |
| CW Setting | 0x24 |
| CW Setting the answer | 0x25 |
| Gets the current device settings CW | 0x26 |
| Gets the current device settings CW answer | 0x27 |
| Antenna Set | 0x28 |
| Setting the answer antenna | 0x29 |
| Get the current antenna device settings | 0x2a |
| Get the current antenna set the answering device | 0x2b |
| regional settings | 0x2c |
| Locale response | 0x2d |
| Gets the locale | 0x2e |
| Gets the locale response | 0x2f |
| Set upInventory data area | 0x30 |
| Set upInventory data areaReply | 0x31 |
| Reserve | 0x32-0x33 |
| Get Device current temperature | 0x34 |
| Get the current temperature-responsive device | 0x35 |
| Setting the temperature protection value | 0x38 |
| Setting the temperature protection value of the response | 0x39 |
| Get temperature protection value | 0x3A |
| Gets temperature protection setting values ​​in the answer | 0x3B |

|  |  |
| --- | --- |
| Reserve | 0x3C-0x49 |
| An antenna working hours | 0x4A |
| An antenna working time response | 0x4B |
| Gets antenna working hours | 0x4C |
| Obtain an antenna response time | 0x4D |
| Multi-antenna work interval | 0x4E |
| Multi-antenna work interval reply | 0x4F |
| Obtain multi-antenna work interval | 0x50 |
| Obtain multi-antenna work interval reply | 0x51 |
| Recommended combinations of RF link provided | 0x52 |
| Set the recommended combination of RF link reply | 0x53 |
| Get recommended setting combination of RF link | 0x54 |
| Get recommended a combination of RF link setting response | 0x55 |
| Reserve | 0x56-0x5B |
| Setting FastID function | 0x5C |
| Answer the setting FastID | 0x5D |
| Get FastID functional status | 0x5E |
| Get FastID functional status response | 0x5F |
| Setting TagFocus function | 0x60 |
| Answer the setting TagFocus | 0x61 |
| Get TagFocus functional status | 0x62 |
| Get TagFocus functional status response | 0x63 |
| Reserve | 0x64-0x67 |
| Software reset | 0x68 |
| Software reset response | 0x69 |
| Reserve | 0x6A |
| Reserve | 0x6B |
| Reserve | 0x6C |
| Reserve | 0x6D |
| Look for the label filter settings | 0x6E |
| Find answer label filter settings | 0x6F |
| Provided simultaneously read EPC + TIDOr EPC + TIDmode | 0x70 |
| Set read simultaneouslyEPC + TID or EPC + TID mode response | 0x71 |
| Get while reading the EPC + TIDOr EPC + TIDmode | 0x72 |
| Get read simultaneouslyEPC + TID or EPC + TID mode response | 0x73 |

|  |  |
| --- | --- |
| reset | 0x74 |
| Factory Reset response | 0x75 |
| Reserve | 0x76 ~ 0x7f |
| To find a single label | 0x80 |
| To find a single label reply | 0x81 |
| Continuous find labels | 0x82 |
| Continuous labels to find answer | 0x83 |
| Stop seeking continuous label | 0x8c |
| Stop the continuous label to find answer | 0x8d |
| Read Data | 0x84 |
| Read data response | 0x85 |
| Write data | 0x86 |
| Write data response | 0x87 |
| Lock tag | 0x88 |
| Lock tag replies | 0x89 |
| Kill label | 0x8a |
| Kill label reply | 0x8b |
| Reserve | 0x8e-0x92 |
| Block Write tag | 0x93 |
| Block Write tag replies | 0x94 |
| Block Erase Label | 0x95 |
| Block Erase transponder tag | 0x96 |
| Set command parameters QT | 0x97 |
| Set QT command parameter response | 0x98 |
| Get command parameter QT | 0x99 |
| Get command parameter response QT | 0x9a |
| QT read | 0x9b |
| QT read answer | 0x9c |
| QT writes | 0x9d |
| QT write response | 0x9e |
| Block Permalock operation | 0x9f |
| Block Permalock operating answer | 0xa0 |
| Reserve | 0xa1 ~ 0xff |

## data

The CMD type, the data comprising data and control information. For the command frame, control information indicating, for the response frame, data representing the information returned.

## BCC code

All bytes of data per frame (frame head and tail removed) XOR.

##### E.g:

0xC8 0x8C 0x00 0x0A 0x43 0x01 0x25 BCC 0x0d 0x0a

BCC = 0x00 ^ 0x0A ^ 0x43 ^ 0x01 ^ 0x25 = 0x6D

# A communications data frame described

## Device version

### Acquisition hardware version number

Data: None

Function: Get hardware version information

Acquisition hardware version command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | BCC code | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x00 | no | 0x08 | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: This command No data

##### Example: Get a card reader hardware version

command:C8 8C 00 08 00 08 0D 0A

### Answer acquisition hardware version number

data:A total of 3 bytesIncluding major version, minor version and supplements

Function: Acknowledge hardware version information

Provides hardware version of the response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | |
| 0xC8 | 0x8C | 0x00 | 0x0B | 0x01 | Major Version | Minor version | Supplementary version  this |
| BCCcode | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

Description: None

Example: the hardware version of the card reader responses V2.0.1

Command: C8 8C 00 0B 01 02 00 01 09 0D 0A

### Obtain the firmware version number

Data: None

Function: get firmware version information

Get firmware version command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | BCC code | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x02 | no | 0x0A | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: Get the reader firmware version

command:C8 8C 00 08 02 0A 0D 0A

### Firmware version number of responses

data:A total of 3 bytesIncluding major version, minor version and supplements

Function: Answer firmware version

Firmware version response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | |
| 0xC8 | 0x8C | 0x00 | 0x0B | 0x03 | Major Version | Minor version | Supplementary version  this |
| BCC code | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

Description: None

Example: The firmware version number is V3.01The reader response

command:C8 8C 00 0B 03 03 00 01 0A 0D 0A

### Acquisition acquisition equipmentID

Data: None

Function: Get the module ID

Get ID command frame module

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x04 | no | 0x0C | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

Example: obtaining module ID

command:C8 8C 00 08 04 0C 0D 0A

### Get DeviceAnswer ID

data:Altogether 4 bytesThe moduleID.

Function: Get the module ID response.

ID acquisition response frame module

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | |
| 0xC8 | 0x8C | 0x00 | 0x0C | 0x05 | Dbyte3 | DByte2 | DByte1 |
| data | BCC code | End of frame | |  | | | |
| DByte0 | 0xxx | 0x0D | 0x0A |

example:ID for 0xF1 0xF20xF30xF4Reader response

command:C8 8C 00 0C 05 F1 F2 F3 F4 0D 0D 0A

## Device parameter settings

### Sets the transmit power

data:6 bytes, Status one byte, one byte antenna number, write power and read power occupies 2 bytes of read power in dBm units are

Function: specific antenna, which is arranged to read and write power.

Set the transmit power command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | |
| 0xA5 | 0x5A | 0x00 | 0x0E | 0x10 | Status | Antenna No. | Read (MSB) |
| data | | | BCC code | End of frame | |  | |
| Read (LSB) | Write (MSB) | Write (LSB) | 0xxx | 0x0D | 0x0A |

Status of each bit instructions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| Rev | Rev | Rev | Rev | Rev | Rev | 0: Do not save  1: Save | Rev |

Description: 1, bit1 0 indicates that the current settings will be lost after a power outage, bit1 1 indicates that the current settings will be saved after the power failure, the default setting on the next power value of the power value. No. hexadecimal representation of the antenna; read after power × 100, and then converted to hexadecimal.

2, the power is currently reserved for reading, there is no real meaning.

Example: an antenna 1The power of reading 0dBmWrite power 30dBm,do not save.

command:C8 8C 00 0E 10 00 01 00 00 0B B8 AC 0D 0A

### Sets the transmit power response

Data: Set whether symbol of success, success: 0x01; failure: 0x00

Function: Set the transmission power is successful.

Set the transmit power response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x11 | OK-0x01  Fail-0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

Example: setting a transmission power success

Command: C8 8C 00 09 11 01 19 0D 0A

### Get the current transmit power

Data: None

Function: Get the current transmit power.

Gets the current transmit power command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x12 | no | 1A | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: Gets the current transmit power

command:C8 8C 00 08 12 1A 0D 0A

### Get the current transmit power response

Data: Status, the antenna and antenna number read power, write power in dBm units are

Function: Get power of each antenna of the reader device.

Get the current transmit power response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x13 | Status | Antenna No. | Read (MSB) |
| data | | | | | | | |
| Read (LSB) | Write (MSB) | Write (LSB) | Antenna No. | Read (MSB) | Read (LSB) | Write (MSB) | Write (LSB) |
| data | | | | | | BCC code | End of frame |
| ... | Antenna No. | Read (MSB) | Read (LSB) | Write (MSB) | Write (LSB) | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: 1, Status, Default is 0x00, is reserved for later extensions;

2, a multi-channel device, the system default settings only 1 antenna power port, the other antenna ports needed to power 0 default setting, the user needs. Power to the antenna port 0 will employ the power of the antenna 1.

Example: Antenna 1 Read power30dBmWrite power 30dBmOther antennaPort power are0dB.

C8 8C 00 1d 13 00 01 0b b8 0b b8 02 00 00 00 00 03 00 00 00 00 04 00 00 00 00 0a 0d 0a

### Fixed frequency settings

Data: given frequency and a given number of frequencies point table.

Function:Fixed frequency operation of equipment, currently supports only one frequency.

Fixed frequency setting command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x14 | Fixed-frequency channel  The number of | Freq [1] (MSB) | Freq [1] |
| data | | | | | BCCcode | End of frame | |
| Freq [1] (LSB) | ... | Freq [n] (MSB) | Freq [n] | Freq [n] (LSB) | 0xxx | 0x0D | 0x0A |

Description:setFrequently point numberdefaultIs 1, Freq [1] represents the frequency of the fixed frequency. The unit is KHz frequency Freq

Example: Set920125KHz (0E0A3D)ofFixed-frequency

command:C8 8C 00 0C 14 01 0E 0A 3D 20 0D 0A

### Fixed-frequency response settings

Data: Set success: 0x01; setup failed: 0x00

Function: hopping answering.

Setting a fixed frequency response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x15 | ok: 0x01  fail: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: fixed frequency set successfully

command:C8 8C 00 09 15 01 1D 0D 0A

### Get current device setting state fixed frequency

Data: None

Function: Get the current device state and a fixed-frequency fixed-frequency table.

Get current device status setting command frame fixed frequency

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x16 | no | 0x1E | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: Get the current device status hopping configuration

command:C8 8C 00 08 16 1E 0D 0A

### Get the current fixed-frequency setting device status response

Data: fixed frequency point number and frequencies given in Table

Function: Get the device state and a fixed-frequency fixed-frequency table.

Obtaining current setting device fixed frequency response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x17 | Channel hopping  The number of | Freq [1] (MSB) | Freq [1] |
| data | | | | BCCcode | End of frame | |  |
| Freq [1] (LSB) | ... | Freq [n] (MSB) | Freq [n] (LSB) | 0xxx | 0x0D | 0x0A |

Description: Sets the number of frequency points 1, Freq [1] represents the frequency of the fixed frequency. The unit is KHz frequency Freq

Example: EquipmentFrequently set point 920125 (0E0A3D).

C8 8C 00 0C 17 01 0E 0A 3D twenty three 0D 0A

### Set up Gen2 parameters

Data: Session, Q, Coding and other settings

Function: gen2 parameters.

Set command frame gen2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0x10 | 0x20 | DByte7 | DByte6 | DByte5 |
| data | | | | | BCCcode | End of frame | |
| DByte4 | DByte3 | DByte2 | DByte1 | DByte0 | 0xxx | 0x0D | 0x0A |

Description: Data Members are defined in the following table

Data Definition Description

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DByte7 | | | | | | | | DByte6 | | | | | | | | DByte5 | | | | | | | | DByte4 | | | | | | | |
| Target | | | Action | | | T | Q | StartQ | | | | MinQ | | | | MaxQ | | | | D | C | | P | Sel | | Ses | | G | LF | | |
| x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| DByte3 | | | | | | | | DByte2 | | | | | | | | DByte1 | | | | | | | | Dbyte0 | | | | | | | |
| Reserve | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

1,Target setting: select command Target parameters

|  |  |
| --- | --- |
| S0 | B'000 |
| S1 | B'001 |
| S2 | B'010 |
| S3 | B'011 |
| SL | B'100 |

2,Action settings: select parameters command Action

|  |  |  |
| --- | --- | --- |
| Action | Matching | Non-Matching |
| 000 | assert SL or inventoried → A | de-assert SL or inventoried → B |
| 001 | assert SL or inventoried → A | do nothing |
| 010 | do nothing | de-assert SL or inventoried → B |
| 011 | negate SL or (A → B, B → A) | do nothing |
| 100 | de-assert SL or inventoried → B | de-assert SL or inventoried → A |
| 101 | de-assert SL or inventoried → B | do nothing |
| 110 | do nothing | de-assert SL or inventoried → A |
| 111 | do nothing | negate SL or (A → B, B → A) |

3,T Set: select Truncate command parameters

|  |  |
| --- | --- |
| Disable truncation | B'0 |
| Enable truncation | B'1 |

4,Q Setting:

|  |  |
| --- | --- |
| Static Q algorithm | B'0 |
| Q dynamic algorithm | B'1 |

Note: Fixed Q Algorithm,Q Fixed StartQ,ignore MinQ with MaxQ.

5,StartQ settings:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | B'0000 | 4 | B'0100 | 8 | B'1000 | 12 | B'1100 |
| 1 | B'0001 | 5 | B'0101 | 9 | B'1001 | 13 | B'1101 |
| 2 | B'0010 | 6 | B'0110 | 10 | B'1010 | 14 | B'1110 |
| 3 | B'0011 | 7 | B'0111 | 11 | B'1011 | 15 | B'1111 |

6,MinQ settings:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | B'0000 | 4 | B'0100 | 8 | B'1000 | 12 | B'1100 |
| 1 | B'0001 | 5 | B'0101 | 9 | B'1001 | 13 | B'1101 |
| 2 | B'0010 | 6 | B'0110 | 10 | B'1010 | 14 | B'1110 |
| 3 | B'0011 | 7 | B'0111 | 11 | B'1011 | 15 | B'1111 |

7,MaxQ settings:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | B'0000 | 4 | B'0100 | 8 | B'1000 | 12 | B'1100 |
| 1 | B'0001 | 5 | B'0101 | 9 | B'1001 | 13 | B'1101 |
| 2 | B'0010 | 6 | B'0110 | 10 | B'1010 | 14 | B'1110 |
| 3 | B'0011 | 7 | B'0111 | 11 | B'1011 | 15 | B'1111 |

8,Set D: DR parameter query command

|  |  |
| --- | --- |
| 8 | B'0 |
| 64/3 | B'1 |

9,Coding Set: M parameter query command

|  |  |
| --- | --- |
| FM0 | B'00 |
| Miller2 | B'01 |
| Miller4 | B'10 |
| Miller8 | B'11 |

10,Set P: query command parameters TRext

|  |  |
| --- | --- |
| No pilot tone | B'0 |
| Use pilot tone | B'1 |

11,sel Setting: query command parameters sel

|  |  |
| --- | --- |
| All | B'00 |
| All | B'01 |
| ~ SL | B'10 |
| SL | B'11 |

12,ses Setting: query command session parameters

|  |  |
| --- | --- |
| S0 | B'00 |
| S1 | B'01 |
| S2 | B'10 |
| S3 | B'11 |

13,Set G: Target parameter query command

|  |  |
| --- | --- |
| A | B'0 |
| B | B'1 |

14, LF Set up(Reserved).

example:Target Set as S1;Action for B'000;Truncate Parameters Disable Truncate;dynamicQ algorithm;startQ for 4;minQ for 0;maxQ for 15;DR forDR = 64/3;M Parameters Miller4;TRext ParametersUse pilot;sel Parameters for the ALL;Session Parameters S1;Target Parameters A.

command:

C8 8C 00 10 20 21 40 FD 53 00 00 00 00 FF 0D 0A

### Set up Gen2 response parameters

Data: Set success: 0x01; setup failed: 0x00

Function: gen2 parameters.

Gen2 parameter setting response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x21 | Ok: 0x01  fail: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

Example: Set gen2 Parameters success

command:C8 8C 00 09 twenty one 01 29 0D 0A

### Get the current Gen2 parameter settings

Data: None

Function: Get Device gen2 parameter settings.

Get the current parameters command frame gen2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x22 | no | 0x2A | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

Example: Gets the current device gen2 parameter settings

command:C8 8C 00 08 twenty two 2A 0D 0A

### Get the current Gen2 parameter setting response

Data: Session, Q, Coding and other settings

Function: Get Device Gen2 parameter settings.

Get the current Gen2 parameter response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | |
| 0xC8 | 0x8C | 0x00 | 0x10 | 0x23 | DByte7 | DByte6 | DByte5 |
| data | | | | | BCCcode | End of frame | |
| DByte4 | DByte3 | DByte2 | DByte1 | DByte0 | 0xxx | 0x0D | 0x0A |

Note: Data for your Gen2 parameter settings as defined in 3.2.9.

example:Target Set as S1;Action for B'000;Truncate Parameters Disable Truncate;dynamicQ algorithm;startQ for 4;minQ for 0;maxQ for 15;DR forDR = 64/3;M Parameters Miller4;TRext ParametersUse pilot;sel Parameters for the ALL;Session Parameters S1;Target Parameters A.

command:C8 8C 00 10 23 21 40 FD 53 00 00 00 00 FC 0D 0A

### CW Setting

Data: open CW: 0x01; Off CW: 0x00

Functions: opening or closing a continuous wave.

CW Set Command Frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x24 | Open: 0x01  Off: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

Example: open CW

command:C8 8C 00 09 twenty four 01 2C 0D 0A

### CW Setting the answer

Data: Set success: 0x01; setup failed: 0x00

Function: turn on or off a continuous wave response.

CW setting response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x25 | ok: 0x01  fail: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: Set success

command:C8 8C 00 09 25 01 2D 0D 0A

### Get the current equipment CW Setting

Data: None

Function: Gets the current state of the device CW

Gets the current device parameters CW command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Frame Type | data | Check code | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x26 | no | 0x2E | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

Example: Gets the current device CW Set up

command:C8 8C 00 08 26 2E 0D 0A

### Get the current equipment CW Setting the answer

Data: CW open: 0x01; CW OFF: 0x00

Function: Gets the current device CW status.

Gets the current device parameters CW response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x27 | Open: 0x01  Off: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

example:CW In the open state

command:C8 8C 00 09 27 01 2F 0D 0A

### Antenna Set

Data: total of 3 bytes, DByte2 whether the power-down save bytes; DByte1 and DByte0 bytes each corresponding to a common antenna 16, this bit is 1, chosen to correspond to the antenna, the bit is 0, the corresponding deselection antenna. After the antenna is selected when the inventory tag antenna is automatically selected in rotation.

Function: a single-port module default antenna, the other antenna is provided is not valid.

Command frame antenna arrangement

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0x0B | 0x28 | DByte2 | DByte1 | DByte0 |
| BCCcode | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

Description: Dbyte2 = 0x01, an antenna power-down save set, Dbyte2 = 0x00, indicates power-down is not saved settings.

Data bit is defined as follows:

The antenna defined set of data bits

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DByte1 | | | | | | | |
| Ant16 | Ant15 | Ant14 | Ant13 | Ant12 | Ant11 | Ant10 | Ant9 |
| DByte0 | | | | | | | |
| Ant8 | Ant7 | Ant6 | Ant5 | Ant4 | Ant3 | Ant2 | Ant1 |

##### Example: selecting the second antenna 14 and the antenna number is provided power-down save

command:C8 8C 00 0B 28 01 20 02 01 0D 0A

### Setting the answer antenna

Data: Set success: 0x01; setup failed: 0x00

Function: the antenna arrangement apparatus used

Antenna setting response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x29 | ok: 0x01  fail: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: Set success

command:C8 8C 00 09 29 01 twenty one 0D 0A

### Get the current antenna device settings

Data: None

Function: Get an antenna number for your device

Get command frame antenna arrangement

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x2a | no | 0x22 | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: Get current device antenna arrangement

command:C8 8C 00 08 2a twenty two 0D 0A

### Get the current antenna set the answering device

Data: total of 2 bytes, 16 bits, each bit corresponding to one antenna, the bit is 1, the corresponding antenna is selected, the bit is 0, the corresponding antenna is not selected.

Function: Get an antenna number for your device.

Gets antenna setting response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x2b | DByte1 | DByte0 | 0xxx |
| End of frame | |
| 0x0D | 0x0A |

Data bits are defined in Table:

The antenna defined set of data bits

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DByte1 | | | | | | | |
| Ant16 | Ant15 | Ant14 | Ant13 | Ant12 | Ant11 | Ant10 | Ant9 |
| DByte0 | | | | | | | |
| Ant8 | Ant7 | Ant6 | Ant5 | Ant4 | Ant3 | Ant2 | Ant1 |

Example: Current 1 No. antenna, the first 5 No. antenna, the first 10 No. antenna and 14 No. antenna was ordered: C8 8C 00 0A 2b twenty two 11 12 0D 0A

### Band locale

Data: 2 bytes

Function: Set the area.

Locale command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x2c | Save Settings  Mark | DByte0 | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: Save flag 0:00, do not save the settings, save set to 1, the default is the current region during the next boot. Data bits are defined in Table DByte0

|  |  |
| --- | --- |
| China1 (840MHz-845MHz) | 0x01 |
| China2 (920MHz-925MHz) | 0x02 |
| Europe (865MHz-868MHz) | 0x04 |
| USA (902MHz-928MHz) | 0x08 |
| Korea (917MHz-923MHz) | 0x16 |
| Japan (952MHz-953MHz) | 0x32 |

##### Example: save the settings, set area USA

command:C8 8C 00 0A 2C 01 08 2F 0D 0A

### Band locale response

Data: Set success: 0x01; setup failed: 0x00

Function: Set the area

Locale response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x2D | Ok: 0x01  fail: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: Set success

command:C8 8C 00 09 2D 01 25 0D 0A

### Gets band area

Data: None

Function: Get the equipment locale

Acquisition area setting command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x2E | no | 0x26 | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: Get Locale

command:C8 8C 00 08 2E 26 0D 0A

### Gets the locale band response

Data: 2 bytes

Function: Get the equipment locale

Gets the locale response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x2f | Ok: 0x01  fail: 0x00 | DByte0 | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: Data bits are defined in Table

|  |  |
| --- | --- |
| China1 | 0x01 |
| China2 | 0x02 |
| Europe | 0x04 |
| USA | 0x08 |
| Korea | 0x16 |
| Japan | 0x32 |

Example: current device locale China2

command:C8 8C 00 0A 01 2F 02 26 0D 0A

### Provided inventory data area

Data: AP to access password

MMB data area to be inventoried and combinations

MSA is the inventory data area start address(USERAreaeffective unitFor the word)

MDL is the length of visit(USERAreaeffectiveunitFor the word)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x30 | AP (MSB) | AP | AP |
| data | | | | | | BCCcode | frametail |
| AP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | 0xxx | 0x0D |
| frametail |  |
| 0x0A |

Explanation:MMB is defined as follows:

|  |  |
| --- | --- |
| 0x01 | EPC |
| 0x02 | TID |
| 0x03 | USER |
| 0x04 | Reversed |
| 0x05 | EPC + TID |
| 0x06 | EPC + USER |
| 0x07 | EPC +TID +USER |

Set upThe correspondingdataAfter the district,inventoryThe inventory data area corresponding to the label.

### Set inventory data area response

Set response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x31 | Ok: 0x01  fail: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: Set success

command:C8 8C 00 09 31 01 0xxx 0D 0A

### Get Device current temperature

Data: None

Function: Get the current temperature of the device, the maximum error of the temperature and the actual temperature values ​​is ± 3 ℃.

Get Device current temperature

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x34 | no | 0x3C | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: temperature acquiring device

command:C8 8C 00 08 34 3C 0D 0A

### Get the current temperature-responsive device

Data: Get the flag, 0x01 success, 0x00 failure. Temperature × 100, 2 bytes, unit ℃

Function: Gets the current device temperature response.

A current acquisition device temperature response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0x0B | 0x35 | Ok: 0x01  fail: 0x00 | temperature  (MSB) | temperature  (LSB) |
| BCCcode | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

Description: temperature × 100, converted to hexadecimal, then take a negative complement

Examples: succeed, the temperature of the devicetwenty two℃

command:C8 8C 00 0b 35 01 08 98 af 0d 0a

### Setting the temperature protection value

Data: 1 byte, the range of 50 ℃ -75 ℃, other value is invalid.

Function: Set the temperature protection options.

Setting the temperature protection

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x38 | Data0 | 0xXX | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description:whenModulecontinuousjobsHot,Temperature reaches the setWhen the valueWill intervals readerWithmaintainModuleSetting a temperature in the range of values.

##### Example: SetTemperature protectionIs 75 ℃

command:C8 8C 00 09 38 4B 7A 0D 0A

### Setting the temperature protection value of the response

Data: Set symbol of success, 0x01 success, 0x00 failure.

Function: Set the temperature protection response.

Setting the temperature protection response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x39 | Ok: 0x01  fail: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

Example: Set success

Command: C8 8C 00 09 39 01 31 0D 0A

### Gets temperature protection value

Data: None

Function: Gets temperature protection settings.

Gets temperature protection settings

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x3A | no | 0x32 | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description:no

##### Example: obtaining temperature protection is provided

command:C8 8C 00 08 3A 32 0D 0A

### Gets temperature protection value response

Data: Get the success flag, 0x01 success, 0x00 failure.

Function: Gets the value of temperature protection response.

Get response frame set temperature protection

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x3B | Ok: 0x01  fail: 0x00 | Data0 | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: Data0,forTemperature values

##### Example: to succeed,temperatureValue75 ℃

command:C8 8C 00 0A 3B 01 4B 7B 0D 0A

### An antenna working hours

Data: Byte 3

Function: Set the antenna working hours

An antenna working hours

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0x0B | 0x4A | DByte2 | DByte1 | DByte0 |
| BCCcode | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

Description: Low 4bit DByte2 of (bit0-bit3) an antenna number in the range 1-16, DByte2 of

bit4 indicates whether the power-down save, 0 to not save and 1 for power-down save. DByte1 DByte0 work and time, a total of 16 milliseconds, the range of 10ms ~ 65535ms. Note that the single antenna module, only an antenna 1, a multi-antenna module only supports setting other antennas.

Example: an antenna 3 Working hours 300ms, Power-down save settings. command:C8 8C 00 0B 4A 13 01 2C 7F 0D 0A

### An antenna working time response

Data: the success flag, 0x01 success, 0x00 failure.

Function: Set the antenna working time response

An antenna working time response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x4B | Ok: 0x01  fail: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

##### Example: Operation Success

command:C8 8C 00 09 4B 01 43 0D 0A

### Gets antenna working hours

Data: 2 bytes

Function: Get an antenna working hours.

Gets antenna working hours

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x4C | Ant num | Rev | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: ant num represents the antenna number.

##### Example: Get Antenna 1 hours

command:C8 8C 00 0A 4C 01 00 47 0D 0A

### Obtain an antenna response time

data:Four bytes.

Function: Get an antenna response time

Obtain an antenna response time frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0x0C | 0x4D | Ok: 0x01  fail: 0x00 | ant num | DByte1 |
| data | BCCcode | End of frame | |
| DByte0 | 0xxx | 0x0D | 0x0A |

Description: ant num denotes an antenna number, DByte1 DByte0 indicate corresponding antenna and working time.

Note: Single-antenna module, antenna can only get 1, multi-antenna modules only support for additional antennas.

Example: Get the antenna 2 The working time is successful, antenna 2 Working hours 400ms

command:C8 8C 00 0C 4D 01 02 01 90 D3 0D 0A

### Recommended settings RF links combination

data:Three bytes.

Function: Set Recommended RF link combination.

Recommended combinations of RF link provided

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0x0B | 0x52 | Rev | DByte1 | DByte0 |
| BCCcode | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

Description: DByte1 is 1, power-down save set to 0, no power-down save the settings. DByte0 arrangement shown in the following table (the default is 0x01, the best performance is provided).

|  |  |
| --- | --- |
| DByte0 | combination |
| 0x00 | DSB\_ASK / FM0 / 40 KHz |
| 0x01 | PR \_ASK / Miller4 / 250KHz |
| 0x02 | PR \_ASK / Miller4 / 300KHz |
| 0x03 | DSB\_ASK / FM0 / 400KHz |
| other | invalid |

Example: Set RF Link combination DSB\_ASK / FM0 / 40 KHz, Power-down does not save. command:C8 8C 00 0B 52 00 00 00 59 0D 0A

### Recommended settings RF transponder combination of links

Data: the success flag, 0x01 success, 0x00 failure.

Function: Set Recommended RF link response combination.

Recommended settings RF Link response frame combination

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x53 | Ok: 0x01  fail: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

##### Example: Set success

command:C8 8C 00 09 53 01 5B 0D 0A

### Get recommended RF link provided in combination

data:2 bytes.

Function: Get recommended RF link provided in combination.

Get recommended setting combination of RF link

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x54 | Rev | Rev | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: None.

##### Example: acquiring an RF link provided in combination Recommended

command:C8 8C 00 0A 54 00 00 5E 0D 0A

### Get recommended Response RF link provided in combination

data:Three bytes.

Function: Get recommended a combination of RF link setting response.

Get recommended RF Link setting response frame combination

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0x0B | 0x55 | Ok: 0x01  fail: 0x00 | Rev | DByte0 |
| BCCcode | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

Description: Sets the DByte0 shown in the table below

|  |  |
| --- | --- |
| DByte0 | combination |
| 0x00 | DSB\_ASK / FM0 / 40 KHz |
| 0x01 | PR \_ASK / Miller4 / 250KHz |
| 0x02 | PR \_ASK / Miller4 / 300KHz |
| 0x03 | DSB\_ASK / FM0 / 400KHz |
| other | invalid |

Example: The current recommendation RF Link combination DSB\_ASK / FM0 / 400KHz

command:C8 8C 00 0B 55 01 00 03 58 0D 0A

### Set up FastID function

data:2 bytes.

Feature: open or closed FastID function.

Set FastID

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x5C | ON: 1  OFF: 0 | Rev | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: On: 0x01, closed: 0x00.

Example: open FastID Features.

command:C8 8C 00 0A 5C 01 00 57 0D 0A

### Set up Answer the FastID

Data: a byte.

Function: FastID function response.

Set FastID response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x5D | Ok: 0x01  fail: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

##### Example: Set success

command:C8 8C 00 09 5D 01 55 0D 0A

### Obtain FastID functional status

Data: 2 bytes

Function: Gets the current reader FastID state, is turned on.

Get FastID state

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x5E | Rev | Rev | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: None.

Example: Get FastID status.

command:C8 8C 00 0A 5E 00 00 54 0D 0A

### Obtain FastID functional status response

Data: two bytes.

Function: Get FastID state response.

Get FastID state response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x5F | OK: 1  Fail: 0 | ON: 1  OFF: 0 | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Example: to succeed, the current FastID Function is enabled by state

command:C8 8C 00 0A 5F 01 01 55 0D 0A

### Set Tagfocus Features

Data: 2 bytes

Feature: open or closed TagFocus Features.

Set TagFocus

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x60 | ON: 1  OFF: 0 | Rev | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: On: 0x01, closed: 0x00.

Example: open TagFocus Features.

command:C8 8C 00 0A 60 01 00 6B 0D 0A

### Set TagFocus Answer the

Data: a byte.

Function: Set TagFocus Function response.

Set TagFocus response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x61 | Ok: 0x01  fail: 0x00 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

##### Example: Set success

command:C8 8C 00 09 61 01 69 0D 0A

### Get TagFocus Functional status

Data: 2 bytes.

Function: Gets the current reader TagFocus state, is turned on.

Get TagFocus state

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x62 | Rev | Rev | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: None.

Example: Get TagFocus status.

command:C8 8C 00 0A 62 00 00 68 0D 0A

### Get TagFocus Functional status response

Data: two bytes.

Function: Get FastID state response.

Get TagFocus state response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x63 | OK: 1  Fail: 0 | ON: 1  OFF: 0 | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Example: to succeed, the current TagFocus Function is enabled by state

command:C8 8C 00 0A 63 01 01 69 0D 0A

### Software reset

Data: Byte 0

Function: software reset module.

Software reset

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x68 | no | 0x60 | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: send a software reset command, you can reset the reader.

##### Example: sending software reset command.

command:C8 8C 00 08 68 60 0D 0A

### Software reset response

data:A byte.

Function: software reset response.

Software reset response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x69 | OK: 1  Fail: 0 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None.

##### Example: a successful reset.

command:C8 8C 00 09 69 01 61 0D 0A

### Look for the label filter settings

data:n bytes.

Function: to find a label with a range tag group selection.

Look for the label filter settings

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0xxx | 0x6E | DByte0 | MMB | MSA (MSB) |
| data | | | | | | | |
| MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | ... | ... | ... | MData (LSB) |
| BCCcode | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

Description: DByte0: 0x01 indicates power-down setting values ​​are saved, not saved to 0x00 represents;

MMB : Filtering operationbank number, 0x01 represents EPC, 0x02 represents TID, 0x03 represents USR, another value is an invalid value;

MSA : Starting address Start filtering operation, the unit isbit;

MDL : Filtering operation starts filtering data length, inbit, 0x00 indicates no filter; MData: Start filtering data, in bytes, ifMDL is less than an integral multiple of bytes, less than the low order bits 0s.

example 1: Set Hunt tag filtering rules:TID District filtration, filter address TID Zone 0bit, Filter length 96bit, Filtered data 0xE2003414013301001038D2B5, Power-down save the filter settings.

command:C8 8C 00 1A 6E 01 02 00 00 00 60 E2 00 34 14

01 33 01 00 10 38 D2 B5 A9 0D 0A

##### Example 2: Set Hunt Tags filtering rules: no filters, power-down save the filter settings.

command:C8 8C 00 0E 6E 01 00 00 00 00 00 61 0D 0A

### Find answer label filter settings

data:A byte.

Function: label filtering settings to find a response.

Searching tag response frame filter settings

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x6F | OK: 1  Fail: 0 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None.

##### Example: Set success

command:C8 8C 00 09 6F 01 67 0D 0A

### EPC+TIDOr EPC + USER Mode setting

Data: 4 bytes

Function: Enables simultaneous reading EPC + TIDorEPC + TID + USER mode when the reader during continuous searching tag, the tag will be read simultaneously EPC + TIDorEPC + USER data.

Mode setting

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | framelength | | Type CMD | data | | |
| 0xC8 | 0x8C | 0x00 | 0x0C | 0x70 | Dbyte0 | Memory | Adress |
| data | BCCcode | End of frame | |
| Lenth | 0xxx | 0x0D | 0x0A |

Description: Dbyte0: 0x01 indicates power-down setting values ​​are saved, not saved to 0x00 represents;

Memory: Is 0x00, indicating that close; 0x01, indicate on EPC + TID mode(silentIdentified address 0x00, a length of 6 characters); 0x02, indicate onEPC + TID + USER mode

Adress: For the USER areaStarting address(unitFor the word).

Lenth :forUSER areaofLength (UnitFor the word).

### EPC+TIDOr EPC +TID +USER Setting the answer mode

Data: 4 bytes.

Function: read simultaneously EPC + TID or EPC + TID + USER mode setting response.

Mode setting response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x71 | OK: 1  Fail: 0 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None.

##### Example: Set success

command:C8 8C 00 09 71 01 79 0D 0A

### Read EPC+TIDOr EPC +TID +USER Mode status

Data: 2 bytes

Function: Gets the current reader simultaneously reads EPC + TID or EPC + TID + USER mode set state, it is turned on.

Reading mode set state

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x72 | Rev | Rev | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: None.

Example: acquiring read simultaneously EPC + TIDOr EPC + TID + USER mode set state.

command:C8 8C 00 0A 72 00 00 78 0D 0A

### Read EPC+TIDOr EPC +TID +USER Answer mode status

Data: 4 bytes.

Function: Get simultaneously read EPC+TIDOr EPC +TID +USER Mode setting state response.

Acquisition mode set state response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | framelength | | Type CMD | data | | |
| 0xC8 | 0x8C | 0x00 | 0x0C | 0x73 | OK: 1  Fail: 0 | Memory | Adress |
| data | BCCcode | End of frame | |
| Lenth | 0xxx | 0x0D | 0x0A |

### reset

Data: Byte 0

Function: restore factory settings.

reset

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x74 | no | 0x7C | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: send commands to restore factory settings, you can restore the settings of the reader, restore factory settings successfully, the reader will automatically reset. After this command is executed, the set value may vary as follows:

|  |  |  |
| --- | --- | --- |
| Settings | Restore factory settings value | Remark |
| Transmission power | 30dBm |  |
| Temperature protection settings | Open |  |
| Recommended combination of RF link | PR \_ASK / Miller4 / 250KHz |  |
| Buzzer state | Not ringing |  |
| FastID function | shut down |  |
| TagFocus function | shut down |  |
| Module baud rate | 115200 |  |
| Look for the label filter settings | Filtered data length = 0 | Do not enable filtering means that the process of seeking label  Filter mechanism |
| While readingEPC and TID mode  formula | shut down |  |
|  |  |  |

##### Example: send the command to restore the factory settings.

command:C8 8C 00 08 74 7C 0D 0A

### Factory Reset response

data:A byte.

Function: restore factory settings response.

Factory Reset response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x75 | OK: 1  Fail: 0 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None.

##### Example: restore factory settings successfully.

command:C8 8C 00 09 75 01 7D 0D 0A

## Label operation

### Single inventory tags

Data: Timeou (t endian), timeout, milliseconds, or if the card has successfully find Timeout

Time to, RFID module must return a response frame.

Function: to find the label, if find a label, returns only one label.

Single inventory label command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x80 | Timeout (MSB) | Timeout (LSB) | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: None

##### Example: look for labels

command:C8 8C 00 0A 80 00 64 EE 0D 0A

### Single label inventory response

Data: PC + EPC, RSSI.

Function: label inventory response, the tag and reader-related information return.

Single label inventory response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0xxx | 0x81 | PC (MSB) | PC (LSB) | EPC (MSB) |
| data | | | | | | | |
| EPC | EPC | EPC | EPC | EPC | EPC | EPC | EPC |
| data | | | | | | BCCcode | End of frame |
| EPC | EPC | EPC (LSB) | RSSI (MSB) | RSSI (LSB) | Ant Num | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: RSSI represented as a complement, a total 16bit, the actual value × 10. As -65.7dBm, the

RSSI = FD6F.

Note: The length of the EPC decided to have PC, this is based on the Gen2 protocol, so frame length is not fixed. After FastID function is enabled, attempting to read the label TID data, the EPC (LSB) response frame will increase the TID 96bit data, and is the RSSI value.

Example: Label PC = 0x3000 , EPC = 0xE2003411B802011383258566 Response,

RSSI = -65.7dBm,antenna 2 To inventory.

command:C8 8C 00 19 81 30 00 E2 00 34 11 B8 02 01 13

83 25 85 66 FD 6F 02 12 0D 0A

### Continuous inventory labels

Data: continuous inventory tag number, a total of two sections.

Features: continuous inventory labels.

Continuous inventory tag command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x82 | Num [1] | Num [0] | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: Continuous inventory tag number range1 ~ 0xFFFF, the number is 0, an unlimited number of inventory tags.

Note: During continuous inventory tag, the reader does not respond to other commands, execute For other commands to be sentStop the continuous inventory label command, waiting for a response after stopping continuous inventory tags, and then send commands to be executed**.**

Example: the number of continuous inventory tag10000 (0x2710)Secondary

command:C8 8C 00 0A 82 27 10 BF 0D 0A

### Continuous label inventory response

Data: PC + EPC, RSSI, antenna number.

Function: label inventory response, the tag and reader-related information return.

Continuous label inventory response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0xxx | 0x83 | PC (MSB) | PC (LSB) | EPC (MSB) |
| data | | | | | | | |
| EPC | EPC | EPC | EPC | EPC | EPC | EPC | EPC |
| data | | | | | | BCCcode | End of frame |
| EPC | EPC | EPC (LSB) | RSSI (MSB) | RSSI (LSB) | Ant Num | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: RSSI represented as a complement, a total 16bit, the actual value × 10. As -65.7dBm, the

RSSI = FD6F.

Note: The length of the EPC decided to have PC, this is based on the Gen2 protocol, so frame length is not fixed.in FastID functionOr TID or the EPC + EPC + USER function is enabled, if the read tagTID data, the EPC (LSB) response frame will increase the TID USER data area or region, and only thenRSSI value.

Example: Label PC = 0x3000 , EPC = 0xE2003411B802011383258566 Response,

RSSI = -65.7dBm,antenna 2 To inventory.

command:C8 8C 00 19 83 30 00 E2 00 34 11 B8 02 01 13

83 25 85 66 FD 6F 02 10 0D 0A

### Stop continuous inventory labels

Data: None

Function: stop the continuous inventory labels.

Stop the continuous inventory label command

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x08 | 0x8C | no | 0x84 | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

##### Example: stop continuous inventory tag

command:C8 8C 00 08 8C 84 0D 0A

### Stop the continuous label inventory response

Data: flags flag: success: 0x01; failure: 0x00

Function: stop the continuous label inventory response

Stop the continuous label inventory response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x8D | Flag | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None

##### Example: Success

command:C8 8C 00 09 8D 01 85 0D 0A

### Read the label data area

Data: AP (access code), MMB, MSA, MDL, MData, memory Bank, SA start address (words), the data length DL is to be read (word units).Word length of 2 bytes.

Function: data read tag specified data area.

Read Data command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x84 | AP (MSB) | AP | AP |
| data | | | | | | | |
| AP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | MData |
| data | | | | | | | |
| MData | ... | ... | ... | ... | MData | MData (LSB) | MB |
| data | | | | BCCcode | End of frame | |  |
| SA (MSB) | SA (LSB) | DL (MSB) | DL (LSB) | 0xxx | 0x0D | 0x0A |

Description:

AP:4Byte access password

MMB: Mask of the data area(0x00for Reserve 0x01 forEPC, 0x02 represents the TID, 0x03 represents USR).

MSA: To mask the address.

MDL: Length mask.

Mdata: To mask data.

MB: To write data area(0x00for Reserve 0x01 forEPC, 0x02 represents the TID, 0x03 represents USR).

SA :forToAddress write data area.

DL :Length of data to be written(Word units).

Data : To write data, the previous high.

Example 1: No filtering, the read data word TID area 3, the starting address is 2, access password 0x55555555

command:C8 8C 00 16 84 55 55 55 55 00 00 00 00 00 02

00 02 00 03 91 0D 0A

example 2:TID District filtration, filter address TID Zone 2bit, Filter length 13bit, Filtered data 1110001000000'bRead EPC Area 6 Data word, the starting address is 2Access password0x00000000

command:C8 8C 00 18 84 00 00 00 00 02 00 02 00 0D E2

00 01 00 02 00 06 76 0D 0A

### Read label data area response

Data: Flag: Flag read data is successful, success: 0x01; failure: 0x00

Errflag: error flag, After the failure of the error flag returned

DL: readingThe length of the data unit is a word

Data:readData

Function: read data return.

Read data response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x85 | Flag | Errflag | DL (MSB) |
| data | | | | | Check code | End of frame | |
| DL (LSB) | Data (MSB) | ... | ... | Data (LSB) | 0xxx | 0x0D | 0x0A |

Description: Reads data Data read data is determined by the length of the command data, in addition, read data is fail, then no response frame data Data; read data, the error flag Errflag to 0x00. Read operation fails,

ErrFlag prompt type of failure, ErrFlag of 1, indicating no label; ErrFlag is 2, represents an access password is wrong; ErrFlag 3, showing Read operation failed.

Example: Successful read TID Area 3 Words of data:0x123456789ABC

command:C8 8C 00 12 85 01 00 00 03 12 34 56 78 9A BC BB 0D 0A

### Write tag data area

Function: the data into the specified storage area.

Write Data command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x86 | AP (MSB) | AP | AP |
| data | | | | | | | |
| AP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | MData |
| data | | | | | | | |
| MData | ... | ... | ... | ... | MData | MData (LSB) | MB |
| data | | | | | | | |
| SA (MSB) | SA (LSB) | DL (MSB) | DL (LSB) | Data (MSB) | ... | ... | Data (LSB) |
| BCCcode | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

AP:4Byte access password

MMB: Mask of the data area(0x00for Reserve 0x01 forEPC, 0x02 represents the TID, 0x03 represents USR).

MSA: To mask the address.

MDL: Length mask.

Mdata: To mask data.

MB: To write data area(0x00for Reserve 0x01 forEPC, 0x02 represents the TID, 0x03 represents USR).

SA :forToAddress write data area.

DL :Length of data to be written(Word units).

Data : To write data, the previous high.

example 1: Do not filter, write EPC Area 6 Words of data,Data = 0x00112233445566778899 aabbStarting at address 2Access password 0x12345678

command:C8 8C 00 twenty two 86 12 34 56 78 00 00 00 00 00 01

00 02 00 06 00 11 twenty two 33 44 55 66 77 88 99 AA BB A9

0D 0A

example 2:TID District filtration, filter address TID Zone 0bit, Filter length 96bit, Filtered data 0xE2003414013301001038D2B5Write EPC Area 6 Words of data,Data = 0x00112233

445566778899aabbStarting at address 2Access password 0x00000000

command:C8 8C 00 2E 86 00 00 00 00 02 00 00 00 60 E2

00 34 14 01 33 01 00 10 38 D2 B5 01 00 02 00 06 00

11 twenty two 33 44 55 66 77 88 99 AA BB 71 0D 0A

### Write data area response

Data: Flag: write data is successful sign, success: 0x01; failure: 0x00

Errflag: error flagError flag returned after a failed write

Function: write data response

Write data response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x87 | Flag | Errflag | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: write data, the error flag Errflag to 0x00. Write operation failed,ErrFlag prompt type of failure, ErrFlag of 1, indicating no label; ErrFlag is 2, represents an access password is wrong; ErrFlag 3, showing Write operation failed.

Example: Data write failure, the error flag is 0x03

command:C8 8C 00 0A 87 00 03 8E 0D 0A

### Locklabel

Data: AP (access code), MMB, MSA, MDL, MData, LD (3 bytes total)

Function: Lock memory bank labels

lock tag command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0xxx | 0x88 | AP (MSB) | AP | AP |
| data | | | | | | | |
| AP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | MData |
| data | | | | | | | |
| MData | ... | ... | ... | ... | MData | MData (LSB) | LD (MSB) |
| data | | BCCcode | End of frame | |  | | |
| LD | LD (LSB) | 0xxx | 0x0D | 0x0A |

Description:

AP:4Bytelock password

MMB: Mask of the data area(0x00for Reserve 0x01 forEPC, 0x02 represents the TID, 0x03 represents USR).

MSA: To mask the address.

MDL: Length mask.

Mdata: To mask data.

LD: 24bit total of 3 bytes, wherein the high 4bit invalid, the first 0 ~ 9bit (co 10bit) Action of bits, the first 10 ~ 19bit (co 10bit) of mask bits,For details, seeISO18000-6C protocol manual.

Example: filtering region TID, TID Zone address filtering 0bit filtered 96bit length, filtering data 0xE2003414013301001038D2B5, EPC + RFU locking zone (LD = 0x0FC2A0), access password 0x760039AD

Command: C8 8C 002 088 76 00 39 AD 02 00 00 00 60 E2

0034 140133 01 00 10 38 D2 B5 0F C2 A0 FB 0D 0A

### LockTag replies

Data: Lock label success symbol flag: success: 0x01; failure: 0x00

Error flag Errflag: error flag returned after the failure of the lock tag

Function: Lock labels response.

Lock tag response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x89 | Flag | Errflag | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: Lock operation is successful, the error flagErrflag to 0x00. Lock operation fails, Errflag prompt type of failure, Errflag of 1, indicating no label; Errflag 2, represents an access password is wrong; Errflag 3, represents the Lock operation failed.

##### Example: Lock success

command:C8 8C 00 0A 89 01 00 82 0D 0A

### Kill label

Data: KP (kill password), MMB, MSA, MDL, MData

Function: kill tag

kill command frame label

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0x00 | 0xxx | 0x8A | KP (MSB) | KP | KP |
| data | | | | | | | |
| KP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | MData |
| data BCCcode | | | | | | | BCC code |
| MData | ... | ... | ... | ... | MData | MData (LSB) | 0xxx |
| End of frame | |
| 0x0D | 0x0A |

Description:

KP:4KILL-byte password

MMB: Mask of the data area(0x00for Reserve 0x01 forEPC, 0x02 represents the TID, 0x03 represents USR).

MSA: To mask the address.

MDL: Length mask.

Mdata: To mask data.

When the label 0x00000000 value is KillPwd zone, the label ignores the kill command, kill command will not succeed

example:EPC District filtration, filter address EPC Zone 32bit, Filter length 96bit, Filtered data 0x00112233445566778899AABB,kill Password 0x760039AD

command:C8 8C 00 1D 8A 76 00 39 AD 01 00 20 00 60 00

11 22 33 44 55 66 77 88 99 AA BB 34 0D 0A

### Kill label reply

Data: kill label success symbol flag: success: 0x01; failure: 0x00

Error flag Errflag: error flag label failed to return after the kill

Function: kill tag replies

kill tag response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | Type CMD | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x8B | Flag | Errflag | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: kill operation is successful, the error flagErrflag to 0x00. kill operation failed, suggesting ErrFlag type of failure, ErrFlag of 1, indicating no label; ErrFlag 3, showing the kill operation failed

example:kill success

command:C8 8C 00 0A 8B 01 00 80 0D 0A

### Block Write Data

Data: AP (access code), MMB, MSA, MDL, MData, memory Bank, SA start address (word unit), needs to be written in the data length DL (words) of the data needs to be written in the Data

Function: Data Block Write tag to a specific length of a specific address.

Block Write command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x93 | AP (MSB) | AP | AP |
| data | | | | | | | |
| AP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | MData |
| data | | | | | | | |
| MData | ... | ... | ... | ... | MData | MData (LSB) | MB |
| data | | | | | | | |
| SA (MSB) | SA (LSB) | DL (MSB) | DL (LSB) | Data (MSB) | ... | ... | Data (LSB) |
| BCCcode | End of frame | |
| 0xxx | 0x0D | 0x0A |

Description:

AP:4Byte access password

MMB: Mask of the data area(0x00for Reserve 0x01 forEPC, 0x02 represents the TID, 0x03 represents USR).

MSA: To mask the address.

MDL: Length mask.

Mdata: To mask data.

MB: To write data area(0x00for Reserve 0x01 forEPC, 0x02 represents the TID, 0x03 represents USR).

SA :forToAddress write data area.

DL :Length of data to be written(Word units).

Data : To write data, the previous high.

Example: Label PC = 0x3000,EPC = 0xE2003411B802011383258566Write EPC Area 6

Words of data,Data = 0x00112233445566778899aabbStarting at address 2Access password0x74290fd8

command:C8 8C 00 2B 93 74 29 0f d8 30 00 E2 00 34 11

B8 02 01 13 83 25 85 66 01 00 02 00 06 00 11 twenty two 33

44 55 66 77 88 99 AA BB 2D 0D 0A

### Block Write response data

Data: Block Write data is successful logo flag: success: 0x01; failure: 0x00

Error flag Errflag: error flag returned after a failed operation.

Features: Block Write data response.

Block Write response frame data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x94 | Flag | Errflag | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: The operation is successful, the error flag Errflag to 0x00. Read data, the error flag Errflag

for 0x00. Blockwrite operation failed, suggesting ErrFlag type of failure, ErrFlag of 1, indicating no label;

Errflag 2, represents an access password is wrong; Errflag 3, that the implementation of Blockwrite command failed.

Example: Data write failure, the error flag is 0x01

command:C8 8C 00 0A 94 00 01 9F 0D 0A

### Block Erase Data

Data: AP (access code), MMB, MSA, MDL, MData, memory Bank, SA start address (word units), DL erased required length (word units)

Function: Block Erase specific length to a specific address tag.

Block Erase command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x95 | AP (MSB) | AP | AP |
| data | | | | | | | |
| AP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | MData |
| data | | | | | | | |
| MData | ... | ... | ... | ... | MData | MData (LSB) | MB |
| data | | | | BCCcode | End of frame | |  |
| SA (MSB) | SA (LSB) | DL (MSB) | DL (LSB) | 0xxx | 0x0D | 0x0A |

Description:

AP:4Byte access password

MMB: Mask of the data area,0x01 forEPC, 0x02 represents the TID, 0x03 represents USR.

MSA: To mask the address.

MDL: Length mask.

Mdata: To mask data.

MB: To write data area.

SA :forToAddress write data area.

DL :Length of data to be written(Word units).

Example: Label PC = 0x3000, EPC = 0xE2003411B802011383258566, Erase EPC tag

Area 6 Data word, the starting address is 2Access password 0x74290fd8

command:C8 8C 00 1F 95 74 29 0f d8 30 00 E2 00 34 11

B8 02 01 13 83 25 85 66 01 00 02 00 06 1F 0D 0A

### Block Erase data response

Data: Block Erase data is successful logo flag: success: 0x01; failure: 0x00

Error flag Errflag: error flag returned after a failed operation.

Features: Block Erase data response.

Block Erase data response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x96 | Flag | Errflag | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Explanation: The operation was successful, an error flag Errflag to 0x00. Read data, the error flag Errflag to 0x00. Block Erase operation fails, the failure prompt type ErrFlag, ErrFlag of 1, indicating no label; ErrFlag is 2, represents an access password is wrong; ErrFlag 3, showing performs Block Erase operation failed.

Example: Data Erase Successful, the error flag is 0x00

command:C8 8C 00 0A 96 01 00 9D 0D 0A

### Set up QT parameters

Data: AP (access code), MMB, MSA, MDL, MData, QTData

Function: Set the QT command parameters.

Set command parameters QT

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x97 | AP (MSB) | AP | AP |
| data | | | | | | | |
| AP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | MData |
| data | | | | | | | |
| MData | ... | ... | ... | ... | MData | MData (LSB) | QTData |
| BCCcode | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

Note: Only support QT label command to respond to the command.

AP : In order to access the labelpassword.

MMB : To start filtering operationbank number, 0x01 represents EPC, 0x02 represents TID, 0x03 represents USR, other valuesinvalid.

MSA : Starting address for the filtering operation, the unit isbit.

MDL : To filter the data length inbit, 0x00 indicates no filter.

Mdata: Start filtering data, in bytes, ifMDLIt is not an integer multiple of bytes, the low complement0.

QTData :QTData high 6bit are reserved, bit0 0 indicates no close control, bit0 to 1 to enable close control;

bit1 Enable Private Memory Map 0 indicates the label, bit1 1 indicates tab to enable Public Memory Map.

example:TID District filtration, filter address TID Zone 2bit, Filter length 13bit, Filtered data1110001000000'bAccess password 0x00000000Enable Private Memory Map, Enable close control.

command:C8 8C 00 14 97 00 00 00 00 02 00 02 00 0D E2

00 01 6D 0D 0A

### Set up QT parameter response

Data: a byte.

Function: QT command parameter response.

QT parameter setting response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | BCCcode | End of frame |
| 0xC8 | 0x8C | 0x00 | 0x09 | 0x98 | OK: 1  Fail: 0 | 0xxx | 0x0D |
| End of frame |  | | | | | | |
| 0x0A |

Description: None.

##### Example: Set succeeded.

command:C8 8C 00 09 98 01 90 0D 0A

### Obtain QT parameters

Data: Byte 0

Function: Gets tag QT command parameters.

Get command parameter QT

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x99 | AP (MSB) | AP | AP |
| data | | | | | | | |
| AP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | MData |
| data | | | | | | | |
| MData | ... | ... | ... | ... | ... | MData | MData (LSB) |
| BCCcode | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

Description: Get tag QT command parameters. Only supports QT command tag to respond to the command.

example:TID District filtration, filter address TID Zone 2bit, Filter length 13bit, Filtered data1110001000000'bAccess password 0x00000000.

command:C8 8C 00 12 99 00 00 00 00 02 00 02 00 0D E2

00 64 0D 0A

### Obtain QT parameter response

Data: two bytes.

Function: Get QT command parameter response.

QT parameters to obtain response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x9A | OK: 1  Fail: 0 | DByte0 | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: DByte0 high 6bit reserved bits, bit0 close to 0 indicates no control, bit0 is 1 enables close control; bit1 0 indicates tags enabled Private Memory Map, bit1 1 represents tags to enable Public Memory Map.

Example: Enable Private Memory Map, Enable close control. command:C8 8C 00 0A 9A 01 01 90 0D 0A

### QTlabelRead

Data: AP (access code), MMB, MSA, MDL, MData, MB, SA, DL

Function: QT read operation, read the label Private Memory Map data read operation is complete, regardless of success or failure of the operation, the label automatically return to the state before the QT memory map read.

Set command parameters QT

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x9B | AP (MSB) | AP | AP |
| data | | | | | | | |
| AP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | MData |
| data | | | | | | | |
| MData | ... | ... | ... | MData | MData (LSB) | QTData | MB |
| data | | | | BCCcode | End of frame | |  |
| SA (MSB) | SA (LSB) | DL (MSB) | DL (LSB) | 0xxx | 0x0D | 0x0A |

Note: Only support QT label command to respond to the command.

AP : In order to access the labelpassword.

MMB : To start filtering operationbank number, 0x01 represents EPC, 0x02 represents TID, 0x03 represents USR, other valuesinvalid.

MSA : Starting address for the filtering operation, the unit isbit.

MDL : To filter the data length inbit, 0x00 indicates no filter.

Mdata: Start filtering data, in bytes, ifMDLIt is not an integer multiple of bytes, the low complement0.

QTData :high7bit are reserved, bit0 0 indicates no close control, bit0 to 1 to enable close control.

MB :memory bank, the user needs to write data bank number

SA : Starting address for the data needs to be written, in units of word.

DL : Data length needs to be written in units of the word.

example:TID District filtration, filter address TID Zone 2bit, Filter length 13bit, Filtered data1110001000000'bRead EPC Area 6 Data word, the starting address is 2Access password

0x00000000, Close Read

command:C8 8C 00 19 9B 00 00 00 00 02 00 02 00 0D E2

00 01 01 00 02 00 06 69 0D 0A

### QTlabelRead answer

Data: Flag: Flag read data is successful, success: 0x01; failure: 0x00

Errflag: error flagThe error flag is returned after the failure of the read data.

DL: length of read dataThe unit is the wordRead data: Data

Features: QT tag read response.

QT tag read response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x9C | Flag | Errflag | DL (MSB) |
| data | | | | | BCCcode | End of frame | |
| DL (LSB) | Data (MSB) | ... | ... | Data (LSB) | 0xxx | 0x0D | 0x0A |

Description: Reads data Data read data is determined by the length of the command data, in addition, read data is fail, then no response frame data Data; read data, the error flag Errflag to 0x00. Read operation fails,

ErrFlag prompt type of failure, ErrFlag of 1, indicating no label; ErrFlag is 2, represents an access password is wrong; ErrFlag 3, showing operation failed.

Example: Successful read TID Area 3 Words of data:0x123456789abc

command:C8 8C 00 12 9C 01 00 00 03 12 34 56 78 9A BC A2 0D 0A

### QTlabelWrite

Data: AP (access code), MMB, MSA, MDL, MData, memory Bank, SA start address (word unit), needs to be written in the data length DL (words) of the data needs to be written in the Data

Function: QT write operation, data is written to a specific length of the label Private Memory Map, a specific address, after the write operation is complete, regardless of success or failure of the operation, the label automatically return to the state before the QT memory map read.

QT label write command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x9D | AP (MSB) | AP | AP |
| data | | | | | | | |
| AP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | MData |
| data | | | | | | | |
| MData | ... | ... | ... | MData | MData | QTData | MB |
| data | | | | | | | |
| SA (MSB) | SA (LSB) | DL (MSB) | DL (LSB) | Data (MSB) | ... | ... | Data (LSB) |
| BCCcode | End of frame | |
| 0xxx | 0x0D | 0x0A |

Note: Only support QT label command to respond to the command.

AP : In order to access the labelpassword.

MMB : To start filtering operationbank number, 0x01 represents EPC, 0x02 represents TID, 0x03 represents USR, other valuesinvalid.

MSA : Starting address for the filtering operation, the unit isbit.

MDL : To filter the data length inbit, 0x00 indicates no filter.

Mdata: Start filtering data, in bytes, ifMDLIt is not an integer multiple of bytes, the low complement0.

QTData :high7bit are reserved, bit0 0 indicates no close control, bit0 to 1 to enable close control.

MB :memory bank, the user needs to write data bank number

SA : Starting address for the data needs to be written, in units of word.

DL : Data length needs to be written in units of the word.

Data : To write data, the previous high.

example:TID District filtration, filter address TID Zone 0bit, Filter length 96bit, Filtered data0xE2003414013301001038D2B5Write EPC Area 6 Words of data,Data = 0x00112233

445566778899aabbStarting at address 2Access password 0x00000000, Close-write operation

command:C8 8C 00 2F 9D 00 00 00 00 02 00 00 00 60 E2

00 34 14 01 33 01 00 10 38 D2 B5 01 01 00 02 00 06 00 11 22 33 44 55 66 77 88 99 AA BB 6A 0D 0A

### QTWrite transponder tag

Data: Data is successful write flag flag: success: 0x01; failure: 0x00

Error Flag Errflag: write error flag return data after a failure.

Function: write data response.

QT label write response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0x0A | 0x9E | Flag | Errflag | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: write data, the error flag Errflag to 0x00. The write operation failed, Errflag prompt type of failure, Errflag of 1, indicating no label; Errflag 2, represents an access password is wrong; Errflag 3, represents the Write operation failed.

Example: Data write failure, the error flag is 0x03

command:C8 8C 00 0A 9E 00 03 97 0D 0A

### Block Permalock operation

Data: AP (access code), MMB,MSA, MDL, MData, ReadLock, MemBank,

BlockPtr, BlockRange, Mask.

Function: BlockPermalock operation.

Block Permalock operation command frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | |
| 0xC8 | 0x8C | 0xxx | 0xxx | 0x9F | AP (MSB) | AP | AP |
| data | | | | | | | |
| AP (LSB) | MMB | MSA (MSB) | MSA (LSB) | MDL (MSB) | MDL (LSB) | MData (MSB) | MData |
| data | | | | | | | |
| MData | ... | ... | ... | MData | MData (LSB) | ReadLo ck | MB |
| data | | | | | | | |
| BlockPt r (MSB) | BlockPt r (LSB) | BlockRa nge(MS B) | BlockRa nge(LSB  ) | Mask  (MSB) | ... | ... | Mask  (LSB) |
| BCCcode | End of frame | |  | | | | |
| 0xxx | 0x0D | 0x0A |

Note: Only support Label Block Permalock order to respond to the command.

AP: Labelaccesspassword .

MMB : Start the filter operationbank No.,0x01forEPC0x02For the TID, 0x03As USER area,No other values.

MSA:Start address to start the filtering operation, the unit is bit.

MDL : Filtering operation starts filtering data length, inbit, 0x00 indicates no filter.

Mdata: Start filtering data, in bytes, ifMDLIt is not an integer multiple of bytes, the low complement0.

ReadLock: High 7bit reserved bits, bit0 indicates 0 Read, bit0 indicates Permalock 1.

MB:formemory bank, To operate the data area.

BlockPtr :Block start address is, in units of 16 blocks,A block of 8 bytes.

BlockRange :block range, the unit block of 16

Mask: Mask data block, the previous high, whether the two bytes corresponding to 16 16-bit block selection.

example:TID District filtration, filter address TID Zone 0bit, Filter length 96bit, Filtered data0xE2003414013301001038D2B5,readlock = 0,MB = 3,BlockPtr = 0, BlockRange = 1, Access password 0x00000000

command:C8 8C 00 twenty three 9F 00 00 00 00 02 00 00 00 60 E2

00 34 14 01 33 01 00 10 38 D2 B5 00 03 00 00 00 01

62 0D 0A

### Block Permalock operating answer

Data: Flag NO symbol of success, success: 0x01; failure: 0x00;

ErrflagFor the error flag,Error flag returned after a failed operation.

Features: Block Permalock operating answer.

Block Permalock operation response frame

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Header | | Frame length | | CMDTypes of | data | | BCCcode |
| 0xC8 | 0x8C | 0x00 | 0xxx | 0xA0 | Flag | Errflag | 0xxx |
| End of frame | |  | | | | | |
| 0x0D | 0x0A |

Description: BlockPermalock successful, the error flag Errflag to 0x00. Block Permalock failed, suggesting ErrFlag type of failure, ErrFlag of 1, indicating no label; ErrFlag is 2, represents an access password is wrong; ErrFlag 3, showing Block Permalock operation failed.

in case Block Permalock readlock command parameter is 0, the corresponding data response after ErrFlag, the data word length BlockRange.

example:Block Permalock success,Readlock = 0,BlockRange = 1The data is 0xF000

command:C8 8C 00 0C A0 01 00 F0 00 5D 0D 0A