

**UHF RFID Reader  
Series  
User Manual V2.20**

# Contents

1	Interface specification.....	1
2	Protocol Description.....	1
3	Formats of data frames.....	2
3.1	Command Frame Format (From HOST).....	2
3.2	Response Frame Format (From READER).....	2
4	Command summary.....	4
4.1	EPC C1 G2 (ISO18000-6C) Commands.....	4
4.2	18000-6B Commands.....	4
4.3	Reader customised command.....	5
5	List of the response status.....	6
6	RFID tag error code.....	12
7	Tag memory and access instructions.....	12
8	Command detail description.....	13
8.1	Overview.....	13
8.2	EPC C1G2 commands.....	13
8.2.1	Tags inventory.....	13
8.2.2	Read data.....	18
8.2.3	Write data.....	19
8.2.4	Write EPC number.....	21
8.2.5	Kill tag.....	22
8.2.6	Set read/write protection for specific memory.....	23
8.2.7	Block erase.....	25
8.2.8	Read protection configuration (according to EPC number).....	27
8.2.9	Read protection configuration (Without EPC number).....	29
8.2.10	Unlock read protection.....	29
8.2.11	Read protection status check.....	30
8.2.12	EAS configuration.....	30
8.2.13	EAS alert detection.....	32
8.2.14	Single tag inventory.....	32
8.2.15	Block writing command.....	33
8.2.16	Obtain Monza4QT working parameters.....	35
8.2.17	Modify Monza4QT working parameters.....	36
8.2.18	Extended data reading with assigned mask.....	38
8.2.19	Extended data reading with assigned mask.....	40
8.2.20	Inventory with memory buffer.....	41
8.2.21	Mix inventory.....	44
8.2.22	Inventory with EPC number.....	49
8.2.23	QT inventory.....	50
8.2.24	Select.....	53
8.3	18000-6B Commands.....	55
8.3.1	Single tag inventory.....	55
8.3.2	Inquire tags according specified condition.....	55

8.3.3	Reader data .....	56
8.3.4	Write data.....	57
8.3.5	Obtain lock status .....	58
8.3.6	Lock byte .....	58
8.4	Reader customised commands.....	59
8.4.1	Obtain reader information.....	59
8.4.2	Modify working frequency .....	60
8.4.3	Modify reader address .....	61
8.4.4	Modify reader inventory time .....	62
8.4.5	Modify serial baud rate .....	62
8.4.6	Modify RF power .....	63
8.4.7	Buzzer/LED control.....	64
8.4.8	Setup antenna multiplexing .....	65
8.4.9	Enable/disable buzzer .....	67
8.4.10	GPIO control.....	67
8.4.11	Obtain GPIO state.....	68
8.4.12	Obtain the reader unique serial number .....	68
8.4.13	Modify tag customised function .....	69
8.4.14	Enable antenna check.....	69
8.4.15	Modify communication interface.....	70
8.4.16	Modify or load Antenna return loss threshold configuration .....	70
8.4.17	Modify maximum EPC/TID length configuration for memory buffer .....	71
8.4.18	Load the maximum EPC/TID length configuration.....	72
8.4.19	Obtain data from memory buffer .....	72
8.4.20	Clear memory buffer.....	73
8.4.21	Obtain the total tag amount from memory buffer .....	73
8.4.22	Modify parameters of real time inventory mode .....	74
8.4.23	Modify reader working mode .....	76
8.4.24	Obtain reader working mode .....	78
8.4.25	Load/modify heartbeat packet time break of real time inventory .....	79
8.4.26	Modify RF power configuration separately for write operations.....	80
8.4.27	Load the RF power configuration of write operations .....	80
8.4.28	Modify or load maximum write retry time configuration.....	81
8.4.29	Modify password of tag customised functions .....	82
8.4.30	Obtain password of tag customised functions.....	82
8.4.31	Load/modify reader profile.....	83
8.4.32	Synchronise EM4325 timestamp .....	84
8.4.33	Obtain EM4325 temperature data.....	85
8.4.34	Obtain external data via EM4325 SPI .....	87
8.4.35	Reset EM4325 alert .....	89
8.4.36	Modify or load DRM configuration.....	90
8.4.37	Measure antenna return loss .....	91
8.4.38	Measure current reader temperature .....	92
8.4.39	Set extended parameters .....	92

8.4.40	Set extended parameters .....	93
8.4.41	Start fast inventory.....	94
8.4.42	Stop fast Inventory.....	95
8.4.43	Read Antenna Power.....	95
8.4.44	Stop Inventory .....	95
8.5	Reader customised commands.....	96
8.5.1	Reader configuration parameter table.....	96
8.5.2	Reader parameter function description .....	96

# 1 Interface specification

Communication between host and reader is based on UART or USB. Host delivers commands on these interfaces and reader will produce response and execute the delivered commands.

<b>Data Frame Format</b>	1 start bit, 8 data bits, 1 stop bit
<b>Parity</b>	NONE
<b>Default Baud Rate</b>	57600 bps
<b>Transmission order</b>	Least-significant bit first

# 2 Protocol Description

In a typical communication, host delivers command and parameters to reader, reader should execute the command and provide result or data response. Reader will receive and execute ONE command in every communication cycle. Reader will not be able to receive new command before the last execution and response delivery is completed. Any command which is transmitted during command execution will be ignored by the reader.

Command delivery from host:

Host	Direction	Reader
Command frame	→	

Description:

Once a command is received by reader, the reader will produce a response to host within the predefined inventory time. Data communication time is negligible here.

During communication, the transmitting time gap between every two consecutive bytes should be less than **15ms**. If the time gap exceeds 15ms, all the received data should be discarded and a normal receiving process restarts from the next byte.

Response delivery from reader:

Reader	Direction	Host
Response frame	→	

Description:

During communication, the transmitting time gap between every two consecutive bytes should be less than **15ms**.

A typical communication process is stated below:

1. Host delivers command to reader and wait for response;
2. Reader receives command, starts specific operation according to the received frame and produce response (of operation status and/or data) to host;
3. Host receives response from reader.

## 3 Formats of data frames

### 3.1 Command Frame Format (From HOST)

Len	Adr	Cmd	Data[]	LSB-CRC16	MSB-CRC16
-----	-----	-----	--------	-----------	-----------

#### Format definition:

	Length (in byte)	Description
Len	1	Length of the command frame, defined by the amount of bytes between Adr and MSB_CRC16, i.e. the length of Data[] + 4. The minimum and maximum values of Len are 4 and 255.
Adr	1	Reader device address, the valid range of device address is 0x00 ~ 0xFE. 0xFF is the broadcast address. Reader will only response to command frame stated with a identical address or 0xFF. Default device address: 0x00.
Cmd	1	Command code.
Data[]	Variable	Parameter field, host may or may not provides parameters according to different commands formats.
LSB-CRC16	1	Low byte of CRC16. These 2 bytes are the CRC16 calculation of Len to Data[].
MSB-CRC16	1	High byte of CRC16

### 3.2 Response Frame Format (From READER)

Len	Adr	reCmd	Status	Data[]	LSB-CRC16	MSB-CRC16
-----	-----	-------	--------	--------	-----------	-----------

#### Format description:

	Length (in byte)	Description
Len	1	Length of the response frame, defined by the amount of bytes between Adr and MSB_CRC16), i.e. the length of Data[] + 5.
Adr	1	Reader address.
reCmd	1	reCmd indicates which command this frame is responding to. For unrecognised command, reCmd should be 0x00.
Status	1	Command execution status/result
Data[]	Variable	Data field, reader may or may not provide parameters according to actual commands.
LSB-CRC16	1	Low byte of CRC16.

		These 2 bytes are the CRC16 calculation of Len to Data[.].
MSB-CRC16	1	High byte of CRC16

### C language algorithm of CRC16:

```

#define PRESET_VALUE 0xFFFF
#define POLYNOMIAL 0x8408
unsigned int uiCrc16Cal(unsigned char const * pucY, unsigned char ucX)
{
    unsigned char ucI,ucJ;
    unsigned short int uiCrcValue = PRESET_VALUE;
    for(ucI = 0; ucI < ucX; ucI++)
    {
        uiCrcValue = uiCrcValue ^ *(pucY + ucI);
        for(ucJ = 0; ucJ < 8; ucJ++)
        {
            if(uiCrcValue & 0x0001)
            {
                uiCrcValue = (uiCrcValue >> 1) ^ POLYNOMIAL;
            }
            else
            {
                uiCrcValue = (uiCrcValue >> 1);
            }
        }
    }
    return uiCrcValue;
}

```

pucY is the pointer of the array entry for CRC16 calculation, ucX is the amount of bytes included in this calculation. A 0x0000 return value indicates the input data had passed the CRC16 verification.

Whenever the host receive a data frame from reader, the host should always verify the frame by calling this CRC16 algorithm.

## 4 Command summary

### 4.1 EPC C1 G2 (ISO18000-6C) Commands

No.	Command code	Function
1	0x01	Tag Inventory
2	0x02	Read data command
3	0x03	Write data command
4	0x04	Write EPC number
5	0x05	Kill tag
6	0x06	Set memory read/write protection for specific memory
7	0x07	Block erase
8	0x08	Read protection configuration (according to EPC number)
9	0x09	Read protection configuration (Without EPC number)
10	0x0a	Unlock read protection
11	0x0b	Read protection status check
12	0x0c	EAS configuration
13	0x0d	EAS alert detection
14	0x0f	Single tag inventory
15	0x10	Write blocks
16	0x11	Obtain Monza4QT working parameters
17	0x12	Modify Monza4QT working parameters
18	0x15	Extended data reading with assigned mask
19	0x16	Extended data writing with assigned mask
20	0x18	Inventory with memory buffer
21	0x19	Mix inventory
22	0x1a	Inventory with EPC number
23	0x1b	QT inventory
24	0x9A	Select commad

### 4.2 18000-6B Commands

No.	Command code	Function
1	0x50	Inventory command for SINGLE tag. This command will only inquire one tag in every cycle, with NO inventory condition.
2	0x51	Inventory command for MULTIPLE tags. Inquire tags according

		predefined condition and only return UIDs of the corresponding tags.
3	0x52	Read data command. Read data from tag, maximum 32 bytes in every command cycle.
4	0x53	Write data command. Write data to tag, maximum 32 bytes in every command cycle.
5	0x54	Obtain lock status command. Check the lock status of a particular memory unit.
6	0x55	Byte locking command. Lock a particular (unlocked) byte in a tag.

### 4.3 Reader customised command

No.	Command code	Function
1	0x21	Obtain reader information
2	0x22	Modify working frequency
3	0x24	Modify reader address
4	0x25	Modify reader inventory time
5	0x28	Modify serial baud rate
6	0x2f	Modify RF power
7	0x33	LED/Buzzer control
8	0x3f	Setup antenna multiplexing
9	0x40	Enable/disable buzzer
10	0x46	GPIO control
11	0x47	Obtain GPIO state
12	0x4c	Obtain the reader unique serial number
13	0x3a	Modify tag customised function
14	0x66	Enable antenna check
15	0x6a	Modify communication interface
16	0x6e	Modify or load Antenna return loss threshold configuration
17	0x70	Modify maximum EPC/TID length configuration for memory buffer
18	0x71	Load the maximum EPC/TID length configuration
19	0x72	Obtain data from memory buffer
20	0x73	Clear memory buffer
21	0x74	Obtain the total tag amount from memory buffer
22	0x75	Modify parameters of real time inventory mode
23	0x76	Modify working mode
24	0x77	Load real time inventory mode parameters
25	0x78	Load/modify heartbeat packet time break of real time inventory
26	0x79	Modify RF power configuration separately for write operations
27	0x7a	Load the RF power configuration of write operations

28	0x7b	Modify or load maximum write retry time configuration
29	0x7d	Modify password of tag customised functions
30	0x7e	Obtain password of tag customised functions
31	0x7f	Load/modify reader profile
32	0x85	Synchronise EM4325 timestamp
33	0x86	Obtain EM4325 temperature data
34	0x87	Obtain external data via EM4325 SPI
35	0x88	Reset EM4325 alert
36	0x90	Modify or load DRM configuration
37	0x91	Measure antenna return loss
38	0x92	Measure current reader temperature
39	0x50	Start fast inventory (Ex10 series only)
40	0x51	Stop fast inventory (Ex10 series only)
41	0xEA	Set extended parameters (only support Ex10 series)
42	0xEB	Get extended parameters (only support Ex10 series)
43	0x94	Get RF power by antenna
44	0x93	Stop inventory

## 5 List of the response status

Response frame						Status definition	Description
Len	Adr	reCmd	Status	Data[]	CRC16		
5 + the length of Data[] field	0xXX	0xXX	0x00	...	LSB + MSB	Operation succeed	Command is successfully executed, all the requested data is included in the Data[] field of response frame.
5 + the length of Data[] field	0xXX	0x01	0x01	...	LSB + MSB	Inventory succeed	Host delivered <b>G2</b> tag inventory command, tag inventory was completed successfully and reader is able to deliver data response within the predefined inventory time.

5 + the length of Data[] field	0xXX	0x01	0x02	...	LSB + MSB	Inventory timeout	Host delivered <b>G2</b> tag inventory command, reader fails to complete the inventory within the predefined inventory time.
5 + the length of Data[] field	0xXX	0x01	0x03	...	LSB + MSB	Further data is waiting to be delivered	Host delivered <b>G2</b> tag inventory command, reader is not able to response all the data in a single frame, further data will be transmitted in the following frames.
5 + the length of Data[] field	0xXX	0x01	0x04	...	LSB + MSB	Reader memory is full	Host delivered <b>G2</b> tag inventory command, reader has completed parts of the inventory and run out of memory space due to the amount of tags. Reader will response the inquired EPC number.
5	0xXX	0xXX	0x05	N/A	LSB + MSB	Access password error	Host delivered an incorrect access password to reader.
5	0xXX	0x05	0x09	N/A	LSB + MSB	Tag killing failed	Host delivered a <b>G2</b> tag killing command, but the kill operation is failed due to incorrect tag killing password or poor communication between reader and tag.

5	0xXX	0x05	0x0a	N/A	LSB + MSB	All-zero tag killing password	It is not possible to kill a Tag with all- zero tag killing password.
5	0xXX	0xXX	0x0b	N/A	LSB + MSB	Command is not support by the tag	Some <b>optional</b> commands stated in G2 protocol and some manufacturers' customised command may not be supported by the tags.
5	0xXX	0xXX	0x0c	N/A	LSB + MSB	All-zero access password	For <b>NXP UCODE EPC G2X</b> tag, it is not possible to enable reader protection or enable EAS alert if the tag access password is all-zero.
5	0xXX	0x0a	0x0d	N/A	LSB + MSB	Fail to enable read protection	The read protection is enabled for the target NXP <b>UCODE EPC G2X</b> tag. It is not possible to lock a protection enabled tag.
5	0xXX	0x0a	0x0e	N/A	LSB + MSB	Fail to unlock the tag	The target <b>NXP UCODE EPC G2X</b> tag is already unlocked or the tag locking function is not support by the target tag.
5	0xXX	0x53	0x10	N/A	LSB + MSB	Fail to perform write operation	Some bytes stored in the target <b>6B</b> tag are locked.
5	0xXX	0x55	0x11	N/A	LSB + MSB	Fail to perform lock operation	Fail to perform lock operation on a <b>6B</b> tag.

5	0xXX	0x55	0x12	N/A	LSB + MSB	Fail to perform lock operation	The target <b>6B</b> tag is locked.
5	0xXX	0xXX	0x13	N/A	LSB + MSB	Fail to stored the parameter value	Fail to store the value of some preserved parameters. Configuration will still valid before reader shut down
5	0xXX	0xXX	0x14	N/A	LSB + MSB	Modification failed	Fail to adjust the RF power.
5 + the length of Data[] field	0xXX	0x51	0X15	...	LSB + MSB	response within the predefined inventory time	Host delivered <b>6B</b> tag inventory command, tag inventory was completed successfully and reader is able to deliver data response within the predefined inventory time.
5 + the length of Data[] field	0xXX	0x51	0x16	...	LSB + MSB	Inventory timeout	Host delivered <b>6B</b> tag inventory command, reader fails to complete the inventory within the predefined inventory time.
5 + the length of Data[] field	0xXX	0x51	0x17	...	LSB + MSB	Further data is waiting to be delivered	Host delivered <b>6B</b> tag inventory command, reader is not able to response all the data in a single frame, further data will be transmitted in the following frames.

5 + the length of Data[] field	0xXX	0x51	0x18	...	LSB + MSB	Reader memory is full	Host delivered 6B tag inventory command, reader has completed parts of the inventory and run out of memory space due to the amount of tags. Reader will response the inquired EPC number.
5	0xXX	0xXX	0x19	N/A	LSB + MSB	All-zero access password or function is not supported	Fail to enable EAS alert. It may caused by an all-zero tag access password or the EAS alert is not supported by the target tag.
5	0xXX	0xXX	0x1A	N/A	LSB + MSB	Fail to execute tag customised command	Some special tag functions are enabled, but fail to execute some commands.
5 + the length of Data[] field	0xXX	0x01	0x26	...	LSB + MSB	This frame contains statistic data packet	Host delivered <b>G2</b> tag inventory command, tag inventory was completed successfully and reader is now delivering statistic data packet.
5 + the length of Data[] field	0xXX	0xee	0x28	...	LSB + MSB	This frame contains heartbeat packet	In real time inventory mode, no tag is detected within the heartbeat packet time break, reader will upload heartbeat data.
5	0xXX	0xXX	0xF8	N/A	LSB + MSB	Antenna connection check failure	Antenna connection error detected before tag operation.

5	0xXX	0xXX	0xF9	N/A	LSB + MSB	Fail to execute command	Command execution error.
5	0xXX	0xXX	0xFA	N/A	LSB + MSB	Operation failed	Tags are detected within the effective field, but failed to complete the operation due to poor communication between reader and tags.
5	0xXX	0xXX	0xFB	N/A	LSB + MSB	No operatable tags	No operatable tag is detected in the effective range.
6	0xXX	0xXX	0xFC	Err_code	LSB + MSB	Error code returned from tags	RFID tag reported an error to reader, reader will present this code in the "Err_code" field in the response.
5	0xXX	0xXX	0xFD	N/A	LSB + MSB	Command length error	Host delivered a command frame with incorrect frame length
5	0xXX	0x00	0xFE	N/A	LSB + MSB	Illegal command	Host delivered a incorrect command frame, e.g. unrecognised command code or a command frame with CRC error (failed to pass CRC16 check)
5	0xXX	0xXX	0xFF	N/A	LSB + MSB	Parameter error	Host delivered unrecognised parameters in a command frame.

## 6 RFID tag error code

EPC C1G2 (ISO18000 -6C) RFID tag error code

Error code type	Error code	Error code definition	Error description
Specified error code	0x00	Other errors	All other errors supported by the tag.
	0x03	Memory full, or illegal PC value	Memory location is not found or unsupported PC value.
	0x04	Memory locked	Memory is being temporarily / permanently locked, unable to perform write operation.
	0x0b	Insufficient power supply	Unable to perform write operation due to insufficient power supply to tag.
Non-specified error code	0x0f	Undefined error	Tag unsupported errors.

## 7 Tag memory and access instructions

### A. EPC C1G2 tags (G2 tags)

There are 4 memory areas in a G2 tags, reserved memory (also known as password memory), EPC memory, EID memory and user memory.

**Reserved memory:** 4 bytes, the first 2 bytes are the kill tag password, and the other 2 bytes are the access password. This memory is readable and writable by default. The read/write protection configurations of the two password blocks are isolated, they can be modified separately.

**EPC memory:** the EPC number of tag is stored in this memory. Word0 is CRC16 of PC value and EPC number. Word1 is PC value, indicates the length of EPC number. Actual EPC number stars from Word2. This memory is readable and writable by default.

**TID memory:** the inlay manufacturer predefined tag ID is stored in this memory. It is readable by default but unwritable.

**User memory:** user data is stored in this memory. It is readable and writable by default.

#### NOTE:

- Data length is required in many G2 commands, the unit of data length is either word or byte. The length of 1 word is equal to length of 2 bytes.

- Some commands may require access password, if access password is not setup in the target tag, fill the password field with 0. The password field cannot be empty.

## 8 Command detail description

### 8.1 Overview

There are 2 types of commands:

- protocol related commands;
- reader related commands;

If an incorrect command frame is delivered from host, e.g. unrecognised command code or command frame with CRC error (failed to pass CRC16 check). A similar frame as shown below will be responded from reader.

Len	Adr	reCmd	Status	CRC-16	
0x05	0xXX	0x00	0xFE	LSB	0x05

For incorrect command frame length, the response will be similar to the following frame,

Len	Adr	reCmd	Status	CRC-16	
0x05	0xXX	0xXX	0xFD	LSB	MSB

There are 2 situations which commands will be ignored by reader:

- Incorrect address (neither broadcasting address 0xff, nor corresponding reader address)
- Incomplete command frame, i.e. the actual command frame length is less than the value indicated in the "Len" field

Reader will not provide any response for the situations stated above.

### 8.2 EPC C1G2 commands

#### 8.2.1 Tags inventory

Inventory command is used for inquire RFID tags with corresponding protocol and obtain tag information within the effective field. In order to process further operation on a new tag with unknown EPC, user should firstly obtain tag EPC number by using inventory command.

According to different circumstances, user can modify the setting of maximum operation time (inventory time). Reader will return an inventory result within the predefined inventory time. If the reader fails to complete inventory processes for all the tags in the effective field within the predefined inventory time, the reader will not perform further inventory after inventory time is exceed. Reader will report all the already inquired tags information, as well as the current

(uncompleted) inventory status to host. Reader will then wait for the next command from host.

Default inventory time is 2s, it can be modified by using reader customised command "modify reader inventory time" (refer to chapter 8.4.4). The valid range of inventory time is from 3\*100ms to 255\*100ms (the actual responding time may slightly larger than the setting value and this difference will varies from 0 up to 75ms).

**Note:** an unreasonable short inventory time setting may cause tag inventory failure.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x01	——	LSB	MSB

#### Parameters of Data[]:

Data[]							
QValue	Session	MaskMem	MaskAdr	MaskLen	MaskData	AdrTID	LenTID
0xXX	0xXX	0xXX	2 bytes	0xXX	Variable	0xXX	0xXX

Data[]		
Target	Ant	Scantime
0xXX	0xXX	0xXX

#### Parameters definition

**QValue:** 1 byte.

bit7: Statistic data packet flag.

0 – after inventory, DO NOT deliver statistic data packet of inventory process;

1 – after inventory, deliver statistic data packet of inventory process

bit6: strategy indicator.

0 – general strategy;

1 – special strategy.

bit5: FastID inventory indicator.

0 – disable Impinj FastID function;

1 – enable Impinj FastID function.

Bit4: Phase information.

0 – disable;

1 – enable.

Bit3 ~ bit0: the original Q-value of the EPC tag inventory, the setting of Q-value should follow the rule:  $2^Q \approx$  total amount of tags within the effective field. The range of Q-value is 0 ~ 15, if other value is delivered in this field, reader will return a parameter error status in the response frame.

**Session:** 1 byte, the Session-value of the EPC tag inventory.

0x00 – apply S0 as Session value;

- 0x01 – apply S1 as Session value;
- 0x02 – apply S2 as Session value;
- 0x03 – apply S3 as Session value;
- 0xff – apply reader smart configuration (only valid in EPC inventory).

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskMem:** 1 byte, mask area indication.

- 0x01 – EPC memory;
- 0x02 – TID memory;
- 0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**AdrTID:** entry address of inventory TID memory. If command delivered this parameter and LenTID at the same frame, the mode of inventory should be TID, not EPC.

**LenTID:** data length for TID inventory operation, the valid range of LenTID is 0 ~ 15.

All other values are reserved, and reader will return a parameter error status in the response frame if other value is delivered in this field. If command delivered this parameter and AdrTID, the mode of inventory should be TID, not EPC.

**Target (optional parameter):** 1 byte, the Target value of EPC tag inventory.

- 0x00 – apply A as Target value;
- 0x01 – apply B as Target value.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**Ant (optional parameter):** 1 byte, antenna selection for the current inventory.

- 0x80 – antenna 1;
- 0x81 – antenna 2;
- 0x82 – antenna 3;
- 0x83 – antenna 4;
- ...
- 0x8F – antenna 16.

All other values are reserved, and reader will return a parameter error status in the response frame if other value is delivered in this field.

**ScanTime (optional parameter):** 1 byte, the inventory time configuration for the current inventory. Reader will set the maximum operation time as  $\text{ScanTime} \times 100\text{ms}$ .

**NOTE:**

1. When MaskMem, MaskAdr, MaskLen and MaskData are not provided, no mask pattern will be involved in inventory
2. Target, Ant and ScanTime are optional parameters, all three parameters must be delivered in one command frame. Reader will apply default setting for these parameter, until all three parameters setting are being properly deliver from host, the new settings will then become effective for following inventory.
3. FastID function only valid on FastID supported inlay from Impinj.

(1) Response status is 0x26, the response should be as following:

Len	Adr	reCmd	Status	Data[]			CRC-16	
				Ant	ReadRate	TotalCount		
0xXX	0xXX	0x01	0x26	0xXX	2 bytes	4 bytes	LSB	MSB

Parameters definition

**Status:** the status of response, detail description is shown in the following table

Status	Description
0x26	After inventory, deliver statistic data of the last inventory process

**Ant:** 1/4/8 port reader indicates which antenna had inquired a tag. For example, 0x04 is 0000 0100 in binary, this indicates Antenna 3 had inquired this specific tag. For 0x08, it is 0000 1000 in binary, then it indicates Antenna 4 had inquired tag information.

16 port reader indicates which antenna had inquired a tag. 0~15 indicates ANT1~ANT16.

**ReadRate:** tag identification rate for the current inventory (time of successful identification / sec), a tag being indentified multiple times will also increase the time of successful identification.

**TotalCount:** the total amount of tag detected in the current inventory, a tag being accessed multiple times will also increase the total amount.

(2) Response status is not 0x26, the response should be as following:

Len	Adr	reCmd	Status	Data[]			CRC-16	
				Ant	Num	EPC ID		
0xXX	0xXX	0x01	0xXX	0xXX	0xXX	EPC-1, EPC-2, EPC-3...	LSB	MSB

Parameters definition

**Status:** the status of response, detail description is shown in the following table

Status	Description
--------	-------------

0x01	Operation completed, reader will report all inquired tags information to host
0x02	Inventory timeout, operation is aborted. reader will report all the already inquired tags to host
0x03	For status = 0x03, reader is not able to response all the data in a single frame, further data will be transmitted in the following frames.
0x04	Reader only completed parts of the inventory but run out of memory space due to the amount of tags. Reader will report all the already inquired tags as well as this status value to host.
0xF8	Antenna error detected, the current antenna might be disconnected.

**Ant:** 1/4/8 port reader indicates which antenna had inquired a tag. For example, 0x04 is 0000 0100 in binary, this indicates Antenna 3 had inquired this specific tag. For 0x08, it is 0000 1000 in binary, then it indicates Antenna 4 had inquired tag information.

16 port reader indicates which antenna had inquired a tag. 0~15 indicates ANT1~ANT16.

**Num:** the amount of RFID EPC/TID included in this response.

**EPC ID:** inquired EPC/TID data. For instance, EPC-1 is the EPC/TID length + EPC number/TID data + RSSI value of the first tag etc. The RSSI and EPC length values are one byte parameters. The unit of EPC C1 G2 data is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first.

In the situation when FastID function is enabled, every EPC ID data block follows this format:

Total length of EPC plus TID (1 byte) + tag EPC number + 12 bytes TID data + tag RSSI value

The most-significant bit of total length of EPC and TID indicates whether TID is included in this block.

#### Detail structure of EPC ID

EPC ID					
EPC-1					...
Data length	Data	RSSI	phase	Freq(khz)	
1 byte	N byte	1 byte	4 bytes	3 bytes	
<p><b>bit7:</b> 0 – data block contains EPC or TID; 1 – data block contains EPC plus TID (FastID enabled).</p> <p><b>Bit6:</b> 0 – disable phase and Freq; 1 – enabled phase and Freq.</p> <p><b>bit 5 ~ bit 0:</b> length of data N</p>					

NOTE: FastID function only valid on FastID supported inlay from Impinj.

## 8.2.2 Read data

The function of this command is to read data from reserved memory, EPC memory, TID memory, user memory of tag. Read operation starts from a defined address. The unit of data is word.

### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x02	—	LSB	MSB

### Parameters of Data[]:

Data[]									
ENum	EPC	Mem	WordPtr	Num	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	variable	0xXX	0xXX	0xXX	4 bytes	0xXX	2 bytes	0xXX	variable

### Parameters definition

#### **ENum:**

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMen, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number, and the length value must be an integer. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**Mem:** 1 byte, the type of target memory.

0x00 – reserved memory

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved. Reader will return a parameter error status in the response frame if other value is delivered in this field.

**WordPtr:** 1 byte, the initial word address of target. For instance, 0x00 represent the read operation starts from 1st word (the first 16-bit memory area), 0x01 represents operation starts from the 2nd word, etc.

**Num:** 1 byte, the amount of words will be read in this operation. Hence 0x00 is not a valid value for this parameter. Furthermore the value of Num should not exceed 120, i.e. the maximum reading amount is 120 words. Reader will return a parameter error status in the response frame if a 0 or a larger than 120 value is delivered in this field.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Access password will only be used for verification during reserved memory reading operation, when the associated target memory is password protected and the tag access password is non-zero. Otherwise Psw should be set to all-zero or set to the correct access password.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0xXX	0xXX	0x02	0x00	Word1, Word2,...	LSB	0xXX

Parameters definition

**Word1, Word2...:** the unit of data in this field is word. Every word contains 2 bytes, most-significant byte first. Word1 is the 1st word from the starting address and the 2nd word is Word 2 etc.

### 8.2.3 Write data

The function of this command is to write multiple words to reserved memory, TID memory or user memory in one communication cycle.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x03	—	LSB	MSB

**Parameters of Data[]:**

Data[]					
WNum	ENum	EPC	Mem	WordPtr	Wdt
0xXX	0xXX	variable	0xXX	0xXX	variable
Pwd	MaskMem	MaskAdr	MaskLen	MaskData	
4 bytes	0xXX	2 bytes	0xXX	variable	

**Parameters definition**

**WNum:** the amount of words to be written. Every word contains 2 bytes. The value of WNum must be larger than 0, and less or equal to 32, as well as identical to the actual amount of word to be written. Reader will return a parameter error status in the response frame if a 0 or an incorrect value of WNum is delivered in this field.

**ENum:**

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMen, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**Mem:** 1 byte, the type of target memory.

0x00 – reserved memory

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved. Reader will return a parameter error status in the response frame if other value is delivered in this field.

**WordPtr:** 1 byte, the initial word address of target.

**Wdt:** words to be written to tag memory. The amount of words in Wdt must be identical to the value of WNum. The most-significant byte comes first in a word. Words will be written to tag from lower address to higher address according to the order in Data[]. For instance, WordPtr is 0x02, the 1st word in Data[(from left to right) will be written to the address 0x02 in the memory specified by Mem, the 2nd word will then be written to 0x03 etc.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a

tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Access password must be properly provided and will be verified before write operations. If target memory is not password protected, Pwd can be set to all-zero.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

#### NOTE:

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x03	0x00	—	LSB	MSB

### 8.2.4 Write EPC number

The function of this command is to write EPC number to a tag. During this write EPC operation, only one tag is allowed to be place in the antenna effective area.

#### Command frame:

Len	Adr	Cmd	Data[]			CRC-16	
			ENum	Pwd	WEPC		
0xXX	0xXX	0x04	0xXX	4 bytes	variable	LSB	MSB

#### Parameters definition

**ENum:** 1 byte, the length of EPC number. The unit of ENum is word, and the length of EPC can be 0, but it should be less than 15 words. Reader will return a parameter error status in the response frame if other value is delivered in this field.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a

tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Access password will only be used for verification during reserved memory reading operation, when the associated target memory is password protected and the tag access password is non-zero. Otherwise Psw should be set to all-zero or set to the correct access password.

**WEPC:** the EPC number to be written to tag. The amount of words in WEPC must be identical to the value of ENum. The minimum and maximum length of WEPC is 1 and 31. Reader will return a parameter error status in the response frame if data with inappropriate length is delivered in this field.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x04	0x00	——	LSB	MSB

### 8.2.5 Kill tag

The function of this command is to kill tags. After the kill operation, the tags will no longer process any command from reader.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x05	——	LSB	MSB

**Parameters of Data[]:**

Data[]						
ENum	EPC	Killpwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	variable	4 bytes	0xXX	2 bytes	0xXX	variable

Parameters definition

**ENum:**

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMen, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**Killpwd:** 4 bytes, it is the 4 bytes kill password of a tag. From left to right, the first two bytes are

the high bytes of kill password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit kill password is the most-significant bit in the 1st byte of Killpwd (from left to right). Likewise, the least-significant bit of a tag 32-bit kill password is the least-significant bit in the 4th (last) byte of Killpwd. In order to kill a tag, the kill password of this tag must be non-zero. It is not possible to kill a tag with zero kill password, hence reader will return a parameter error status in the response frame if all-zero are delivered in this field.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x05	0x00	—	LSB	MSB

### 8.2.6 Set read/write protection for specific memory

The function of this command is to set the read/write protection status for the following memory

- Reserved memory  
Readable/ writable without protection, permanently readable/ writable, readable/writeable with password protected or permanently unreadable/ un-writable;
- EPC memory /user reserved memory  
Writable without protection, permanently writable, writeable with password protected or permanently un-writable;

EPC memory, user reserved memory and TID memory are permanently readable. Furthermore TID memory is readable only.

Once the protection status of reserved memory is set to permanently readable/ writable or permanently unreadable/ un-writable, the status configuration will not able to accept any future modification. Likewise, if the protection status of EPC memory, user reserved memory or TID

memory is set to permanently writable/ un-writable, the status configuration will not be able to accept any future modification. Tag will return an error status in the response frame if user attempted to perform future modification.

Access password must be provided to set a specific memory to password protected readable/writable, or change the status from password protected to no protection. Hence, it is essential to ensure access password is already setup for the RFID tag before this operation.

### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x06	——	LSB	MSB

### Parameters of Data[]:

Data[]								
ENum	EPC	Select	SetProtect	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	variable	0xXX	0xXX	4 bytes	0xXX	2 bytes	0xXX	variable

### Parameters definition

#### ENum:

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMem, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**Select:** 1 byte.

- 0x00 – modify kill password read/write protection;
- 0x01 – modify access password read/write protection;
- 0x02 – modify EPC memory read/write protection;
- 0x03 – modify TID memory read/write protection;
- 0x04 – modify user memory read/write protection.

All other values are reserved, reader will not execute the command and will return a parameter error status in the response frame if other value is delivered in this field.

**SetProtect:** the value of SetProtect depends on the value of Select.

- For Select = 0x00 or 0x01, i.e. the kill password or access password protection setting, the value of SetProtect can be the following,

0x00 – set to readable/ writable without protection;  
 0x01 – set to permanently readable/ writable;  
 0x02 – set to readable/writeable with password protected;  
 0x03 – set to permanently unreadable/ un-writable.

- For Select = 0x02, 0x03, 0x04, i.e. the EPC, TID and user memory protection setting, the value of SetProtect can be the following,
  - 0x00 – set to writable without protection;
  - 0x01 – set to permanently writable;
  - 0x02 – set to writeable with password protected;
  - 0x03 – set to permanently un-writable.

All other values of Select and SetProtect are reserved, reader will not execute the command and will return a parameter error status in the response frame if other value is delivered in this field.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. A correct access password must be provided to start the operation.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;  
 0x02 – TID memory;  
 0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x06	0x00	—	LSB	MSB

### 8.2.7 Block erase

The function of this command is to erase multiple words in reserved, EPC, TID or user memory.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x07	——	LSB	MSB

**Parameters of Data[]:**

Data[]									
ENum	EPC	Mem	WordPtr	Num	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	variable	0xXX	0xXX	0xXX	4 bytes	0xXX	2 bytes	0xXX	variable

Parameters definition**ENum:**

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMem, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**Mem:** 1 byte, the type of target memory.

0x00 – reserved memory

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved. Reader will return a parameter error status in the response frame if other value is delivered in this field.

**WordPtr:** 1 byte, the initial word address of target. For instance, 0x01 represents erase operation starts from the 2nd word (the second 16-bit memory area), etc. For EPC memory erase operation, the minimum value of WordPtr should be 0x01. Reader will return a parameter error status in the response frame if inappropriate value is delivered in this field.

**Num:** 1 byte, the amount of words to be erased in this operation. Erase will start from the address stated in WordPtr and will erase amount of words stated in Num. 0x00 is not a valid value for this parameter. Reader will return a parameter error status in the response frame if a 0 or a larger than 120 value is delivered in this field.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a

tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Access password will only be used for verification if the associated memory is password protected. Psw should be set to all-zero or set to the correct access password.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x07	0x00	—	LSB	MSB

### 8.2.8 Read protection configuration (according to EPC number)

The function of this command is to setup read protection for a tag with particular EPC number; hence the tag will be unreadable by device. Device will not able to inquire EPC number of this tag via inventory operation. This command only valid for **NXP UCODE EPC G2X** tags.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x08	—	LSB	MSB

**Parameters of Data[]:**

Data[]						
ENum	EPC	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	variable	4 bytes	0xXX	2 bytes	0xXX	variable

Parameters definition**ENum:**

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMem, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters. Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. It is not possible to setup the read protection status if the tag has all-zero access password, hence reader will return a parameter error status in the response frame if zeros or incorrect access password are delivered in this field.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x08	0x00	—	LSB	MSB

## 8.2.9 Read protection configuration (Without EPC number)

The function of this command is to setup read protection for tags within the effective field. Unlike the previous command in chapter 8.2.8, this command will perform operation on multiple inquired tags without tag identification. In order to perform operation on multiple tags, it is vital to keep access password consistent on those tags. This command only valid for **NXP UCODE EPC G2X** tags.

### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
			Pwd		
0x08	0xXX	0x09	4Byte	LSB	MSB

### Parameter definition

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. It is not possible to setup the read protection status if the tag has all-zero access password, hence reader will return a parameter error status in the response frame if zeros or incorrect access password are delivered in this field.

### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x09	0x00	—	LSB	MSB

## 8.2.10 Unlock read protection

The function of this command is to unlock read protection of a tag. Only one tag is allowed to be placed in the antenna effective area. This command only valid for **NXP UCODE EPC G2X** tags.

### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
			Pwd		
0x08	0xXX	0x0a	4Byte	LSB	MSB

### Parameter definition

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. It is essential to provide a correct access password in this command, reader will return a parameter error status in the response frame if incorrect access password is delivered in this field.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x0a	0x00	—	LSB	MSB

**NOTE:**

For tags with no read protection function, the default status will be unlocked.

**8.2.11 Read protection status check**

This command is NOT ABLE TO identify if a specific tag supports the read protection function, the command ONLY inquire the read protection status of the tag. For tags with no read protection function, the default status will be unlocked. Only one tag is allowed to be place in the antenna effective area. This command only valid for **NXP UCODE EPC G2X** tags.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x0b	—	LSB	MSB

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x06	0xXX	0x0b	0x00	ReadPro	LSB	MSB

Parameter definition

**ReadPro:** 0x00 – read protection is disabled for the tag;

0x01 – read protection is enabled for the tag.

For tags with no read protection function, the default status will be unlocked.

**8.2.12 EAS configuration**

The function of this command is to modify or reset EAS status. This command only valid for **NXP UCODE EPC G2** tags.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x0c	—	LSB	MSB

**Parameters of Data[]:**

Data[]							
ENum	EPC	Pwd	EAS	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	variable	4 bytes	0xXX	0xXX	2 bytes	0xXX	variable

Parameters definition**ENum:**

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMen,

MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. It is not possible to setup EAS if the tag has all-zero access password, hence reader will return a parameter error status in the response frame if zeros or incorrect access password are delivered in this field.

**EAS:** 1 byte.

bit0 = 0, EAS alert is disabled;

bit0 = 1, EAS alert is enabled.

bit1 ~ bit7 are reserved, with 0 as default value.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

#### NOTE

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0xXX	0xXX	0x0c	0x00	—	LSB	MSB

### 8.2.13 EAS alert detection

The function of this command is to detect EAS alert, only valid for **NXP UCODE EPC G2** tags.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x0d	—	LSB	MSB

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x0d	0x00	—	LSB	MSB

Return "no operatable tag" if EAS alert is not detected.

### 8.2.14 Single tag inventory

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x0f	—	LSB	MSB

#### Response frame:

Len	Adr	reCmd	Status	Data[]			CRC-16	
				Ant	Num	EPC ID		
0xXX	0xXX	0x0f	0x01	0xXX	0x01	EPC-1	LSB	MSB

#### Parameters definition

**Ant:** 1/4/8 port reader indicates which antenna had inquired a tag. For example, 0x04 is 0000 0100 in binary, this indicates Antenna 3 had inquired this specific tag. For 0x08, it is 0000 1000 in binary, then it indicates Antenna 4 had inquired tag information.

16 port reader indicates which antenna had inquired a tag. 0~15 indicates ANT1~ANT16.

**Num:** 0x01 (constant)

**EPC ID:** inquired tag EPC data. For instance, EPC-1 is the EPC length + EPC number + **RSSI value** of the first tag etc. The RSSI and EPC length values are one byte parameters. The unit of EPC C1 G2 data is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first.

### 8.2.15 Block writing command

The function of this command is to write multiple words to reserved memory, EPC, TID and user memory.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x10	—	LSB	MSB

#### Parameters of Data[]:

Data[]					
WNum	ENum	EPC	Mem	WordPtr	Wdt
0xXX	0xXX	variable	0xXX	0xXX	variable
Pwd	MaskMem	MaskAdr	MaskLen	MaskData	
4 bytes	0xXX	2 bytes	0xXX	variable	

#### Parameters definition

**WNum:** the amount of words to be written. Every word contains 2 bytes. The value of WNum must be larger than 0, as well as identical to the actual amount of words to be written. Reader will return a parameter error status in the response frame if a 0 or an incorrect value of WNum is delivered in this field.

#### ENum:

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMen, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**Mem:** 1 byte, the type of target memory.

0x00 – reserved memory

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved. Reader will return a parameter error status in the response frame if other value is delivered in this field.

**WordPtr:** 1 byte, the initial word address address of target.

**Wdt:** words to be written to tag memory. The amount of words in Wdt must be identical to the value of WNum. The most-significant byte comes first in a word. Words will be written to tag from lower address to higher address according to the order in Data[]. For instance, WordPtr is 0x02, the 1st word in Data[] (from left to right) will be written to the address 0x02 in the memory specified by Mem, the 2nd word will then be written to 0x03 etc.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Access password must be properly provided and will be verified before the write operation. If target memory is not password protected, Pwd can be set to all-zero.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x10	0x00	—	LSB	MSB

## 8.2.16 Obtain Monza4QT working parameters

This command is used to obtain the current working parameters of a tag, only valid for **Monza 4QT** tags from **Impinj**.

### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x11	—	LSB	MSB

### Parameters of Data[]:

ENum	EPC	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	variable	4 bytes	0xXX	2 bytes	0xXX	variable

### Parameters definition

#### ENum:

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMem, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Access password must be properly provided in the command.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple

of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]		CRC-16	
0x07	0xXX	0x11	0x00	NC	QTcontrol	LSB	MSB
				0x00	1Byte		

Parameter definition

**QTcontrol:** Working parameters of tag.

bit0: The current mirror page setting.

- 0 – private;
- 1 – public.

bit1: distance protection setting.

- 0 – disabled;
- 1 – enabled.

All other values are reserved.

**8.2.17 Modify Monza4QT working parameters**

The function of this command is to modify the current working parameters of a tag, only valid for **Monza 4QT** tags from **Impinj**.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x12	—	LSB	MSB

**Parameters of Data[]:**

ENum	EPC	QTcontrol1	QTcontrol0	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	variable	0x00	0xXX	4 bytes	0xXX	2 bytes	0xXX	variable

Parameter definition**ENum:**

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMem, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The

unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**QTcontrol1:** 0x00 (Constnat), reserved.

**QTcontrol0:** working parameters of tag.

bit0: the current mirror page setting.

0 – private;

1 – public.

bit1: distance protection setting.

0 – disabled;

1 – enabled.

All other values are reserved.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Access password must be properly provided in the command.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x12	0x00	——	LSB	MSB

### 8.2.18 Extended data reading with assigned mask

The function of this command is to read data from reserved memory, EPC memory, TID memory, user memory of tag. Read operation starts from a defined address, and the unit of data is word.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x15	—	LSB	MSB

#### Parameters of Data[]:

Data[]									
ENum	EPC	Mem	WordPtr	Num	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	variable	0xXX	2 bytes	0xXX	4 bytes	0xXX	2 bytes	0xXX	variable

#### Parameters definition

##### ENum:

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMem, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**Mem:** 1 byte, the type of target memory.

0x00 – reserved memory

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved. Reader will return a parameter error status in the response frame if other value is delivered in this field.

**WordPtr:** 2 bytes, the initial word address of target. For instance, 0x0000 represent the read operation starts from 1st word (the first 16-bit memory area), 0x0001 represents operation starts from the 2nd word, etc.

**Num:** 1 byte, the amount of words to be read in this operation, hence 0x00 is not a valid value for this parameter. Furthermore the value of Num should not exceed 120, i.e. the maximum reading amount is 120 words. Reader will return a parameter error status in the response frame if a 0 or a larger than 120 value is delivered in this field.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Access password will only be used for verification during reserved memory reading operation, when the associated target memory is password protected and the tag access password is non-zero. Otherwise Psw should be set to all-zero or set to the correct access password.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0xXX	0xXX	0x15	0x00	Word1, Word2,...	LSB	MSB

Parameter definition

**Word1, Word2...:** the unit of this field is word. Every word contains 2 bytes, most-significant byte first. Word1 is the 1st word from the starting address and the 2nd word is Word 2 etc.

### 8.2.19 Extended data reading with assigned mask

The function of this command is to write multiple words to reserved memory, TID memory or user memory in one communication cycle.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x16	—	LSB	MSB

#### Parameters of Data[]:

Data[]					
WNum	ENum	EPC	Mem	WordPtr	Wdt
0xXX	0xXX	variable	0xXX	2 bytes	variable
Pwd	MaskMem	MaskAdr	MaskLen	MaskData	
4 bytes	0xXX	2 bytes	0xXX	variable	

#### Parameter definition

**WNum:** the amount of words to be written. Every word contains 2 bytes. The value of WNum must be larger than 0 and less or equal to 32, as well as identical to the actual amount of words to be written. Reader will return a parameter error status in the response frame if a 0 or an incorrect value of WNum is delivered in this field.

#### ENum:

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMen, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**WordPtr:** 2 byte, most-significant byte first, the initial word address of target.

**Wdt:** words to be written to tag memory. The amount of words in Wdt must be identical to the value of WNum. The most-significant byte comes first in a word. Words will be written to tag from lower address to higher address according to the order in Data[]. For instance, WordPtr is 0x02, the 1st word in Data[] (from left to right) will be written to the address 0x02 in the memory specified by Mem, the 2nd word will then be written to 0x03 etc.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low byte. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Access password must be properly provided and will be verified before the write operation. If target memory is not password protected, Pwd can be set to all-zero.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

#### NOTE:

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

#### Command frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x16	0x00	—	LSB	MSB

#### 8.2.20 Inventory with memory buffer

Inventory command is used for inquiring/identifying RFID tags, and stored the inquired tag data into reader memory buffer.

The main distinction between Inventory with or without memory buffer (refer to Chapter 8.2.1) is by using the internal memory buffer, reader can continuously inquired tags during the inventory time. At the end of every inventory time, reader will report the total amount of tags stored in the memory buffer (a tag being accessed multiple times will also increase the amount) as well as the

amount of tags inquired in the current inventory. User can access data from the memory buffer via obtain tag data, obtain tag amount or clear memory buffer commands.

Default inventory time is 2s, it can be modified by using reader customised command, "Modify reader inventory time"(refer to Chapter 8.4.4) or perform temporary configuration via optional parameter in the inventory. The valid range of inventory time is from 3\*100ms to 255\*100ms (the actual responding time may slightly larger than the setting value and this difference will varies from 0 up to 75ms).

#### NOTE

- An unreasonable short inventory time setting may causes tag inventory failure.
- Buffer will be cleared if an "inventory without memory buffer" command is delivered from host.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x18	——	LSB	MSB

#### Parameters of Data[]:

Data[]							
QValue	Session	MaskMem	MaskAdr	MaskLen	MaskData	AdrTID	LenTID
0xXX	0xXX	0xXX	2 bytes	0xXX	variable	0xXX	0xXX

Data[]		
Target	Ant	Scantime
0xXX	0xXX	0xXX

#### Parameters definition

**QValue:** 1 byte.

bit7: reserved, set to 0.

bit6: strategy indicator.

0 – general strategy;

1 – special strategy.

bit5 ~ bit0: the original Q-value of the EPC tag inventory, the setting of Q-value should follow the rule:  $2^Q \approx$  total amount of tags within the effective field. The range of Q-value is 0 ~ 15, if other value is delivered in this field, reader will return a parameter error status in the response frame.

**Session:** 1 byte, the Session-value of the EPC tag inventory.

0x00 – apply S0 as Session value;

0x01 – apply S1 as Session value;

0x02 – apply S2 as Session value;

0x03 – apply S3 as Session value;

0xff – apply reader smart configuration (only valid in EPC inventory).

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**AdrTID:** entry address of inventory TID memory. If command delivered this parameter and LenTID at the same frame, the mode of inventory should be TID, not EPC.

**LenTID:** data length for TID inventory operation, the valid range of LenTID is 0 ~ 15.

All other values are reserved, and reader will return a parameter error status in the response frame if other value is delivered in this field. If command delivered this parameter and AdrTID, the mode of inventory should be TID, not EPC.

**Target (optional parameter):** 1 byte, the Target value of EPC tag inventory.

0x00 – apply A as Target value;

0x01 – apply B as Target value.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**Ant (optional parameter):** 1 byte, antenna selection for the current inventory.

0x80 – antenna 1;

0x81 – antenna 2;

0x82 – antenna 3;

0x83 – antenna 4;

...

0x8F – antenna 16.

**ScanTime (optional parameter):** 1 byte, the inventory time configuration for the current inventory. Reader will set the maximum operation time as  $\text{ScanTime} * 100\text{ms}$ .

**NOTE:**

1. When MaskMem, MaskAdr, MaskLen and MaskData are not provided, no mask pattern will be involved in inventory
2. Target, Ant and ScanTime are optional parameters, all three parameters must be delivered in one command frame. Reader will apply default setting for these parameters, until all three parameters setting are being properly deliver from host, the new settings will then become effective for following inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]		CRC-16	
				BufferCount	TagNum		
0x09	0xXX	0x18	0x00	2 bytes	2 bytes	LSB	MSB

Parameter definition

**BufferCount:** 2 bytes, the total amount of tag stored in the memory buffer, tags with identical EPC/TID data will be treated as one tag. BufferCount is the sum of tag amount from multiple inventories, until the memory buffer is being cleared.

**TagNum:** 2 bytes, the amount of tag inquired in the current inventory, a tag being accessed multiple times will also increase the amount.

**8.2.21 Mix inventory**

Inventory command is used for inquire RFID tags with corresponding protocol within the effective field, and obtain specific tag information once the tag EPC number is inquired.

According to different circumstances, user can modify the setting of maximum operation time (inventory time). Reader will return an inventory result within the predefined inventory time. If the reader fails to complete inventory processes for all the tags in the effective field within the predefined inventory time, reader will not perform further inventory after inventory time is exceed. Reader will report all the already inquired tags, as well as the current (uncompleted) inventory status to host. Reader will then wait for the next command from host.

Default inventory time is 2s, it can be modified by using reader customised command "modify reader inventory time" (refer to chapter 8.4.4). The valid range of inventory time is from 3\*100ms to 255\*100ms (the actual responding time may slightly larger than the setting value and this difference will varies from 0 up to 75ms).

**NOTE:**

An unreasonable short inventory time setting may cause tag inventory failure.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x19	——	LSB	MSB

**Parameters of Data[]:**

Data[]					
QValue	Session	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	0xXX	0xXX	2 bytes	0xXX	variable

Data[]						
ReadMem	ReadAdr	ReadLen	Pwd	Target	Ant	Scantime
0xXX	2 bytes	0xXX	4 bytes	0xXX	0xXX	0xXX

Parameters definition**QValue:** 1 byte.

bit7: Statistic data packet flag.

0 – After inventory, DO NOT deliver statistic data packet of inventory process;

1 – After inventory, deliver statistic data packet of inventory process

Bit4: phase data flag.

0 – disable;

1 – enable

bit3 ~ bit0: the original Q-value of the EPC tag inventory, the setting of Q-value should follow the rule:  $2^Q \approx$  total amount of tags within the effective field. The range of Q-value is 0 ~ 15, if other value is delivered in this field, reader will return a parameter error status in the response frame.

**Session:** 1 byte, the Session-value of the EPC tag inventory.

0x00 – apply S0 as Session value;

0x01 – apply S1 as Session value;

0x02 – apply S2 as Session value;

0x03 – apply S3 as Session value;

0xff – apply reader smart configuration (only valid in EPC inventory).

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskMem (optional parameter):** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr (optional parameter):** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen (optional parameter):** 1 byte, bit length of mask (unit: bits).

**MaskData (optional parameter):** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**ReadAdr:** 2 bytes, the entry address of target, most-significant byte first. For instance, 0x0000 represent the read operation starts from 1st word (the first 16-bit memory area), 0x0001 represents operation starts from the 2nd word, etc.

**ReadLen:** 1 byte, the amount of words to be read in this operation, the minimum length is 1 word and the maximum is 120 words. 0x00 is not a valid value for this parameter; furthermore the value of Num should not exceed 120. Reader will return a parameter error status in the response frame if a 0 or a larger than 120 value is delivered in this field.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Access password will only be used for verification during reserved memory reading operation, when the associated target memory is password protected and the tag access password is non-zero. Otherwise Psw should be set to all zero or set to the correct access password.

**Target (optional parameter):** 1 byte, the Target value of EPC tag inventory.

0x00 – apply A as Target value;

0x01 – apply B as Target value.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**Ant (optional parameter):** 1 byte, antenna selection for the current inventory.

0x80 – antenna 1;

0x81 – antenna 2;

0x82 – antenna 3;

0x83 – antenna 4;

...

0x8F – antenna 16.

**ScanTime (optional parameter):** 1 byte, the inventory time configuration for the current inventory. Reader will set the maximum operation time as  $\text{ScanTime} * 100\text{ms}$ .

#### NOTE:

1. When MaskMem, MaskAdr, MaskLen and MaskData are not provided, no mask pattern will be

involved in inventory

2. Target, Ant and ScanTime are optional parameters, all three parameters must be delivered in one command frame. Reader will apply default setting for these parameters, until all three parameters setting are being properly deliver from host, the new settings will then become effective for following inventory.

(1) Response status is 0x26, the response should be as following:

Len	Adr	reCmd	Status	Data[]			CRC-16	
				Ant	ReadRate	TotalCount		
0xXX	0xXX	0x19	0xXX	0xXX	2 bytes	4 bytes	LSB	MSB

#### Parameters definition

**Status:** the status of response, detail description is shown in the following table

Status	Description
0x26	After inventory, deliver statistic data of the current inventory process

**Ant:** 1/4/8 port reader indicates which antenna had inquired a tag. For example, 0x04 is 0000 0100 in binary, this indicates Antenna 3 had inquired this specific tag. For 0x08, it is 0000 1000 in binary, then it indicates Antenna 4 had inquired tag information.

16 port reader indicates which antenna had inquired a tag. 0~15 indicates ANT1~ANT16.

**ReadRate:** tag identification rate for the current inventory (time of successful identification / sec), a tag being identified multiple times will also increase the time of successful identification.

**TotalCount:** the total amount of tag detected in the current inventory, a tag being accessed multiple times will also increase the total amount.

(2) Response status is not 0x26, the response should be as following:

Len	Adr	reCmd	Status	Data[]			CRC-16	
				Ant	Num	Data Packet		
0xXX	0xXX	0x19	0xXX	0xXX	0xXX	Packet-1, Packet-2, Packet-3...	LSB	MSB

#### Parameters definition

**Status:** the status of response, detail description is shown in the following table

Status	Description
0x01	Operation completed, reader will report all inquired tags information to host
0x02	Inventory timeout, operation is aborted. reader will report all the already inquired tags to host
0x03	Reader is not able to response all the data in a single frame, further data will be transmitted in the following frames.
0x04	Reader only completed parts of the inventory but run out of memory space

	due to the amount of tags. Reader will report all the already inquired tags as well as this status value to host.
0xF8	Antenna error detected, the current antenna might be disconnected.

**Ant:** 1/4/8 port reader indicates which antenna had inquired a tag. For example, 0x04 is 0000 0100 in binary, this indicates Antenna 3 had inquired this specific tag. For 0x08, it is 0000 1000 in binary, then it indicates Antenna 4 had inquired tag information.

16 port reader indicates which antenna had inquired a tag. 0~15 indicates ANT1~ANT16.

**Num:** the amount of tag information data packets included in this response.

**Data Packet:** inquired tag data, Packet-1 is data of the first packet, Packet-2 is data of the second packet etc. The format of a typical packet is shown below:

Packet					
PacketParam	Len	Data	RSSI	phase	Freq
0xXX	0xXX	variable	0xXX	4 byts	3 bytes

Parameters data packet:

**PacketParam:** 1 byte, parameters of this packet.

bit7: the type indicator of this data packet.

0 – this data packet contains tag EPC number;

1 – this data packet contains inquired tag data (the EPC number of this tag is delivered in the last data packet).

bit6 ~ bit0: the serial number of the data packet. The range of the serial number is 0 ~ 127. At the beginning of ever mix inventory, the packet serial number will restart from 0, and increase for ever uploaded data packet. Once serial number reaches the maximum value 127, it will recount again from 0.

**Len:** 1 byte, the length of data in this packet. if bit6 =1, enable phase and freq, this data are 7bytes,

**Data:** the EPC number or the inquired data from a specific area of a tag. The unit of **EPC C1 G2** tag data is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first.

**RSSI:** 1 byte, the reader signal strength at the time when the tag is inquired.

**Phase:** 4 byte, if **Len** 's bit6=1, 2 bytes begin phase, 2 bytes end phase;

**Freq:** 3 byte, if **Len** 's bit6=1, unit is khz;

**NOTE:**

Data packet will be uploaded according to EPC inquired order. For every tag, reader will firstly upload the tag EPC in the initial data packet, further tag data will be uploaded in the second data packet. These two packets will have sequential serial number.

In the case when reader fails to read data of a tag after uploaded the EPC packet, reader will not

upload any tag data packet. Reader will continue to inquire the next tag and restart the normal mix inventory process.

### 8.2.22 Inventory with EPC number

The function of this command is to inquire tag with specific EPC. Reader will only response data from the tag with corresponding EPC. All other tags with different EPC will be ignored.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x1a	—	LSB	MSB

#### Parameters of Data[]:

Data[]			
MatchType	MatchLen	MatchOffset	EPCData
0xXX	2 bytes	2 bytes	variable

#### Parameter definition

**MatchType:** EPC number matching type.

- 0 – inquire tag with corresponding EPC;
- 1 – inquire all other tags with different EPC.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MatchLen:** 2 bytes, most-significant byte first, the bit length of the defined EPC number. The range of MatchLen is 1 ~ 196 bits.

**MatchOffset:** 2 bytes, most-significant byte first. The entry address of the defined EPC (unit: bits). The range of MatchOffset is 0 ~ 495 bits.

#### NOTE:

The sum of **MatchLen** and **MatchOffset** should not exceed 496 bits. Reader will return a parameter error status in the response frame if incorrect value is delivered in this field.

**EPCData:** the defined EPC number data. The length of EPCData equals to MatchLen/8. If MatchLen is not a multiple of 8 integer, the length of EPCData is equal to the  $\text{int}[\text{MatchLen} / 8] + 1$ . Non-specified lower significant figures should be filled up with 0.

#### Response frame:

Len	Adr	reCmd	Status	Data[]			CRC-16	
				Ant	Num	EPC ID		
0xXX	0xXX	0x1a	0xXX	0xXX	0xXX	EPC-1	LSB	MSB

### Parameters definition

**Status:** the status of response, detail description is shown in the following table

Status	Description
0x01	Operation completed, reader will report all inquired tags information to host
0x02	Inventory timeout, operation is aborted. reader will report all the already inquired tags to host
0x03	Reader is not able to response all the data in a single frame, further data will be transmitted in the following frames.
0xF8	Antenna error detected, the current antenna might be disconnected.

**Ant:** 1/4/8 port reader indicates which antenna had inquired a tag. For example, 0x04 is 0000 0100 in binary, this indicates Antenna 3 had inquired this specific tag. For 0x08, it is 0000 1000 in binary, then it indicates Antenna 4 had inquired tag information.

16 port reader indicates which antenna had inquired a tag. 0~15 indicates ANT1~ANT16.

**Num:** the amount of tag EPC included in this response.

**EPC ID:** inquired EPC/TID data. For instance, EPC-1 is the EPC/TID length + EPC number/TID data + RSSI value of the first tag etc. The RSSI and EPC length values are one byte parameters. The unit of EPC C1 G2 data is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first.

### 8.2.23 QT inventory

This command is used to inquiry the private EPC number of a Monza4QT tag with public mirroring. The command seeks RFID tags with corresponding protocol. In order to process further operation on a new tag with unknown EPC, user should firstly obtain tag EPC number by using inventory command.

According to different circumstances, user can modify the setting of maximum operation time (inventory time). Reader will return an inventory result within the predefined inventory time. If the reader fails to complete inventory processes for all the tags in the effective field within the predefined inventory time, reader will not perform further inventory after inventory time is exceed. Reader will report all the already inquired tags, as well as the current (uncompleted) inventory status to host. Reader will then wait for the next command from host.

Default inventory time is 1s, it can be modified by using reader customised command "modify reader inventory time" (refer to chapter 8.4.4). The valid range of inventory time is from 3\*100ms to 255\*100ms (the actual responding time may slightly larger than the setting value and this

difference will varies from 0 up to 75ms).

**Note:** an unreasonable short inventory time setting may cause tag inventory failure.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0x0XX	0xXX	0x1b	—	LSB	MSB

**Parameters of Data[]:**

Data[]				
QValue	Session	Target	Ant	Scantime
0xXX	0xXX	0xXX	0xXX	0xXX

Parameters definition

**QValue:** 1 byte.

bit7: Statistic data packet flag.

0 – After inventory, DO NOT deliver statistic data packet of inventory process;

1 – After inventory, deliver statistic data packet of inventory process

bit6 ~ bit0: the original Q-value of the EPC tag inventory, the setting of Q-value should follow the rule:  $2^Q \approx$  total amount of tags within the effective field. The range of Q-value is 0 ~ 15, if other value is delivered in this field, reader will return a parameter error status in the response frame.

**Session:** 1 byte, the Session-value of the EPC tag inventory.

0x00 – apply S0 as Session value;

0x01 – apply S1 as Session value;

0x02 – apply S2 as Session value;

0x03 – apply S3 as Session value;

0xff – apply reader smart configuration (only valid in EPC inventory).

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**Target (optional parameter):** 1 byte, the Target value of EPC tag inventory.

0x00 – apply A as Target value;

0x01 – apply B as Target value.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**Ant (optional parameter):** 1 byte, antenna selection for the current inventory.

0x80 – antenna 1;

0x81 – antenna 2;

0x82 – antenna 3;

0x83 – antenna 4;  
 ...  
 0x8F – antenna 16.

All other values are reserved, and reader will return a parameter error status in the response frame if other value is delivered in this field.

**ScanTime (optional parameter):** 1 byte, the inventory time configuration for the current inventory. Reader will set the maximum operation time as ScanTime\*100ms.

**NOTE:**

1. Target, Ant and ScanTime are optional parameters, all three parameters must be delivered in one command frame. Reader will apply default setting for these parameter, until all three parameters setting are being properly deliver from host, the new settings will then become effective for following inventory.

(1) Response status is 0x26, the response should be as following:

Len	Adr	reCmd	Status	Data[]			CRC-16	
				Ant	ReadRate	TotalCount		
0xXX	0xXX	0x1b	0xXX	0xXX	2 bytes	4 bytes	LSB	MSB

Parameters definition

**Status:** the status of response, detail description is shown in the following table,

Status	Description
0x26	After inventory, deliver statistic data of the last inventory process

**Ant:** 1/4/8 port reader indicates which antenna had inquired a tag. For example, 0x04 is 0000 0100 in binary, this indicates Antenna 3 had inquired this specific tag. For 0x08, it is 0000 1000 in binary, then it indicates Antenna 4 had inquired tag information.

16 port reader indicates which antenna had inquired a tag. 0~15 indicates ANT1~ANT16.

**ReadRate:** tag identification rate for the current inventory (time of successful identification / sec), a tag being identified multiple times will also increase the time of successful identification.

**TotalCount:** the total amount of tag detected in the current inventory, a tag being accessed multiple times will also increase the total amount.

(2) Response status is not 0x26, the response should be as following:

Len	Adr	reCmd	Status	Data[]			CRC-16	
				Ant	Num	EPC ID		
0xXX	0xXX	0x1b	0xXX	0xXX	0xXX	EPC-1, EPC-2, EPC-3...	LSB	MSB

Parameters definition

**Status:** the status of response, detail description is shown in the following table,

Status	Description
0x01	Operation completed, reader will report all inquired tags information to host
0x02	Inventory timeout, operation is aborted. reader will report all the already inquired tags to host
0x03	Reader is not able to response all the data in a single frame, further data will be transmitted in the following frames.
0x04	Reader only completed parts of the inventory but run out of memory space due to the amount of tags. Reader will report all the already inquired tags as well as this status value to host.

**Ant:** 1/4/8 port reader indicates which antenna had inquired a tag. For example, 0x04 is 0000 0100 in binary, this indicates Antenna 3 had inquired this specific tag. For 0x08, it is 0000 1000 in binary, then it indicates Antenna 4 had inquired tag information.

16 port reader indicates which antenna had inquired a tag. 0~15 indicates ANT1~ANT16.

**Num:** the amount of EPC/TID included in this response.

**EPC ID:** inquired EPC/TID data. For instance, EPC-1 is the EPC/TID length + EPC number/TID data + **RSSI value** of the first tag etc. The RSSI and EPC length values are one byte parameters. The unit of EPC C1 G2 data is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first.

### 8.2.24 Select

This command is used to send the Select command in the EPC C1G2 protocol separately from multiple antennas. This command can send the Select command from multiple specified ports on the same time.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x9A	——	LSB	MSB

**Parameters of Data[]:**

Ant	SelTarget	SelAction	MaskMem	MaskAdr	MaskLen	MaskData	Truncate
0xXX/ 0XXXXX	0xXX	0x00	0xXX	2 bytes	0xXX	variable	0xXX

Parameter definition

**Ant:** 1/4/8 port reader, 1 byte ,represents one or more antenna numbers to send the selection command this time, and each Bit represents an antenna number. For example, 0x04, the binary number is 0000 0100, which means it needs to be sent on antenna 3; For example, 0x0A, the binary number is 000010, which means it needs to be sent on antenna 2 and 4.

16-port reader, 2 bytes, represents one or more antenna numbers for this selection command, and each Bit represents an antenna number. If 0x0004, the binary number is 0000 0000 0000 0100, it means that it needs to be sent on antenna 3; For example, 0x000A, the binary number is 0000 0000

0000 1010, which means it needs to be sent on antenna 2 and 4..

**SelTarget:** 1 byte, select the Target parameter in the command.

0x00:Target uses S0;

0x01:Target uses S1;

0x02:Target uses S2;

0x03: Target uses S3.

0x04: Target uses SL.

0x05~0x07:RFU reserved value, currently not available.

Other values are retained. If other values appear in the command, a message with an error in the parameter is returned..

**SelAction:** 1 byte, select the Action parameter in the command. The parameters range from 0 to 7, corresponding functions are as follows:

Action	Matching	Non-Matching
0x00	assert SL or inventoried → A	deassert SL or inventoried → B
0x01	assert SL or inventoried → A	do nothing
0x02	do nothing	deassert SL or inventoried → B
0x03	negate SL or (A → B, B → A)	do nothing
0x04	deassert SL or inventoried → B	assert SL or inventoried → A
0x05	deassert SL or inventoried → B	do nothing
0x06	do nothing	assert SL or inventoried → A
0x07	do nothing	negate SL or (A → B, B → A)

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**Truncate:** 1 byte, 0: Disable truncation1: Enable truncation.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16
-----	-----	-------	--------	--------	--------

0x05	0xXX	0x9A	0x00	—	LSB	MSB
------	------	------	------	---	-----	-----

## 8.3 18000-6B Commands

### 8.3.1 Single tag inventory

The function of this command is to inquire a single tag. If multiple tags are placed within the effective field, reader may fail to inquire any tag.

#### Command frame:

Len	Adr	Cmd	CRC-16	
0x04	0xXX	0x50	LSB	MSB

#### Response frame:

Len	Adr	reCmd	Status	Data[]		CRC-16	
				Ant	ID		
0x10	0xXX	0x50	0x00	0Xxx	10 bytes	LSB	MSB

#### Parameters definition

**Ant:** indicates which antenna had inquired a tag. For example, 0x04 is 0000 0100 in binary, this indicates Antenna 3 had inquired this specific tag. For 0x08, it is 0000 1000 in binary, then it indicates Antenna 4 had inquired tag information.

**ID:** 10 bytes, the 1st byte is the length of UID, 0x08. The 2nd ~ 9th byte is the tag UID number, least-significant byte first. The 10th byte is RSSI.

### 8.3.2 Inquire tags according specified condition

The function of this command is to inquire tags according to specified condition.

#### Command frame:

Len	Adr	Cmd	Data[]				CRC-16	
			Condition	Address	Mask	Word_data		
0x0f	0xXX	0x51	0xXX	0xXX	0xXX	8 bytes	LSB	MSB

#### Parameters definition

**Condition:** the condition for inventory;

- 0x00 – equal to the reference;
- 0x01 – no equal to the reference;
- 0x02 – larger than the reference
- 0x03 – smaller than the reference

**Address:** address of the comparing data

**Mask:** mask, specify the data for comparison.

**Word\_data:** the value of reference. Every bit in mask is corresponds to a byte in Word\_data. The most-significant bit (bit7) corresponds to the 1st byte in Word\_data (from left to right). Consequently, the least- significant bit (bit0) corresponds to the last byte in Word\_data (from left to right). Word\_data is the reference data for comparison with data in tag starts from the location stated in Address. The 1st byte of Word\_data will be compare with the data store in Address, the last byte of Word\_data will be compare with the data store in Address+7. Set the byte corresponding bit of Mask to 1, reader will perform comparison on that particular byte. Set the bit to 0, no comparison will be performed on that particular byte.

**Response frame:**

Len	Adr	reCmd	Status	Ant	Num	Data[]	CRC-16	
0xXX	0xXX	0x51	0xXX	0xXX	0xXX	UID1, UID2...	LSB	MSB

Parameters definition

**Status:** the status of response, detail description is shown in the following table

Status	Description
0x15	Operation completed, reader will report all inquired tags information to host.
0x16	Inventory timeout, operation is aborted. Reader will report all the already inquired tags to host.
0x17	Reader is not able to response all the data in a single frame, further data will be transmitted in the following frames.
0x18	Reader only completed parts of the inventory but run out of memory space due to the amount of tags. Reader will report all the already inquired tags as well as this status value to host..
0xF8	Antenna error detected, the current antenna might be disconnected.

**Ant:** indicates which antenna had inquired a tag. For example, 0x04 is 0000 0100 in binary, this indicates Antenna 3 had inquired this specific tag. For 0x08, it is 0000 1000 in binary, then it indicates Antenna 4 had inquired tag information.

**Num:** the amount of UID included in this response.

**Data[]:** UID of the tag. The length of every UID data block is 10 bytes. The 1st byte is the length value of UID number, i.e. 0x08. The 2nd ~ 9th bytes are the tag UID number, least-significant byte first. The 10th byte is the RSSI value. For Num = 0, i.e. no UID data is provided, Data[] is empty.

### 8.3.3 Reader data

The function of this command is to read multiple bytes from a specific address.

**Command frame:**

Len	Adr	Cmd	Data[]			CRC-16	
			Address	ID	Num		
0x0e	0xXX	0x52	0xXX	8 bytes	0xXX	LSB	MSB

#### Parameters definition

**Address:** the entry address of target. The valid value range of Address is 0 ~ 223. Reader will return a parameter error status in the response frame if incorrect address is delivered.

**Num:** the amount of words will be read in this operation. The valid value range of Num is 1 ~ 32. If the sum of Address and Num is larger than 224, Num is equal to 0 or larger than 32, return a parameter error status in the response frame.

**ID:** the ID of the target tag, 8 bytes, least significant byte first.

#### **Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0xXX	0xXX	0x52	0x00	Data	LSB	MSB

#### Parameter definition

Data: inquired data from tag, least-significant byte first.

### 8.3.4 Write data

The function of this command is to write multiple bytes to a specific tag.

#### **Command frame:**

Len	Adr	Cmd	Data[]			CRC-16	
			Address	ID	Wdata		
0xXX	0xXX	0x53	0xXX	8 bytes	variable	LSB	MSB

#### Parameter definition

**Address:** 1 byte, the address of the target byte. The range of address is 8 ~ 223. Reader will return a parameter error status in the response frame if other value is delivered in this field.

**ID:** 8 bytes, the ID number of the target tag, the least-significant byte first.

**Wdata:** the data to be written to the tag. The length of Wdata should not exceed 32 bytes. The most-significant byte will be written to the lowest byte from entry stated in Address. If the length of Wdata is 0 (no data) or exceed 32, reader will return a parameter error status in the response frame. Besides, if the sum of Address and Wdata exceeds 224, reader will also return parameter error.

#### **Command frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x53	0x00	——	LSB	MSB

### 8.3.5 Obtain lock status

The function of this command is to obtain the lock status of a specific byte.

#### Command frame:

Len	Adr	Cmd	Data[]		CRC-16	
			Address	ID	LSB	MSB
0x0d	0xXX	0x54	0xXX	8 bytes	LSB	MSB

#### Parameters definition

**Address:** 1 byte, the address of the target byte. The range of address is 0 ~ 223. Reader will return a parameter error status in the response frame if other value is delivered in this field.

**ID:** 8 bytes, the ID number of the target tag. The least-significant byte should be transmitted first.

#### Command frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x06	0xXX	0x54	0x00	LockState	LSB	MSB

#### Parameter definition

LockState: 0x00: the byte is unlocked;  
0x01: the byte is locked.

### 8.3.6 Lock byte

The function of this command is to lock a specific byte in a tag.

#### Command frame:

Len	Adr	Cmd	Data[]		CRC-16	
			Address	ID	0x0d	0xXX
0x0d	0xXX	0x55	0xXX	8 bytes	0x0d	0xXX

#### Parameters definition

**Address:** 1 bytes, the address of the target byte. The range of address is 8 ~ 223. Reader will return a parameter error status in the response frame if other value is delivered in this field.

**ID:** 8 bytes, the ID number of the target tag. The least-significant byte should be transmitted first.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x55	0x00	——	LSB	MSB

## 8.4 Reader customised commands

### 8.4.1 Obtain reader information

The function of this command is to obtain reader information, including reader address, firmware version, reader model code, supported protocols, working frequency band, RF power, inventory time, buzzer status, GPIO for notification etc.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x21	—	LSB	MSB

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x0f	0xXX	0x21	0x00	Version, Type, Tr_Type, dmaxfre, dminfre, Power, Scntm, Ant, Reserved, Reserved, CheckAnt	LSB	MSB

#### Parameters of Data[]:

Parameters	Length (bytes)	Description
Version	2	Firmware version, the high byte states the main version number and the low byte states the subversion number.
Type	1	Reader model code.
Tr_Type	1	Supported prototypes. bit1 = 1: supports 18000-6C. bit0 = 1: supports 18000-6B. All other bits are reserved.
dmaxfre	1	bit7 ~ bit6: frequency band configuration; bit5 ~ bit0: maximum frequency point.
dminfre	1	bit7 ~ bit6: frequency band configuration; bit5 ~ bit0: minimum frequency point.
Power	1	Output RF power, range from 0 to 30.
Scntm	1	Inventory time. Reader will respond the inventory command delivered from host within this specific inventory time.
Ant	1	Antenna configuration.
Reserved	1	Reserved.
Reserved	1	Reserved.
CheckAnt	1	Antenna check configuration 0: antenna check off; 1: antenna check on.

#### Frequency band configuration table

MaxFre(bit7)	MaxFre(bit6)	MinFre(bit7)	MinFre(bit6)	FreqBand
--------------	--------------	--------------	--------------	----------

0	0	0	0	Reserved
0	0	0	1	Chinese band2
0	0	1	0	US band
0	0	1	1	Korean band
0	1	0	0	EU band
0	1	0	1	Reserved
0	1	1	0	Ukraine band
0	1	1	1	Peru band
1	0	0	0	Chinese band1
1	0	0	1	EU3 band
1	0	1	0	Taiwan band
1	1	0	0	US band3
...	...	...	...	...
1	1	1	1	Reserved

#### 8.4.2 Modify working frequency

The function of this command is to select frequency band, modify maximum and minimum frequency points. Maximum frequency point must be larger or equal the minimum frequency point.

##### Command frame:

Len	Adr	Cmd	Data[]		CRC-16	
			MaxFre	MinFre	LSB	MSB
0x06	0xXX	0x22	0xXX	0xXX	LSB	MSB

##### Parameters definition

**MaxFre:** 1 byte.

bit7 ~ bit6: frequency band configuration;

bit5 ~ bit0: maximum frequency point.

**MinFre:** 1 byte.

bit7 ~ bit6: frequency band configuration;

bit5 ~ bit0: minimum frequency point.

##### NOTE:

Maximum frequency point must be larger or equal the minimum frequency point. Reader will return a parameter error status in the response frame if other value is delivered in this field.

##### Frequency band configuration table

MaxFre(bit7)	MaxFre(bit6)	MinFre(bit7)	MinFre(bit6)	FreqBand
0	0	0	0	Reserved
0	0	0	1	Chinese band2
0	0	1	0	US band

0	0	1	1	Korean band
0	1	0	0	EU band
0	1	0	1	Reserved
0	1	1	0	Ukraine band
0	1	1	1	Peru band
1	0	0	0	Chinese band1
1	0	0	1	EU3 band
1	0	1	0	Taiwan band
1	1	0	0	US band3
...	...	...	...	...
1	1	1	1	Reserved

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x22	0x00	—	LSB	MSB

**Formulae for different frequency bands:**

Chinese band2:  $F_s = 920.125 + N * 0.25$  (MHz) where  $N \in [0, 19]$ .

US band:  $F_s = 902.75 + N * 0.5$  (MHz) where  $N \in [0, 49]$ .

Korean band:  $F_s = 917.1 + N * 0.2$  (MHz) where  $N \in [0, 31]$ .

EU band:  $F_s = 865.1 + N * 0.2$  (MHz) where  $N \in [0, 14]$ .

Ukraine band:  $F_s = 868.0 + N * 0.1$  (MHz) where  $N \in [0, 6]$ .

Peru band:  $F_s = 916.2 + N * 0.9$  (MHz) where  $N \in [0, 11]$ .

Chinese band1:  $F_s = 840.125 + N * 0.25$  (MHz) where  $N \in [0, 19]$ .

EU3 band:  $F_s = 865.7 + N * 0.6$  (MHz) where  $N \in [0, 3]$ .

US band3:  $F_s = 902 + N * 0.5$  (MHz) where  $N \in [0, 52]$ .

Taiwan band:  $F_s = 922.25 + N * 0.5$  (MHz) where  $N \in [0, 11]$ .

**8.4.3 Modify reader address****Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
			address		
0x05	0xXX	0x24	0xXX	LSB	MSB

Parameter definition

**Adr:** new address setting of reader. 0xFF is not a valid address, reader will return a parameter error status in the response frame if 0xFF is delivered in this field.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x24	0x00	—	LSB	MSB

**NOTE:**

Reader will response this command with the ORIGINAL address.

**8.4.4 Modify reader inventory time****Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0x05	0xXX	0x25	Scantime	LSB	MSB
			0xXX		

Parameter definition

**Scantime:** inventory time. Reader will modify the maximum response time according to user defined value (0\*100ms ~ 255\*100ms), and reader will apply this new setting for future inventories. Default setting of Scantime is 0x14 (corresponding to 20\*100ms). Valid setting of Scantime is 0x00 ~ 0xff (corresponding to 3\*100ms ~ 255\*100ms).

**NOTE:** the actual responding time may slightly larger than the setting value and this difference will varies from 0 up to 75ms. For Scantime = 0, operation time is not limited until inventories on all the tags are completed.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x25	0x00	—	LSB	MSB

**8.4.5 Modify serial baud rate**

The function of this command is to modify baud rate for the serial port.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0x05	0xXX	0x28	baudrate	LSB	MSB
			0xXX		

Parameters definition

**baudrate:** the new baud rate setting, options of baudrate is 0/1/2/5/6 (refer to the following table for proper baud rate), all other values are reserved.

**baudrate configuration table**

baudrate value	Actual baud rate
0	9600bps
1	19200 bps
2	38400 bps
5	57600 bps
6	115200 bps

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x28	0x00	—	LSB	MSB

**NOTE:**

Response will be delivered using the ORIGINAL baud rate setting. The new baud rate setting will take effect in the NEXT COMMUNICATION.

**8.4.6 Modify RF power**

The function of this command is to modify the RF power of reader. The valid value of this parameter is 0 ~ 30. For setting of 30, the output power is approximately 1W.

**Format 1:**

Len	Adr	Cmd	Data[]	CRC-16	
0x05	0xXX	0x2F	Pwr	LSB	MSB
			0xXX		

**Format 2:**

Len	Adr	Cmd	Data[]	CRC-16	
0x08	0xXX	0x2F	Pwr	LSB	MSB
			4bytes		

**Format 3:**

Len	Adr	Cmd	Data[]	CRC-16	
0x0C	0xXX	0x2F	Pwr	LSB	MSB
			8bytes		

**Format 4:**

Len	Adr	Cmd	Data[]	CRC-16	
0x14	0xXX	0x2F	Pwr	LSB	MSB
			16bytes		

Parameter definition

**Format1:Pwr:** 1 byte, the RF power parameters.

bit0 ~ bit6: RF power setting, the valid value of this parameter is 0 ~ 30. For setting of 30, the output power is approximately 1W.

bit7: configuration preservation status during power off.

0 – configuration preserved during reader power off;

1 – configuration is not preserved.

**Format2:Pwr:** 4 byte, From left to right, there are antennas 1 to 4, each represented as follows the RF power parameters.

bit0 ~ bit6: RF power setting, the valid value of this parameter is 0 ~ 30. For setting of 30, the output power is approximately 1W.

bit7: configuration preservation status during power off.

0 – configuration preserved during reader power off;

1 – configuration is not preserved

**Format3:Pwr:** 8 byte, From left to right, there are antennas 1 to 8, each represented as follows the RF power parameters.

bit0 ~ bit6: RF power setting, the valid value of this parameter is 0 ~ 30. For setting of 30, the output power is approximately 1W.

bit7: configuration preservation status during power off.

0 – configuration preserved during reader power off;

1 – configuration is not preserved

**Format4:Pwr:** 16 byte, From left to right, there are antennas 1 to 16, each represented as follows the RF power parameters.

bit0 ~ bit6: RF power setting, the valid value of this parameter is 0 ~ 30. For setting of 30, the output power is approximately 1W.

bit7: configuration preservation status during power off.

0 – configuration preserved during reader power off;

1 – configuration is not preserved

#### Command frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x2F	0x00	—	LSB	MSB

#### 8.4.7 Buzzer/LED control

The function of this command is to control the TAG LED and buzzer (share with GPO1 pin) to perform specific actions.

#### Command frame:

Len	Adr	Cmd	Data[]			CRC-16	
			ActiveT	SilentT	Times		
0x07	0xXX	0x33	0xXX	0xXX	0xXX	LSB	MSB

#### Parameters definition

**ActiveT:** TAG LED and buzzer (GPO1 pin low TTL level) switch ON time is ActiveT\*50ms. The range of ActiveT is 0 ~ 255. Default setting is 0.

**SilentT:** TAG LED and buzzer (GPO1 pin low TTL level) switch OFF time is SilentT \*50ms. The range of SilentT is 0 ~ 255. Default setting is 0.

**Times:** TAG LED and buzzer action time, the range of Times is 0 ~ 255. Default setting is 0.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16
-----	-----	-------	--------	--------	--------

0x05	0xXX	0x33	0x00	—	LSB	MSB
------	------	------	------	---	-----	-----

### 8.4.8 Setup antenna multiplexing

The function of this command is to modify reader antenna configuration.

#### Format 1:

Len	Adr	Cmd	Data[]	CRC-16	
			Ant		
0x05	0xXX	0x3f	0xXX	LSB	MSB

#### Format 2:

Len	Adr	Cmd	Data[]			CRC-16	
			SetOnce	AntCfg1	AntCfg2		
0x07	0xXX	0x3f	0xXX	0xXX	0xXX	LSB	MSB

#### Parameter definition

##### Format1:

**Ant:** antenna configuration information. 1 antenna is the minimum requirement for operation.

**bit0:** antenna 1 configuration bit.

0 – disable antenna 1;

1 – enable antenna 1.

**bit1:** antenna 2 configuration bit.

0 – disable antenna 2;

1 – enable antenna 2.

**bit2:** antenna 3 configuration bit.

0 – disable antenna 3;

1 – enable antenna 3.

**bit3:** antenna 4 configuration bit.

0 – disable antenna 4;

1 – enable antenna 4.

**bit4 ~ bit6:** reserved, default value is 0.

**bit7:** configuration preservation status.

0 – configuration preserved during reader power off;

1 – configuration not preserved.

##### Format2:

**SetOnce:** 1 byte, save or not when power off. Bit0=0, save; Bit0=1, not save.

**AntCfg1:** 1 byte, 8 port reader is Reserve. 16 port reader as follows:

**bit0:** antenna 9 configuration bit.

0 – disable antenna 9;

1 – enable antenna 9.

bit1: antenna 10 configuration bit.

0 – disable antenna 10;

1 – enable antenna 10.

bit2: antenna 11 configuration bit.

0 – disable antenna 11;

1 – enable antenna 11.

bit3: antenna 12 configuration bit.

0 – disable antenna 12;

1 – enable antenna 12.

bit4: antenna 13 configuration bit.

0 – disable antenna 13;

1 – enable antenna 13.

bit5: antenna 14 configuration bit.

0 – disable antenna 14;

1 – enable antenna 14.

bit6: antenna 15 configuration bit.

0 – disable antenna 15;

1 – enable antenna 15.

bit7: antenna 16 configuration bit.

0 – disable antenna 16;

1 – enable antenna 16.

**AntCfg1:** 1 byte, antenna configuration information. 1 antenna is the minimum requirement for operation.

bit0: antenna 1 configuration bit.

0 – disable antenna 1;

1 – enable antenna 1.

bit1: antenna 2 configuration bit.

0 – disable antenna 2;

1 – enable antenna 2.

bit2: antenna 3 configuration bit.

0 – disable antenna 3;

1 – enable antenna 3.

bit3: antenna 4 configuration bit.

0 – disable antenna 4;

1 – enable antenna 4.

bit4: antenna 5 configuration bit.

0 – disable antenna 5;

1 – enable antenna 5.

bit5: antenna 6 configuration bit.

0 – disable antenna 6;

1 – enable antenna 6.

bit6: antenna 7 configuration bit.

0 – disable antenna 7;

1 – enable antenna 7.

bit7: antenna 8 configuration bit.

0 – disable antenna 8;

1 – enable antenna 8.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x3f	0x00	—	LSB	MSB

#### 8.4.9 Enable/disable buzzer

The function of this command is enable/disable buzzer (share with GPO1 pin).

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
			BeepEn		
0x05	0xXX	0x40	0Xxx	LSB	MSB

#### Parameter definition

**BeepEn:** buzzer working mode selection.

bit0: 0 – disable buzzer;

1 – enable buzzer, the buzzer will beep on every success tag operation(GPO1 pin output low pulse).

bit1 ~ bit7 are reserved, default value is 0.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x40	0x00	—	LSB	MSB

#### 8.4.10 GPIO control

The function of this command is to control the output of GPIO. The default output is high TTL level.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
			OutputPin		
0x05	0xXX	0x46	0xXX	LSB	MSB

#### Parameter definition

**OutputPin:** the output state of GPIO (pin Out1 to Out2). Bit0 ~ bit3 control the Out1 to Out3

respectively. Bit2 ~ bit7 are reserved.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x46	0x00	——	LSB	MSB

### 8.4.11 Obtain GPIO state

The function of this command is to obtain the input and output states of GPIO

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x47	——	LSB	MSB

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				OutputPin		
0x06	0xXX	0x47	0x00	0xXX	LSB	MSB

Parameter definition

**OutputPin:** the input and output states of GPIO. Bit0 represents IN1 pin status, bit4 and bit5 represent Out1 to Out2 status respectively. All other values are reserved.

### 8.4.12 Obtain the reader unique serial number

The function of this command is to obtain the unique serial number of a reader.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x4c	——	LSB	MSB

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				SeriaNo		
0x09	0xXX	0x4c	0x00	4bytes	LSB	MSB

Parameter definition

**SeriaNo:** 4 bytes, the unique serial number of the reader.

### 8.4.13 Modify tag customised function

The function of this command is to launch the customised utilities of some particular tags, to achieve specific tag function.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
			InlayType		
0x05	0xXX	0x3a	0xXX	LSB	MSB

#### Parameter definition

**InlayType:** the type of tag, the valid value of this parameter is 0 ~ 254.

0 – the default value of this parameter, represent tag type is not specified.

1 – launch the Peek function of Monza4QT tag (the status of tag will temporary change from public to private). Launching this function will make affection on the data reading/writing, block writing, protection word writing and EPC number writing operation.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x06	0xXX	0x3a	0x00	InlayType	LSB	MSB

#### Parameter definition

**InlayType:** the type of tag, the valid value of this parameter is 0 ~ 254.

### 8.4.14 Enable antenna check

The function of this command is to enable/disable the antenna check during tag reading/writing operation.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
			CheckAnt		
0x05	0xXX	0x66	0xXX	LSB	MSB

#### Parameter definition

**CheckAnt:** 1 byte, antenna check switch.

0x00 – disable antenna check;

0x01 – enable antenna check.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x66	0x00	—	LSB	MSB

### 8.4.15 Modify communication interface

The function of this command is to setup communication interface between RRU2881 and host. The setting of this parameter will be preserved even reader power is down. Only one type interface can be selected, user should connect the reader according to this configuration.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
			CommType		
0x05	0xXX	0x6A	0xXX	LSB	MSB

#### Parameter definition

CommType: 1 byte, communication interface type.

0x00 – enable USB connection;

0x01 – enable UART connection.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

#### NOTE:

1. When USB connection is enabled, the UART connection will not be available. Likewise, when UART connection is enabled, USB connection will not be available.
2. After modification on communication interface, the setting will not valid until the reader is being properly power off and restart again.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x6A	0x00	—	LSB	MSB

### 8.4.16 Modify or load Antenna return loss threshold configuration

The function of this command is to modify or load the return loss threshold for antenna check. The default return loss threshold is -6dB. The setting of this parameter will be preserved even reader power is down.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
			ReturnLoss		
0x05	0xXX	0x6e	0xXX	LSB	MSB

#### Parameter definition

**ReturnLoss:** action on the Return loss configuration.

bit7: load/modify return loss flag.

0 – load the current return loss configuration;

1 – modify return loss configuration.

bit6 ~ bit0: return loss configuration

For bit7 = 0, bit6 ~ bit0 will be ignored.

For bit7 = 1, bit6 ~ bit0 is the desired return loss configuration. The range of return loss is 0 ~ 20, according to 0dB ~ 20dB.

Other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
				ReturnLoss		
0x06	0xXX	0x6e	0x00	0xXX	LSB	MSB

#### Parameter definition

**ReturnLoss:** current return loss configuration.

bit7: reserved, default setting is 0.

bit6 ~ bit0: the current antenna return loss threshold. The range is 0 ~ 20.

#### 8.4.17 Modify maximum EPC/TID length configuration for memory buffer

The function of this command is to modify the maximum EPC/TID length for inventory with memory buffer. This modification will clear all the tag data previously stored in the memory buffer.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
			SaveLen		
0x05	0xXX	0x70	0xXX	LSB	MSB

#### Parameter definition

**SaveLen:** 1 byte, the maximum EPC/TID length.

0 – 128bit length, i.e. 16 bytes. The maximum storable tag amount is 528 for this setting;

1 – 496bit length, i.e. 62 bytes. The maximum storable tag amount is 160 for this setting.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x70	0x00	—	LSB	MSB

### 8.4.18 Load the maximum EPC/TID length configuration

The function of this command is to load the configuration of maximum EPC/TID length for reader memory buffer

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x71	—	LSB	MSB

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x06	0xXX	0x71	0x00	SaveLen	LSB	MSB

#### Parameter definition

**SaveLen:** 1 byte, the maximum EPC/TID length.

0 – 128bit length, i.e. 16 bytes;

1 – 496bit length, i.e. 62 bytes.

### 8.4.19 Obtain data from memory buffer

The function of this command is to obtain all the tag information from reader memory. This command will not cause affection on the status of the stored data. Data can be access multiple times. On the other hand, the tag inventory command (refer to Chapter 8.2.1) will clear all the data from memory buffer.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x72	—	LSB	MSB

#### Response frame:

Len	Adr	reCmd	Status	Data[]		CRC-16	
				Num	EPC Data		
0xXX	0xXX	0x72	0xXX	0xXX	EPC-1, EPC-2, ... EPC-n	LSB	MSB

#### Parameters definition

**Status:** the status of response, detail description is shown in the following table

Status	Description
0x01	operation completed, reader will report all inquired tags information to host
0x03	reader is not able to response all the data in a single frame, further data will be transmitted in the following frames.

**Num:** 1 byte, the amount of EPC/TID included in this response.

**EPC Data:** tag EPC/TID data stored in the memory buffer.

EPC-n				
Ant	Len	EPC/TID	RSSI	Count
0xXX/0XXXXX	0xXX	N bytes	0xXX	0xXX

**Ant:** 1/4/8 port reader 1 bytes, indicates which antenna had inquired the tag. For example, 0x05 is 00000101 in binary, this indicates Antenna 3 and Antenna 1.

16 port reader 1 bytes, indicates which antenna had inquired the tag. For example, 0x0005 is 0000000000000101 in binary, this indicates Antenna 3 and Antenna 1.

**Len:** 1 byte, the byte length of the EPC/TID.

**EPC/TID:** EPC number/TID data of tag, the length of data should identical with Len. Every tag EPC number or TID data is the most-significant word and the most-significant byte in a word first.

**RSSI:** 1 byte, the reader signal strength at the time when the tag is inquired.

**Count:** the total detected times for a particular tag.

0xFF – total detection time is larger or equal to 255.

#### 8.4.20 Clear memory buffer

The function of this command is to clear all the stored tag data from the memory buffer.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x73	—	LSB	MSB

**Command frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x73	0x00	—	LSB	MSB

#### 8.4.21 Obtain the total tag amount from memory buffer

The function of this command is to obtain the total tag amount stored in memory buffer.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x74	—	LSB	MSB

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				Count		
0x07	0xXX	0x74	0x00	2 bytes	LSB	MSB

Parameters definition

**Count:** 2 bytes, the total amount of tag stored in the memory buffer, most-significant byte first.

**8.4.22 Modify parameters of real time inventory mode**

The function of this command is to modify the real time inventory associated parameters.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x75	—	LSB	MSB

**Parameters of Data[]:**

Data[]				
TagProtocol	ReadPauseTime	FliterTime	QValue	Session
0xXX	0xXX	0xXX	0xXX	0xXX

Data[]					
MaskMem	MaskAdr	MaskLen	MaskData	AdrTID	LenTID
0xXX	2Bytes	0xXX	variable	0xXX	0xXX

Parameters definition

**TagProtocol:** 1 byte, tag type definition for real time inventory.

0 – EPC C1G2 (ISO18000-6C) tags.

1 – ISO18000-6B tags. Configurations of the following parameters will be ignored for 6B, QValue, Session, MaskMem, MaskAdr, MaskLen, MaskData, AdrTID, LenTID.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**ReadPauseTime:** 1 byte, time break between 2 real time inventories.

0x00 – 10ms;

0x01 – 20ms;

0x02 – 30ms;

0x03 – 50ms;

0x04 – 100ms.

All other values are reserved, reader will return a parameter error status in the response frame if

other value is delivered in this field.

**FliterTime:** 1 byte, tag filtering time. The valid value of this parameter is 0 ~ 255, corresponds to  $(0 \sim 255) \times 1s$ . In real time inventory, if reader detects a particular tag for more than 1 time, reader will only upload tag information of this tag once within the pre-defined filtering time. For FliterTime = 0, disable tag filtering function.

**QValue:** 1 byte.

bit7: reserved, set to 0.

bit6: strategy indicator.

0 – general strategy;

1 – special strategy.

bit5 ~ bit0: the original Q-value of the EPC tag inventory, the setting of Q-value should follow the rule:  $2^Q \approx$  total amount of tags within the effective field. The range of Q-value is 0 ~ 15, if other value is delivered in this field, reader will return a parameter error status in the response frame.

**Session:** 1 byte, the Session-value of the EPC tag inventory.

0x00 – apply S0 as Session value;

0x01 – apply S1 as Session value;

0x02 – apply S2 as Session value;

0x03 – apply S3 as Session value;

0xff – apply reader smart configuration (only valid in EPC inventory).

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

1. When MaskMem, MaskAdr, MaskLen and MaskData are not provided, no mask pattern will be involved in inventory
2. In real time inventory mode, reader will apply the predefined antenna and inventory time parameters for inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x75	0x00	—	LSB	MSB

**8.4.23 Modify reader working mode**

The function of this command is to modify reader working mode.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
			ReadMode		
0x05	0xXX	0x76	0xXX	LSB	MSB

Parameters definition

**ReadMode:** 1 byte, the current working mode of reader.

0 – answering mode;

1 – real time inventory mode;

2 – real time inventory mode with trigger. For this configuration, reader will only perform inventory when the TTL level of GPI1 port is low. Once TTL level of GPI1 rise high, reader will enter standby mode.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x76	0x00	—	LSB	MSB

**NOTE:**

The setting of this parameter will be preserved even reader power is down, i.e. reader will apply the previous working mode setting on every restart.

**Output format in real time inventory mode:**

Once reader is set to real time inventory mode, reader will perform auto inventory according to predefined configuration.

If a tag is inquired and the tag matches the predefined condition, reader will automatically upload data in the following format:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0xXX	0xXX	0xee	0xXX	—	LSB	MSB

**Status:** the status of response, detail description is shown in the following table

Status	Description
--------	-------------

0x00	Detected appropriate tag, reader will also report the detected tag data.
0x28	No appropriate tag detected within the heartbeat packet time interval, reader will upload heartbeat packet.

(1) Response status is 0x00, parameters of Data[] are shown as below,

Data[]			
Ant	Len	EPC/TID	RSSI
0xXX	0xXX	N bytes	0xXX

#### Parameters definition

**Ant:** 1/4/8 reader indicates which antenna had inquired the tag. For example, 0x05 is 00000101 in binary, this indicates Antenna 3 and Antenna 1.

16 port reader Represented by number, 0~15 respectively represent antenna 1~antenna 16.

**Len:** 1 byte, the byte length of the EPC/TID.

**EPC/TID:** EPC number/TID data of tag, the length of data should identical with Len. The unit of EPC C1 G2 data is word. Every tag EPC number or TID data is the most-significant word and the most-significant byte in a word first.

**RSSI:** 1 byte, the reader signal strength at the time when the tag is inquired.

(2) Response status is 0x28, parameters of Data[] are shown as below

Data[]		
PacketNo	AntStatus	TotalCount
4 bytes	1/4/8/16 bytes	4 bytes

#### Parameters definition

**PacketNo:** heartbeat serial number. 4 bytes, most-significant byte first.

Serial number will increase for ever heartbeat packet uploading operation. If heartbeat serial number exceeds the maximum 4 bytes value, the serial number will recount again from 0.

**AntStatus:** the current status of antennas, queued from 1 to 16 according to byte order.

0x00 – target antenna is idle, not a effective inventory antenna

0x01 – target antenna is working properly;

0x02 – target antenna is disconnected.

**TotalCount:** the total amount of tag inquired since the beginning of the current auto-inventory. A tag being indentified multiple times will also increase the value of TotalCount. 4 bytes, with most-significant byte first. If total amount of inquired tag exceeds maximum 4 bytes value, TotalCount will recount again from 0.

For real time inventory mode, reader will only response to 3 commands:

1. obtain reader information (refer to Chapter 8.4.1);
2. modify reader working mode(refer to Chapter 8.4.22);
3. obtain auto-mode parameters(refer to Chapter 8.4.23).

#### 8.4.24 Obtain reader working mode

The function of this command is to obtain parameters of reader working mode.

##### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x77	——	LSB	MSB

##### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0xXX	0xXX	0x77	0x00	ReadMode, TagProtocol, ReadPauseTime, FliterTime, QValue, Session, MaskMem, MaskAdr, MaskLen, MaskData, AdrTID, LenTID	LSB	MSB

##### Parameters definition

**ReadMode:** 1 byte, the current working mode of reader.

**TagProtocol:** 1 byte, tag type definition of real time inventory.

**ReadPauseTime:** 1 byte, time break between 2 real time inventories.

**FliterTime:** 1 byte, tag filtering time of real time inventory.

**QValue:** 1 byte.

bit7: reserved, set to 0.

bit6: strategy indicator.

0 – general strategy;

1 – special strategy.

bit5 ~ bit0: the original Q-value of the EPC tag inventory, the setting of Q-value should follow the rule:  $2^Q \approx$  total amount of tags within the effective field. The range of Q-value is 0 ~ 15, if other value is delivered in this field, reader will return a parameter error status in the response frame.

**Session:** 1 byte, the Session-value of real time mode for EPC inventory.

**MaskMem, MaskAdr, MaskLen, MaskData:** mask condition for **EPC C1G2** tag inventory in real time mode. MaskMem and MaskLen are 1 byte long respectively. MaskAdr is 2 bytes long, most-significant byte first. MaskData has constant length of 32 bytes, fill the over MaskLen value

content with zero.

**AdrTID:** initial word address of TID memory inventory.

**LenTID:** data length for TID inventory operation.

#### 8.4.25 Load/modify heartbeat packet time break of real time inventory

The function of this command is to modify the heartbeat packet upload time break for real time inventory mode. For HeartBeatTime = 0, disable heartbeat packet delivery.

##### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
			HeartBeatTime		
0x05	0xXX	0x78	0xXX	LSB	MSB

##### Parameter definition

**HeartBeatTime:** time break between 2 heartbeat packets deliveries

bit7: action on heartbeat time break configuration

0 – load heartbeat time break configuration;

1 – modify heartbeat time break configuration.

bit6 ~ bit0: heartbeat time break configuration

For bit7 = 0, bit6 ~ bit0 will be ignored.

For bit7 = 1, bit6 ~ bit0 is the configuration of time break between 2 heartbeat packets deliveries

For HeartBeatTime = 0, disable heartbeat packet delivery.

##### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
				HeartBeatTime		
0x06	0xXX	0x78	0x00	0xXX	LSB	MSB

##### Parameter definition

**HeartBeatTime:** the current configuration of heartbeat packet time break.

bit7: reserved, default value is 0.

bit6 ~ bit0: the current configuration of heartbeat packet time break. The valid value of this parameter is 0 ~ 127, corresponds to (0 ~ 127)\*30s.

##### NOTE:

Reader will only deliver heartbeat packet if no tag is detected for a consecutive period (longer than the heartbeat time break)

### 8.4.26 Modify RF power configuration separately for write operations

The function of this command is to modify RF power for write operations. The default value of the power parameter is 0x00. The setting of this parameter will be preserved even reader power is down.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
			WritePower		
0x05	0xXX	0x79	0xXX	LSB	MSB

#### Parameter definition

**WritePower:** configuration of the power.

bit7: enable/disable separate RF power configuration for write operation.

0 – disable separate RF power configuration, power will be identical for both read and write commands.

1 – enable separate RF power configuration. The valid value of RF power is 0~30, for setting of 30, the output power is approximately 1W.

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x79	0x00	—	LSB	MSB

The separate RF power will take effect for the following commands,

#### EPC C1G2 (ISO18000-6C) commands

- write data command;
- write EPC number command;
- Extended data writing command with assigned mask;
- block erase command;
- block writing command.

#### ISO18000-6B command

- write data command

### 8.4.27 Load the RF power configuration of write operations

The function of this command is to load the predefined separate RF power configuration of write operation.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x7A	—	LSB	MSB

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				WritePower		
0x06	0xXX	0x7A	0x00	0xXX	LSB	MSB

Parameter definition

**WritePower:** configuration of the power.

bit7: configuration of separate RF power configuration for write operation.

0 – disabled separate RF power configuration, power will be identical for both read and write commands.

1 – enabled separate RF power configuration. The valid value of RF power is 0~30, for setting of 30, the output power is approximately 1W.

**8.4.28 Modify or load maximum write retry time configuration**

The function of this command is to modify or load the maximum write retry time configuration. For tag writing and associated commands, if failure occurred, reader will reattempt the operation until success or reach the maximum write retry time. Default value of the retry time is 3. The setting of this parameter will be preserved even reader power is down.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
			RetryTimes		
0x05	0xXX	0x7B	0xXX	LSB	MSB

Parameter definition

**RetryTimes:** action on maximum write retry time configuration

bit7: load/modify maximum write retry time configuration

0 – load maximum write retry time;

1 – modify maximum write retry time;

bit6 ~ bit3: reserved.

bit2 ~ bit0: maximum write retry time

For bit7 = 0, bit6 ~ bit0 will be ignored.

For bit7 = 1, bit2 ~ bit0 is the configuration of maximum retry time. The range of this parameter is 0 ~ 7.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				RetryTimes		
0x06	0xXX	0x7B	0x00	0xXX	LSB	MSB

Parameter definition

**RetryTimes:** value of maximum write retry time

bit7 ~ bit3: reserved, default value is 0.

bit2 ~ bit0: maximum write retry time setting. The valid range of RetryTimes is 0 ~ 7.

The retry time counter will take effect for the following command,

#### **EPC C1G2 (ISO18000-6C) commands**

- write data command;
- write EPC number command;
- Extended data writing command with assigned mask;
- block erase command;
- block writing command.

#### **ISO18000-6B command**

- write data command

### **8.4.29 Modify password of tag customised functions**

The function of this command is to modify the access password of tag customised functions. This password will take effect when the Peek function of Monza4QT tag is launched. The setting of this parameter will be preserved even reader power is down.

#### **Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
			UserPwd	LSB	MSB
0x08	0xXX	0x7D	4 bytes	LSB	MSB

#### Parameter definition

**UserPwd:** 4 bytes, most-significant byte first. User customised access password.

#### **Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x7D	0x00	—	LSB	MSB

### **8.4.30 Obtain password of tag customised functions**

The function of this command is to obtain the access password of tag customised functions.

#### **Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x7E	—	LSB	MSB

#### **Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				UserPwd	LSB	MSB
0x09	0xXX	0x7E	0x00	4 bytes	LSB	MSB

Parameter definition

**UserPwD:** 4 bytes, most-significant byte first. User customised access password.

**8.4.31 Load/modify reader profile**

The function of this command is to modify / load the internal profile configuration of reader. The default value of profile is 1. The setting of this parameter will be preserved even reader power is down.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
			profile		
0x05	0xXX	0x7F	0xXX	LSB	MSB

Parameter definition

**profile:** profile number configuration.

bit7: action on profile number configuration.

0 – load profile number;

1 – modify profile number.

Bit6: Power Down Save Flag (**Ex10 effective**).

0 – save;

1 – not save.

Bit5 ~ bit0: profile number configuration.

For bit7 = 0, bit6 ~ bit0 will be ignored.

For bit7 = 1, bit5 ~ bit0 is the new configuration of profile number.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				profile		
0x06	0xXX	0x7F	0x00	0xXX	LSB	MSB

Parameter definition

**profile:** current profile number configuration.

bit7: reserved, default value is 0.

bit6 ~ bit0: the current profile number of reader. The valid range of this parameter is 0 ~ 3.

**Configuration of different profiles for R2000:**

Profile0: Tari 25uS, FM0 40KHz;

Profile1: Tari 25uS, Miller 4 250KHz (recommend configuration, system default setting)

Profile2: Tari 25uS, Miller 4 300KHz;

Profile3: Tari 6.25uS, FM0 400KHz.

**Configuration of different profiles for Ex10:**

Profile11: 640kHz, FM0, Tari 7.5 us  
 Profile1: 640kHz, Miller2, Tari 7.5 us  
 Profile15: 640kHz, Miller4, Tari 7.5 us  
 Profile12: 320kHz, Miller2, Tari 15 us  
 Profile3: 320kHz, Miller2, Tari 20 us  
 Profile5: 320kHz, Miller4, Tari 20 us  
 Profile7: 250kHz, Miller4, Tari 20 us  
 Profile13: 160kHz, Miller8, Tari 20 us  
 Profile50: 640kHz, FM0, Tari 6.25 us  
 Profile51: 640kHz, Miller2, Tari 6.25 us  
 Profile52: 426kHz, FM0, Tari 15 us  
 Profile53: 640kHz, Miller4, Tari 7.5 us

**8.4.32 Synchronise EM4325 timestamp**

The function of this command is to broadcast the timestamp to all the EM4325 tags within the effective field.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
			UTC Time		
0x08	0xXX	0x85	4 bytes	LSB	MSB

**Parameter definition**

**UTC Time:** 4 bytes, the UTC timestamp to be broadcasted, most-significant byte first. It represents the second amount since 1970-01-01 00:00:00. Synchronisation will fail if the MSB of UTC Time is 0.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x85	0x00	—	LSB	MSB

**NOTE:**

- EM4325 tag must stay in BAP mode to receive and synchronise timestamp.
- Synchronisation will fail if the delivered MSB of UTC Time is 0.
- Synchronisation will fail if any alert flag is set. Alert flag can be cleared via the “Reset EM4325 alert” command or temporarily disabled alert functions via modifying associated EM4325 registers.
- EM4325 tags will not produce response to timestamp synchronisation command. Hence reader will report successful status to host once timestamp synchronisation command is delivered to tag.
- Once the EM4325 internal UTC Time register is setup, this register will increase 1 in every

second. This timer will stop if the 4 bytes register overflow 0xFFFFFFFF and back to 0 or the associated system configuration register in user area is modified.

### 8.4.33 Obtain EM4325 temperature data

The function of this command is to obtain temperature data from EM4325 tag.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x86	—	LSB	MSB

#### Parameters of Data[]:

Data[]				
ENum	EPC	SendUID	NewSample	Pwd
0xXX	variable	0xXX	0xXX	4 bytes
MaskMem	MaskAdr	MaskLen	MaskData	
0xXX	2Bytes	0xXX	variable	

#### Parameter definition

##### ENum:

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMen, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**SendUID:** 1 byte, UID response flag.

0x00 – do not response UID;

0x01 – response UID.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**NewSample:** 1 byte, new temperature data flag.

0x00 – obtain the previous temperature data;

0x01 – obtain a new temperature data.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Psw should be set to all-zero or set to the correct access password.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

#### NOTE:

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

#### Response frame:

Len	Adr	reCmd	Status	Data[]			CRC-16	
				UID	SenserData	UTC		
0xXX	0xXX	0x86	0x00	variable	4Bytes	4Bytes	LSB	MSB

#### Parameters definition

**UID:** unique identification number of EM4325 tag.

For SendUID = 0, no UID field will be delivered in the response;

For SendUID = 1, UID field will be delivered in the response, the length of the field depends on the tag, for different tag response, the length of UID may alter between 8 bytes, 10 bytes of 12 bytes.

**SenserData:** temperature data returned from EM4325 tag, the length of data is 4 bytes, most-significant byte first. Refer to Sensor Data register definition in EM4325 datasheet for detail description.

**UTC:** 4 bytes, current UTC timestamp of tag, most-significant byte first. It represent the second amount since 1970-01-01 00:00:00. If the target tag had not been synchronised or tag UTC timestamp had not been setup, the return value will be all-zero.

### 8.4.34 Obtain external data via EM4325 SPI

The function of this command is to use EM4325 tag as SPI master device, and obtain data from its connected SPI slave device.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x87	—	LSB	MSB

#### Parameters of Data[]:

Data[]						
SPICmdSize	ENum	EPC	SPIResSize	SPISclk	SPIInitDelay	SPIInterval
0xXX	0xXX	variable	0xXX	0xXX	0xXX	0xXX
SPICmd	Pwd	MaskMem	MaskAdr	MaskLen	MaskData	
variable	4 bytes	0xXX	2 bytes	0xXX	variable	

#### Parameters definition

##### ENum:

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMen, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this command.

**SPICmdSize:** 1 byte, the byte length of SPI command. The valid range of SPICmdSize is 1 ~ 8.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**SPIResSize:** 1 byte, the byte length of response from SPI slave device. The valid range of SPIResSize is 0 ~ 7. SPIResSize = 0 represents no response will be produced.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**SPISclk:** 1 byte, clock of the SPI bus.

0 – 40KHz;  
 1 – 80KHz;  
 2 – 160KHz;  
 3 – 320KHz.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**SPIInitDelay:** 1 byte, delay for SPI clock initialisation.

0 – 1 SCLK;  
 1 – 50us;  
 2 – 500us;  
 3 – 5ms.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**SPIInterval:** 1 byte, time break between every SPI data byte.

0 – no delay;  
 1 – 50us;  
 2 – 100us;  
 3 – 500us.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**SPICmd:** SPI command data to be delivered, the length of data must be identical to the value stated in SPICmdSize.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Psw should be set to all-zero or set to the correct access password.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;  
 0x02 – TID memory;  
 0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				SPIResData		
0xXX	0xXX	0x87	0x00	variable	LSB	MSB

Parameter definition

**SPIResData:** data returned from EM4325 SPI slave device, the valid range of data is 0 ~ 7 bytes. The length of SPIResData should be identical with the value stated in SPIResSize.

**NOTE:**

Before obtaining external data via EM4325 SPI, the EM4325 I/O control byte (I/O Control Word) register must be firstly configured. User should enable SPI function, setup SPI port and enable SPI pin pull-up register.

**8.4.35 Reset EM4325 alert**

The function of this command is to reset all the alert flag of EM4325 tag. UTC clock and monitoring functions will be clear.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x88	—	LSB	MSB

**Parameters of Data[]:**

Data[]						
ENum	EPC	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	variable	4 bytes	0xXX	2 bytes	0xXX	variable

Parameters definition**ENum:**

0x00 ~ 0x0f – ENum represents the length of EPC number. The unit of ENum is word, and the length of EPC should be less than 15 words. In this situation, parameters MaskMem, MaskAdr, MaskLen, MaskData should all be left empty.

0xff – MaskMem, MaskAdr, MaskLen, MaskData should be provided instead of EPC parameters.

Reader will return a parameter error status in the response frame if other value is delivered in this field.

**EPC:** EPC number of the target tag. Length of the EPC depends on the provided EPC number. The unit of EPC number is word. During communication, the most-significant word and the most-significant byte in a word will be transmitted first. It is essential to provide FULL EPC in this

command.

**Pwd:** 4 bytes, it is the 4 bytes access password of a tag. From left to right, the first two bytes are the high bytes of access password, and the last two bytes are the low bytes. The most-significant bit of a tag 32-bit access password is the most-significant bit in the 1st byte of Pwd (from left to right). Likewise, the least-significant bit of a tag 32-bit access password is the least-significant bit in the 4th (last) byte of Pwd. Psw should be set to all-zero or set to the correct access password.

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.

**NOTE:**

When MaskMem, MaskAdr, MaskLen and MaskData are not provided, full EPC number will be used as mask pattern in inventory.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x88	0x00	—	LSB	MSB

**NOTE:**

Before using the “reset EM4325 alert” command, the temperature sensor control byte of EM4325I (Temp Sensor Control Word 1) must be firstly configured to enable the EM4325 alert reset function; otherwise the EM4325 tag will not produce any response to this command.

#### 8.4.36 Modify or load DRM configuration

The function of this command is to modify or load DRM switch mode. Setting of these parameters will lost during power off. DRM will be reset to off state on every reader start-up.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16
			DRMMode	

0x05	0xXX	0x90	0xXX	LSB	MSB
------	------	------	------	-----	-----

Parameter definition

**DRMMode:** action on DRM configuration.

bit7: load/modify DRM flag.

0 – load the current DRM configuration;

1 – modify DRM configuration.

bit6 ~ bit0: DRM configuration

For bit7 = 0, bit6 ~ bit0 will be ignored.

For bit7 = 1, bit6 ~ bit0 is the desired DRM configuration.

0 – Switch OFF DRM;

1 – Switch ON DRM.

Other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				DRMMode		
0x06	0xXX	0x90	0x00	0xXX	LSB	MSB

Parameter definition

**DRMMode:** action on DRM configuration.

bit7: reserved, the default setting is 0.

bit6 ~ bit0: current DRM configuration

0 – DRM is switch OFF;

1 – DRM is switch ON.

**8.4.37 Measure antenna return loss**

The function of this command is to measure antenna return loss on specific frequency point.

**Command frame:**

Len	Adr	Cmd	Data[]		CRC-16	
			TestFreq	Ant		
0x09	0xXX	0x91	4 bytes	0xXX	LSB	MSB

Parameters definition

**TestFreq:** target frequency point for the return loss measurement. The unit of TestFreq is KHz, most-significant byte first. A valid frequency point must be a multiple of either 125KHz or 100KHz. Reader will return a parameter error status in the response frame if other value is delivered in this field.

**Ant:** 1 byte, target antenna for this measurement

0 – antenna 1;

- 1 – antenna 2;
- 2 – antenna 3;
- 3 – antenna 4.

Other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				ReturnLoss		
0x06	0xXX	0x91	0x00	0xXX	LSB	MSB

Parameter definition

**ReturnLoss:** result of the return loss measurement, the unit of ReturnLoss is dB.

### 8.4.38 Measure current reader temperature

The function of this command is to obtain reader current temperature.

**Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x92	—	LSB	MSB

**Response frame:**

Len	Adr	reCmd	Status	Data[]		CRC-16	
				PlusMinus	Temp		
0x07	0xXX	0x92	0x00	0xXX	0xXX	LSB	MSB

Parameters definition

**PlusMinus:** 1 byte, temperature range.

- 0 – below 0 °c;
- 1 – above 0 °c.

**Temp:** 1 byte, temperature value, the unit is °c.

**NOTE:**

This command only supported by temperature sensor embedded reader.

### 8.4.39 Set extended parameters

The host computer is used to configure various extended parameters of the reader, currently only supports Ex10 series.

**Command frame:**

Len	Adr	Cmd	Data[]			CRC-16	
0x06+n	0xXX	0xEA	Flag	CFGNo	CFGData (n bytes)	LSB	MSB

#### Parameters definition

**Flag:** 1 byte, 0- save when power off; 1- not save when power off;.

**CFGNo:** 1 byte, Used to specify the parameter sequence number to be set in the command data block.

**CFGData:** Parameter content, see [Reader Parameter Configuration](#) for details.

#### **Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0xEA	0x00	—	LSB	MSB

### 8.4.40 Set extended parameters

The host computer is used to obtain the extended parameter information of the reader, currently only supports Ex10 series

#### **Command frame:**

Len	Adr	Cmd	Data[]	CRC-16	
			CFGNo		
0x05	0xXX	0xEB	0xXX	LSB	MSB

**CFGNo:** 1 byte, Used to specify the parameter sequence number to be set in the command data block.

#### **Response frame:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				CFGData		
0x05+n	0xXX	0xEB	0x00	n(byte)	LSB	MSB

#### Parameters definition

**CFGData:** Parameter content, see [Reader Parameter Configuration](#) for details.

### 8.4.41 Start fast inventory

The host computer is used to start fast inventory, currently only supports Ex10 series **Command frame**:

Len	Adr	Cmd	Data[]	CRC-16	
			target		
0x05	0xXX	0x50	0xXX	LSB	MSB

#### Parameters definition

**Target:** 1 byte, 0 –target A;1-target B;

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x50	0x00	——	LSB	MSB

If a tag is inquired and the tag matches the predefined condition, reader will automatically upload data in the following format:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0xXX	0xXX	0xee	0x00	——	LSB	MSB

parameters of Data[] are shown as below,

Data[]			
Ant	Len	EPC/TID	RSSI
0xXX	0xXX	N bytes	0xXX

#### Parameters definition

**Ant:** indicates which antenna had inquired the tag. For example, 0x05 is 0101 in binary, this indicates Antenna 3 and Antenna 1.

**Len:** 1 byte, the byte length of the EPC/TID.

**EPC/TID:** EPC number/TID data of tag, the length of data should identical with Len. The unit of EPC C1 G2 data is word. Every tag EPC number or TID data is the most-significant word and the most-significant byte in a word first.

**RSSI:** 1 byte, the reader signal strength at the time when the tag is inquired.

### 8.4.42 Stop fast Inventory

The host computer is used to stop fast inventory, currently only supports Ex10 series..

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x51	——	LSB	MSB

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x51	0x00	——	LSB	MSB

### 8.4.43 Read Antenna Power

This command is used to read the output power information of each antenna port of the reader/writer.

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x94	——	LSB	MSB

#### Response frame:

Len	Adr	reCmd	Status	Data[]	CRC-16	
				Pwrs		
0x05+n	0xXX	0x51	0x00	nbytes	LSB	MSB

#### Parameters definition

**Pwrs:** The output power of the current reader/writer. The length is consistent with the number of antenna ports, with 4 bytes for 4-port readers, 8 bytes for 8 ports, and 16 bytes for 16 ports; Output power from left to right is antenna 1, antenna 2, antenna 3, antenna 4,, in order.

### 8.4.44 Stop Inventory

This command is used to immediately stop the command currently being executed by the reader and return the current command reply. This command only applies to the command to immediately stop querying labels (command code 0x01).

#### Command frame:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x93	——	LSB	MSB

This command has no response.

The reader/writer will end the currently executing command within 1 second after receiving it.

## 8.5 Reader customised commands

All configuration parameters of the reader are listed in it, which is convenient for users to set and read in a unified manner.

To read the parameters, use the get reader parameter command, and use the set reader configuration command to set the parameters. After modifying the configuration, except for the network configuration that needs to be restarted to take effect, other configurations take effect immediately. (Currently only supports Ex10 series)

### 8.5.1 Reader configuration parameter table

configuration table:

No.	Name	Length(byte)	describe
7	scan parameter	3	Parameter information used in the scan process
8	TagFocus	1	TagFocus switch
9	QAndSession	2	Q and Session parameter
10	TIDparam	2	TID parameter
11	MaskParam	N	Mask parameter

### 8.5.2 Reader parameter function description

#### 8.5.2.1 CFG7: Scan parameter

This configuration is used to set the parameters used by the reader during the scan process

Byte	0	1	2
Contents	Interval	DwellTime	Count
Default	3	0	5

**Interval:** Interval time between queries, range 0~6, unit 10ms;

**DwellTime:** The query time, the range is 2~255, the unit is 100ms; when the tagfocus switch is turned on, it is fixed to 200ms.

**Count:** The number of queries, only valid for S1, S2, S3, the range is 0~5;

#### 8.5.2.2 CFG8: TagFocus switch

This configuration is used to set the switch of TagFocus

Byte	0
Contents	TagFocus
Default	0

**TagFocus:** 0-disable,1-enable.

### 8.5.2.3 CFG9: Q And Session

This configuration is used to set the anti-collision parameters used by the reader during the quick query process.

Byte	0	1
Contents	Q	Session
Default	6	1

**Q:** range is 0~15.

**Session:**0-S0;1-S1;2-S2;3-S3;255-Auto;

### 8.5.2.4 CFG10: TID Param

This configuration is used to set the TID parameters used by the reader during the fast query process.

Byte	0	1
Contents	AdrTID	LenTID
Default	6	1

**AdrTID:** Query the starting word address of the TID area.

**LenTID:** Query the number of data words in the TID area. LenTID ranges from 0 to 15. When the value is 0, it means EPC query.

### 8.5.2.5 CFG11: Mask Param

This configuration is used to set the mask parameter used by the reader during the fast query process..

Byte	MaskMem	MaskAdr	MaskLen	MaskData
Contents	1	2	1	MaskLen
Default	1	0x0020	0	0x00

**MaskMem:** 1 byte, mask area indication.

0x01 – EPC memory;

0x02 – TID memory;

0x03 – User memory.

All other values are reserved, reader will return a parameter error status in the response frame if other value is delivered in this field.

**MaskAdr:** 2 bytes, entry bit address of the mask, the valid range of MaskAdr is 0 ~ 16383.

**MaskLen:** 1 byte, bit length of mask (unit: bits).

**MaskData:** mask data. The length of MaskData equals to MaskLen/8. If MaskLen is not a multiple of 8 integer, the length of MaskData is equal to the  $\text{int}[\text{MaskLen}/8]+1$ . Non-specified lower significant figures should be filled up with 0.