

**UHFREADER288A12 UHF RFID
Reader User's Manual v1.0**

Content

1. COMMUNICATION INTERFACE SPECIFICATION	1
2. PROTOCOL DESCRIPTION	1
3. DATA BLOCK FORMAT	2
3.1 COMMAND DATA BLOCK	2
3.2 RESPONSE DATA BLOCK.....	2
4. OPERATION COMMAND (CMD) SUMMARY.....	3
4.1 EPC C1G2 (ISO18000-6C) COMMAND.....	3
4.2 18000-6B COMMAND.....	4
4.3 READER DEFINED COMMAND.....	4
5. LIST OF COMMAND EXECUTION RESULT STATUS.....	6
6. TAG RETURNED ERROR CODE.....	8
7. TAG FEATURES REQUIRING ATTENTION.....	8
8. DETAILED DESCRIPTION OF OPERATION COMMANDS	9
8.1 OVERVIEW	9
8.2 EPC C1G2 COMMAND	9
8.2.1 Inventory.....	9
8.2.2 Read Data.....	12
8.2.3 Write Data.....	13
8.2.4 Write EPC.....	14
8.2.5 Kill Tag.....	15
8.2.6 Lock.....	16
8.2.7 BlockErase.....	18
8.2.8 Set Privacy with Mask Pattern.....	19
8.2.9 Set Privacy without Mask Pattern.....	20
8.2.10 Reset Privacy.....	20
8.2.11 Check Privacy.....	20
8.2.12 EAS Configure.....	21
8.2.13 EAS Alarm.....	22
8.2.14 Block Write.....	22
8.2.15 Read Monza4QT parameters.....	23
8.2.16 Set Monza4QT parameters.....	25
8.2.17 Extended Read Data.....	26
8.2.18 Extended Write Data.....	27
8.2.19 Mix Inventory.....	28
8.3 18000-6B COMMAND.....	31
8.4 READER-DEFINED COMMAND.....	31
8.4.1 Get Reader Information.....	31
8.4.2 Set Region.....	33
8.4.3 Set Address.....	34
8.4.4 Set InventoryScanTime.....	34
8.4.5 Set Baud Rate.....	34
8.4.6 Set RF Power.....	35

8.4.7 BuzzerAndLED Control	35
8.4.8 Set Work Mode	36
8.4.9 Get System Parameters	37
8.4.10 Set EAS Sensitivity.....	38
8.4.11 Set Mask	39
8.4.12 Set Response Style of Auto-running Mode.....	39
8.4.13 Set Inventory Interval	40
8.4.14 Select Tag Type	40
8.4.15 Set Antenna Multiplexing	41
8.4.16 Set Beep Notification.....	42
8.4.17 Set Real-time Clock.....	42
8.4.18 Get Time	43
8.4.19 Get Tag Buffer Info.....	43
8.4.20 Clear Tag Buffer	44
8.4.21 Set Relay	44
8.4.22 Set GPIO.....	45
8.4.23 Get GPIO Status.....	45
8.4.24 Set Trigger Delay Time.....	46
8.4.25 Set TID-Inventory Parameters	46
8.4.26 Get Reader Serial Number	46
8.4.27 Change AT Mode	47
8.4.28 AT Transparent Command	47
8.4.29Tag custom function.....	48
8.4.30 Set antenna checked.....	48
8.4.31 Set Write Power	49
8.4.32 Get Write Power.....	49
8.4.33 Get or Set Rerty times.....	49
8.4.35 Set detect multiple tag parameter.....	50
8.4.36 Get detect multiple tag parameter	51
8.4.37 R2000 module ROM update	51

1. COMMUNICATION INTERFACE SPECIFICATION

The reader communicates with host (MCU, MPU, Controller) using serial communication interface RS232/RS485 or TCPIP and complete corresponding operation according to the host command. The default serial communication parameter is 57600bps 1 start bit, 8 data bits, 1 stop bit without parity check bit. In the process of communication, the least significant bit of one byte is transmitted first and the least significant byte of command data sequence is transmitted first.

2. PROTOCOL DESCRIPTION

A communication procedure is sponsored by the host sending commands and data to the reader and the reader returns the result status and data to host after command execution.

Reader executes a command after receiving it. Only after command execution completed, the reader would be able to receive another command. During the implementation of one command, the reader ignores all other command data received.

The following table shows the process of the host computer sending command:

HOST	DIRECTION	READER
Command Data Block	→	

The interval between two consecutive bytes in the command data block should be less than 15ms. During command data block sending, synchronization will lost if the host receives any data from the reader and the host should stop command sending and restart the communication after 15ms.

The reader completes command execution after receiving host command and returns the results. During the period, it doesn't process any host data. The feedback of command execution results is as follows:

READER	DIRECTION	HOST
Response Data Block	→	

The interval between two consecutive bytes in the response data block is less than 15ms.

3. DATA BLOCK FORMAT

3.1 COMMAND DATA BLOCK

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

Remark:

	LENGTH(Byte)	COMMENT
Len	1	Command data block length 1 byte (not including itself). Value range is 4~96. The number of Len equals the length of Data [] plus 4.
Adr	1	Reader address, 1 byte. Value range is 0~254. Only will the reader conforming to the address response the command data block. Value 255 is broadcasting address. All the readers will response to the command data block with a broadcasting address. The default value is 0.
Cmd	1	Operation command symbol, 1 byte.
Data[]	Variable	Operation command parameters.
LSB-CRC16	1	CRC-16 LSB. CRC-16 checksum, 2 bytes with least significant byte first.
MSB-CRC16	1	CRC-16 MSB.

3.2 RESPONSE DATA BLOCK

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

COMMENT:

	LENGTH(Byte)	COMMENT
Len	1	Response data block length 1 byte (not including itself). The number of Len equals the length of Data [] plus 5.
Adr	1	Reader address, 1 byte. Value rang is 0~254.
reCmd	1	Received command symbol, 1 byte. If the command is unrecognized, the reCmd is 0x00.
Status	1	Result status value, 1byte. Refer to following table for details.
Data[]	Variable	Response data.
LSB-CRC16	1	CRC16 LSB. CRC-16 checksum, 2 bytes with least significant byte first.
MSB-CRC16	1	CRC16 MSB.

The default value of the reader address is 0x00. The host may change it by using reader-defined command "Write Adr".

Cyclic Redundancy Check (CRC) computation includes all data from Len. A reference CRC computation program is presented as follow:

C-Example:

```
#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;
}
```

4. OPERATION COMMAND (CMD) SUMMARY

4.1 EPC C1G2 (ISO18000-6C) COMMAND

NUM	COMMAND	CODE	COMMENT
1	Inventory	0x01	The function is used to inventory tags in the effective field and get their EPC values.
2	Read Data	0x02	The function is used to read part or all of a Tag's Password, EPC, TID, or User memory.
3	Write Data	0x03	The function is used to write several words in a Tag's Reserved, EPC, TID, or User memory.
4	Write EPC	0x04	The function is used to write EPC value in one Tag's EPC memory. The writing process is carried out in a broadcast way.
5	Kill Tag	0x05	The function is used to deactivate (kill) one tag. When killed, the tag will not answer to any command.

6	Lock	0x06	The function is used to set the tag memories' access control policy by setting them as readable, writable or on the verse.
7	BlockErase	0x07	The function is used to erase multiple words in a Tag's Password, EPC, TID, or User memory.
8	Set Privacy with Mask Pattern	0x08	The function is used to set a designated tag into privacy state (Only NXP's UCODE EPC G2X tags available).
9	Set Privacy without Mask Pattern	0x09	The function is used to set one tag in the field into privacy state (Only NXP's UCODE EPC G2X tags available).
10	Reset Privacy	0x0a	The function is used to remove one tag from privacy state(Only NXP's UCODE EPC G2X tags available).
11	Check Privacy	0x0b	The function is used to check one tag if it is in the privacy state (Only NXP's UCODE EPC G2X tags available).
12	EAS Configure	0x0c	The function is used to set or reset the EAS bit of one designated tag (Only NXP's UCODE EPC G2X tags available).
13	EAS Alarm	0x0d	The function is used to check EAS bit status of one tag in the field (Only NXP's UCODE EPC G2X tags available).
14	Block Write	0x10	The function is used to write multiple words in a Tag's Reserved, EPC, TID, or User memory.
15	Read Monza4Qt	0x11	The function is used to get Monza4QT work parameters.
16	Set Monza4Qt	0x12	The function is used to Set Monza4QT work parameters.
17	Extended Read data	0x15	The function is used to read large memory of tag.
18	Extended Write data	0x16	The function is used to write large memory of tag.
19	Mix Inventory	0x19	The function is used to mix inventory.

4.2 18000-6B COMMAND

4.3 READER DEFINED COMMAND

NUM	COMMAND	CODE	CONNECT
1	Get Reader Information	0x21	This function is used to get reader-related information such as reader address (Adr), firmware version, supported protocol type, InventoryScanTime, RF power and frequency range.
2	Set Region	0x22	This function is used to set the work region which defines the lower limit and the upper limit of frequency range.
3	Set Address	0x24	This function is used to set the reader's address. The address value will be stored in reader's inner nonvolatile memory with default value 0x00. The value range is 0x00~0xFE. The address 0xFF is reserved as the broadcasting address. When user tries to write a 0xFF address, the reader will set the value to 0x00 automatically.

4	Set InventoryScanTime	0x25	This function is used to set reader's InventoryScanTim. The value range is 3~255 corresponding to 3*100ms~255*100ms with default 10 (10*100ms).
5	Set Baud Rate	0x28	The function is used to change the serial communication baud rate.
6	Set RF Power	0x2F	The function is used to set the RF power of reader.
7	BuzzerAndLED Control	0x33	The function is used to control the reader's inner buzzer and LED.
8	Set Work Mode	0x35	The function is used to set the reader's work mode.
9	Get System Parameters	0x36	The function is used to get the reader's work mode parameters.
10	Set EAS sensitivity	0x37	The function is used to set EAS operation sensitivity.
11	Set Mask	0x3b	The function is used to set mask condition for G2 tag inventory.
12	Set Response Style of Auto-running mode	0x3c	The function is used to set response format parameters in auto-running mode.
13	Set Inventory Interval	0x3d	The function is used to set inventory interval time.
14	Select Tag Type	0x3e	The function is used to select tags type to be inventoried.
15	Set Antenna Multiplexing	0x3f	The function is used to configure antennae' multiplexing.
16	Set Beep Notification	0x40	The function is used to set beep notification action.
17	Set Real-time Clock	0x41	The function is used to set reader's inner real-time clock.
18	Get Time	0x42	The function is used to read out time of the reader's inner real-time clock.
19	Get Tag Buffer Info	0x43	The function is used to read out all tag records in reader's tag buffer.
20	Clear Tag Buffer	0x44	The function is used to clear reader's tag buffer.
21	Set Relay	0x45	The function is used to control the reader's built-in relay.
22	Set GPIO	0x46	The function is used to set GPIO status.
23	Get GPIO Status	0x47	The function is used to read GPIO status.
24	Set Notification Pulse Output	0x48	The function is used to configure GPIO for pulse notification of tag detection.
25	Set Trigger Delay Time	0x49	The function is used to set the delay time in trigger mode.
26	Set TID-Inventory Parameters	0x4a	The function is used to set the parameters used in TID-Inventory.
27	Get serial Number	0x4C	The function is used to get reader's serial number.
28	AT CMD Change	0x4d	The function is used to change AT mode for network module.
29	Transparent CMD	0x4e	The function is used to transparent cmd for network module
30	Set tag custom Function	0x3a	The function is used to set tag's defined function..
31	Set antenna checked	0x66	The function is used to set antenna auto checked
32	Set write power	0x79	The function is used to set power when do write operation
33	Get write power	0x7A	The function is used to get power when do write operation

34	Set/Get retry times	0x7B	The function is used to set or get retry times when write failed.
35	Set QS	0x81	The function is used to set Q and session for mutiple tag query.
36	Get QS	0x82	The function is used to get Q and session for mutiple tag query.
37	Flash ROM	0x84	The function is used to flash ROM

5. LIST OF COMMAND EXECUTION RESULT STATUS

RESPONSE DATA BLOCK						STATUS COMMENT
Len	Adr	reCmd	Status	Data[]	CRC16	
Length of Data[] +5	0xXX	0xXX	0x00	LSB+MSB	Command is executed successfully and data part contains result data.
Length of Data[] +5	0xXX	0x01	0x01	LSB+MSB	G2 tag inventory operation is completely finished in defined InventoryScanTime and all tags' EPC data are returned.
Length of Data[] +5	0xXX	0x01	0x02	LSB+MSB	G2 tag inventory operation is not completely finished when defined InventoryScanTime overflows and some tags' EPC data are returned.
Length of Data[] +5	0xXX	0x01	0x03	LSB+MSB	The number of the inventoried tags is too much and not all tags' EPC data can be returned within this single response data block. Other response data blocks are followed.
Length of Data[] +5	0xXX	0x01	0x04	LSB+MSB	G2 tag inventory operation is finished since the number of the inventoried tags reaches the reader process limit. All inventoried tags' EPC data are returned.
5	0xXX	0xXX	0x05	-	LSB+MSB	The sent command is password-required operation and the password given in the command is wrong.
5	0xXX	0x05	0x09	-	LSB+MSB	Kill command is not executed successfully. The possible reasons include password error and poor reader to tag air communication.
5	0xXX	0x05	0x0a	-	LSB+MSB	Kill Password is zero. Kill tag operation requires a none-zero password.
5	0xXX	0xXX	0x0b	-	LSB+MSB	The operation not supported by the tag.

5	0xXX	0xXX	0x0c	-	LSB+MSB	Password is zero. Privacy or EAS related operations require a none-zero password (Only NXP's UCODE EPC G2X tags available).
5	0xXX	0x0a	0x0d	-	LSB+MSB	The tag is already in privacy. It does not need to be set into privacy state again (Only NXP's UCODE EPC G2X tags available).
5	0xXX	0x0a	0x0e	-	LSB+MSB	The tag is not in privacy state and does not need to be set out of privacy state (Only NXP's UCODE EPC G2X tags available) or the tag does not support this kind of operation.
5	0xXX	0x53	0x10	-	LSB+MSB	The data block of a 6B tag is locked and can not be rewritten.
5	0xXX	0x55	0x11	-	LSB+MSB	The data block of a 6B tag can not be locked.
5	0xXX	0x55	0x12	-	LSB+MSB	The data block of a 6B tag is already locked and not needed to be re-locked.
5	0xXX	0xXX	0x13	-	LSB+MSB	Reader configuration parameters saving operation failed.
5	0xXX	0xXX	0x14	-	LSB+MSB	The RF power can not be adjusted in current situation.
Length of Data[] +5	0xXX	0x51	0x15	LSB+MSB	6B tag inventory operation is completely finished in defined InventoryScanTime and all tags' data are returned.
Length of Data[] +5	0xXX	0x51	0x16	LSB+MSB	6B tag inventory operation is not completely finished when defined InventoryScanTime overflows and some tags' data are returned.
Length of Data[] +5	0xXX	0x51	0x17	LSB+MSB	The number of the inventoried tags is too much and not all tags' data can be returned within this single response data block. Other response data blocks are followed.
Length of Data[] +5	0xXX	0x51	0x18	LSB+MSB	6B tag inventory operation is finished since the number of the inventoried tags reaches the reader process limit. All inventoried tags' data are returned.
5	0xXX	0xXX	0x19	-	LSB+MSB	EAS related operation failed. The possible reasons include the tag does not support this operation or the password is zero.

Length of Data[] +5	0xXX	0xXX	26		LSB+MSB	Return rate packet when inventory finished
5	0xXX	0xXX	0xF8	-	LSB+MSB	Antenna checked error.
5	0xXX	0xXX	0xF9	-	LSB+MSB	Command execution error.
5	0xXX	0xXX	0xFA	-	LSB+MSB	Operation aborted since the poor reader to tag air communication.
5	0xXX	0xXX	0xFB	-	LSB+MSB	No tag in the field.
6	0xXX	0xXX	0xFC	Err_code	LSB+MSB	Tag operation error and the error code is returned in Err_code.
5	0xXX	0xXX	0xFD	-	LSB+MSB	Command length error.
5	0xXX	0x00	0xFE	-	LSB+MSB	Command can not be recognized since it is a non-exist command or CRC error.
5	0xXX	0xXX	0xFF	-	LSB+MSB	Command parameters error.

6. TAG RETURNED ERROR CODE

EPC C1G2 (ISO18000-6C) Tag returned error code:

	Error-Code	Error-Code Name	Description
Specified error	0x00	Other error	Other errors.
	0x03	Memory override	Memory not exist or not-supported PC value.
	0x04	Memory locked	Memory is locked and can not be rewritten.
	0x0b	Insufficient power	No sufficient power to meet writing operation requirement.
Non-specified error	0x0f	Non-specified error	Non-specified error.

7. TAG FEATURES REQUIRING ATTENTION

A. EPC C1G2 TAG (G2 TAG)

G2 tag memory includes four sector: reserved memory (password memory), EPC memory, TID memory and User memory.

Reserved memory (password memory) is a 4-word sector. The former 2 word is kill password and the latter 2 word is access password. All passwords are available to read and write. Reserved memory can be read/write protected by configuration..

EPC memory contains EPC information of the tag. The address 0 word is CRC checksum of the EPC memory. The address 1 word is PC value indicating the EPC word-unit length. From the address 2 word on, it is the EPC of the tag. EPC memory is available to read and write.

TID memory contains information such as UID etc. provided by the tag vendor. TID memory is available to read and not to write.

User memory is a space for tag user. User memory is available to read and write. User memory is optional.

In G2 tag operation, the word unit is used frequently. 1 word equals 2 bytes.

Some G2 tag operation need password and if the password is not set, please use all 0s instead.

B. 18000-6B TAG (6B TAG)

6B tag has one integrated memory space. The lowest 8 bytes is for tag's UID which can not be altered. Other bytes are for user usage and available to read, write and lock. If the byte locked, it should not be able to alter or unlock.

8. DETAILED DESCRIPTION OF OPERATION COMMANDS

8.1 OVERVIEW

The reader supports three kinds of commands: protocol related command, reader command and tag customized command.

If the reader gets an unrecognized command such as non-exist command or CRC error command, it will return following response data block:

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

If the reader gets a command with incorrect length, it will return the following response data block:

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

The reader will not respond to next two kinds of commands:

1. A command with incorrect address. That means the Adr byte in the command is neither equal to the reader's address nor to the broadcast address 0xFF;

2. An un-integrated command. That means the length indicated by the Len byte is greater than the actual received command length.

8.2 EPC C1G2 COMMAND

8.2.1 Inventory

The command is used to inventory tags in the field and get their EPC values. The user may need to use this function to get tags' EPC before any further operation to the tags. This function has also been enhanced to get the tag's TID directly for firmware version V1.20 and above.

This command is executed within predefined InventoryScanTime limit. The default InventoryScanTime is 1 second (10*100ms). The allowable value range is from 3*100ms to 255*100ms.

Command:

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

Data as follows:

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

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

Parameter explanation:

QValue: One byte, Q value,

Bit7: Statistical data packets flag;

0: not send the flag packets when query finished.

1: send the flag packets when query finished.

Bit6-Bit0: range is 0-15. Q value setting should be field tag quantity is approximately equal to 2 Q.
other value return parameter error.

Session: One byte, Session value,

0x00:S0;

0x01:S1;

0x02:S2;

0x03:S3;

0xFF: Auto config session. (Only EPC query effective).

other value return parameter error.

MaskMem: One byte. It specifies the target memory when applying inventory mask with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address of the mask pattern data. The value ranges from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern data. The value ranges from 0 to 255.

MaskData: Mask pattern data. The byte length of the MaskData is MaskLen/8. If MaskLen is not 8bits integer times, the length of MaskData should be $\text{int}[\text{MaskLen}/8]+1$ with 0 patching in the low significant location.

AdrTID: One byte. It specifies the start word address in TID memory when doing the TID-inventory.

LenTID: One byte. It specifies the number of words when doing the TID-inventory. The range is 0~ 15.

Target(Optional parameters):One byte,

0x00: Target value is A;
 0x01: Target value is B;
 Other values are reserved.

Ant(Optional parameters):One byte,antenna number this time to query.

0x80: antenna 1;
 0x81: antenna 2;
 0x82: antenna 3;
 0x83: antenna 4;
 0x84: antenna 5;
 0x85: antenna 6;
 0x86: antenna 7;
 0x87: antenna 8;
 0x88: antenna 9;
 0x89: antenna 10;
 0x8A: antenna 11;
 0x8B: antenna 12.

ScanTime(Optional parameters): One byte,reader will set max scan time to ScanTime*100ms.

Remark:

- ① *The MaskMem , MaskAdr, MaskLen and MaskData can be all vacant. That means tags will be inventoried without mask pattern. The AdrTID and LenTID can be vacant for EPC inventory. If AdrTID and LenTID exist, it means the current inventory operation is a TID-inventory and the inventory procedure will get the tags' TID directly instead of their EPC.*
- ② *The Target, Ant, ScanTime is optional parameters,these parameters must be used to together.*

Response:

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

Parameter explanation:

Status:

Value	Comment
0x01	Tag inventory operation is completely finished in defined InventoryScanTime and all tags' EPC data are returned.
0x02	Tag inventory operation is not completely finished when defined InventoryScanTime overflows and some tags' EPC data are returned.
0x03	The number of the inventoried tags is too much and not all tags' EPC data can be returned within this single response data block. Other response data blocks are followed.
0x04	Tag inventory operation is finished since the number of the inventoried tags reaches the reader process limit. All inventoried tags' EPC data are returned.
0x26	If Qvalue's bit8=1.when inventory finished,return this code, it's rate packet

Ant: It describes from which antenna the tag EPC is collected. For example, **Ant** = 0000 0100b means ANT3 and **Ant** = 0000 1000b means ANT4, etc.

Num: The number of tag detected.

EPC/TID ID: Inventoried tag's EPC/TID data. **EPC-1** is the first tag's **EPC/TID length** plus **EPC/TID** data plus **RSSI** and so on. The most significant word of EPC/TID is transmitted first and the most significant byte of a word is also transmitted first. **EPC/TID length** is one byte.

8.2.2 Read Data

The command is used to read part or all of a Tag's Password, EPC, TID, or User memory.

Command:

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

Data as follows:

Data[]									
ENum	EPC	Mem	WordPtr	Num	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	Variable	0xXX	0xXX	0xXX	4Byte	0xXX	2Bytes	0xXX	Variable

Parameter explanation:

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by ENum with most significant word first and most significant byte in a word first.

Mem: One byte. It specifies the target memory of the operation with 0x00 for Password memory, 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

WordPtr: One byte. It specifies the starting word address for the operation with 0x00 for the first word, 0x01 for the second word and so on.

Num: One byte. It specifies the number of words to read. The value range is from 1 to 120. Other value will incur a parameters error.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password if operation needed or set Pwd all 0s.

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of MaskData is MaskLen/8. If MaskLen can not be divided exactly by 8, the length of MaskData is int(MaskLen/8)+1 with 0 padding in the least significant bit of last byte of MaskData.

*Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** are vacant.*

Response:

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

Parameter explanation:

Word1, Word2.....: read out data in word units with most significant byte of a word first. Word1 is the content of the start-address-defined word in target memory and so on.

8.2.3 Write Data

The command is used to write several words into a Tag's Reserved, EPC, TID or User memory.

Command:

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

Data as follows:

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

Parameter explanation:

WNum: One byte. It specifies the number of words to be written into a tag. The value can not be 0 and should be equal to the word length of **Wdt**.

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by ENum with most significant word first and most significant byte in a word first.

Mem: One byte. It specifies the target memory of the operation with 0x00 for Password memory, 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

WordPtr: One byte. It specifies the starting word address for the operation with 0x00 for the first word, 0x01 for the second word and so on.

Wdt: the data to be written into a tag. The word length of Wdt should be equal to WNum and the data should be arranged as most significant word first and most significant byte in a word first.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password if operation needed or set Pwd all 0s.

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of MaskData is MaskLen/8. If MaskLen can not be divided exactly by 8, the length of MaskData is $\text{int}(\text{MaskLen}/8)+1$ with 0 padding in the least significant bit of last byte of MaskData.

Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the MaskMem, MaskAdr, MaskLen and MaskData are vacant.

Response:

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

8.2.4 Write EPC

The command is used to write EPC code in a Tag's EPC memory. When using this command to write the EPC code, please make sure that there is only one tag in the effective field since this command utilize no mask pattern for selecting one tag to operate and just take one tag in the field to operate instead.

Command:

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

Parameter explanation:

ENum: one byte. It specifies the tag's EPC length in word unit and the value range is from 0 to 15.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password if operation needed or set Pwd all 0s.

WEPC: The EPC code to be written. The word length of EPC should be equal to **ENum** and the data should be arranged as most significant word first and most significant byte in a word first.

Response:

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

8.2.5 Kill Tag

The command is used to kill tag. After killed, a tag will not answer any request.

Command:

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

Data as follows:

Data[]						
ENum	EPC	Killpwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	Variable	4Bytes	0xXX	2Bytes	0xXX	Variable

Parameter explanation:

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by ENum with most significant word first and most significant byte in a word first.

Killpwd: Four bytes kill password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password if operation needed and all 0s kill password is not allowed..

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of MaskData is MaskLen/8. If MaskLen can not be divided exactly by 8, the length of MaskData is int(MaskLen/8)+1 with 0 padding in the least significant bit of last byte of MaskData.

*Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** are vacant.*

Response:

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

8.2.6 Lock

The command is used to set protection strategy of Reserved, EPC, TID and User memory.

Reserved memory can be set as unprotected readable/writable, permanent readable/writable, password-protected readable/writable and permanent unreadable/unwritable;

EPC and User memory can be set as unprotected writable, permanent writable, password-protected writable and permanent writable. These two memories are always readable;

TID memory is always readable and unwritable.

If the Reserved memory is set to permanent readable/writable or permanent unreadable/unwritable, it can not be set to other state any more.

If EPC or TID memory is set to permanent writable or permanent unwritable, it can not be changed to other state any more.

If a memory is set to password protected state, any access to this memory should be carried out with a correct password.

Command:

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

Data as follows:

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

Parameter explanation:

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by ENum with most significant word first and most

significant byte in a word first.

Select: One byte. It specifies the operation target as follows:

- 0x00: Kill Password section. Kill Password section is the first half of the Reserved memory;
 - 0x01: Access Password section. Access Password section is the second half of the Reserved memory;
 - 0x02: EPC memory;
 - 0x03: TID memory;
 - 0x04: User memory.
- Other values are reserved.

SetProtect: one byte. It specifies the protection strategy of the **Select** defined target memory.

When Select is 0x00 or 0x01, **SetProtect** has the following meaning::

- 0x00: unprotected readable/writable;.
- 0x01: permanent readable/writable;
- 0x02: password-protected readable/writable;
- 0x03: permanent unreadable/unwritable.

When Select is 0x02, 0x03 or 0x04, **SetProtect** has the following meaning:

- 0x00: unprotected writable;.
- 0x01: permanent writable;
- 0x02: password-protected writable;
- 0x03: permanent unwritable.

Other values are reserved.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password.

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of MaskData is MaskLen/8. If MaskLen can not be divided exactly by 8, the length of MaskData is int(MaskLen/8)+1 with 0 padding in the least significant bit of last byte of MaskData.

*Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** are vacant.*

Response:

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

8.2.7 BlockErase

The command is used to erase multiple words in a Tag's Reserved, EPC, TID or User memory. **BlockErase** is an advance but non-mandatory feature of EPC C1G2 tag and please check the tag's datasheet before using it.

Command:

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

Data as follows:

Data[]									
ENum	EPC	Mem	WordPtr	Num	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	Variable	0xXX	0xXX	0xXX	4Byte	0xXX	2Bytes	0xXX	Variable

Parameter explanation:

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by ENum with most significant word first and most significant byte in a word first.

Mem: One byte. It specifies the target memory of the operation with 0x00 for Password memory, 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

WordPtr: One byte. It specifies the starting word address for the operation with 0x00 for the first word, 0x01 for the second word and so on. If the target memory is the EPC memory, the **WordPtr** should not be 0.

Num: One byte. It specifies the number of words to erase.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password if operation needed or set Pwd all 0s.

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of MaskData is MaskLen/8. If MaskLen can not be divided exactly by 8, the length of MaskData is int(MaskLen/8)+1 with 0 padding in the least significant bit of last byte of MaskData.

*Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** are vacant.*

Response:

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

8.2.8 Set Privacy with Mask Pattern

The command is used to set designated tag into privacy state. When a tag is in privacy state, it will ignore common command such as read, write and query etc. This is an advanced feature which is available for NXP's UCODE EPC G2X tag.

Command:

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

Data as follows:

Data[]						
ENum	EPC	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	Variable	4Bytes	0xXX	2Bytes	0xXX	Variable

Parameter explanation:

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by ENum with most significant word first and most significant byte in a word first.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password and the password should not be all 0s.

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of MaskData is MaskLen/8. If MaskLen can not be divided exactly by 8, the length of MaskData is int(MaskLen/8)+1 with 0 padding in the least significant bit of last byte of MaskData.

Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** are vacant.

Response:

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

8.2.9 Set Privacy without Mask Pattern

The command is used to set a tag in the field into privacy state. When a tag is in privacy state, it will ignore common command such as read, write and query etc. This is an advanced feature which is available for NXP's UCODE EPC G2X tag.

Command:

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

Parameter explanation:

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password and the password should not be all 0s.

Response:

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

8.2.10 Reset Privacy

The command is used to release a tag from privacy state into normal state. This is an advanced feature which is available for NXP's UCODE EPC G2X tag.

Command:

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

Parameter explanation:

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password and the password should not be all 0s.

Response:

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

8.2.11 Check Privacy

The command is used to check whether a tag in the field is in privacy state or not.

Command:

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

Response:

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

Parameter explanation:

ReadPro	Connect
0x00	Tag is not in privacy state.
0x01	Tag is in privacy state.

8.2.12 EAS Configure

The function is used to set or reset the EAS bit of a designated tag. This is an advanced feature which is available for NXP's UCODE EPC G2X tag.

Command:

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

Data as follows:

Data[]							
ENum	EPC	Pwd	EAS	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	Variable	4Bytes	0xXX	0xXX	2Bytes	0xXX	Variable

Parameter explanation:

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by ENum with most significant word first and most significant byte in a word first.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password and the password should not be all 0s.

EAS: One byte. Bit0=0 means reset EAS state, Bit0=1 means set EAS state. Bit1~Bit7 are reserved.

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of MaskData is MaskLen/8. If MaskLen can not be divided exactly by 8, the length of MaskData is int(MaskLen/8)+1 with 0 padding in the least significant bit of last byte of MaskData.

*Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** are vacant.*

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x0c	0x00	-	LSB	MSB

8.2.13 EAS Alarm

The function is used to check EAS bit's status of a tag in the field. This is an advanced feature which is available for NXP's UCODE EPC G2X tag.

Command:

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

Response:

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

If no tag's EAS bit is set, the response status will be "no tag in the field".

8.2.14 Block Write

The command is used to write multiple words in a Tag's Reserved, EPC, TID or User memory. **BlockWrite** is an advance but non-mandatory feature of EPC C1G2 tag and please check the tag's datasheet before using it.

Command:

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

Data as follows:

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

Parameter explanation:

WNum: One byte. It specifies the number of words to be written into a tag. The value can not be 0 and should be equal to the word length of **Wdt**.

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem** , **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by ENum with most significant word first and most significant byte in a word first.

Mem: One byte. It specifies the target memory of the operation with 0x00 for Password memory, 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

WordPtr: One byte. It specifies the starting word address for the operation with 0x00 for the first word, 0x01 for the second word and so on.

Wdt: the data to be written into a tag. The word length of Wdt should be equal to **WNum** and the data should be arranged as most significant word first and most significant byte in a word first.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password if operation needed or set Pwd all 0s.

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of MaskData is MaskLen/8. If MaskLen can not be divided exactly by 8, the length of MaskData is int(MaskLen/8)+1 with 0 padding in the least significant bit of last byte of MaskData.

*Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** are vacant.*

Response:

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

8.2.15 Read Monza4QT parameters

The command is used to read Monza4Qt work parameters.

Command:

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

Data as follows:

Data[]						
ENum	EPC	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	Variable	4Bytes	0xXX	2Bytes	0xXX	Variable

Parameter explanation:

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by ENum with most significant word first and most significant byte in a word first.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password if operation needed or set Pwd all 0s.

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of MaskData is MaskLen/8. If MaskLen can not be divided exactly by 8, the length of MaskData is int(MaskLen/8)+1 with 0 padding in the least significant bit of last byte of MaskData.

*Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** are vacant.*

Response:

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

Parameter explanation:

QTcontrol: Tag work parameters.

Bit0: mirror page.bit0=0:private;bit0=1:public.

Bit1: enabled distance protect or not.bit1=0:disabled;bit1=1:enabled.

8.2.16 Set Monza4QT parameters

The command is used to Set Monza4Qt work parameters.

Command:

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

Data as follows:

Data[]								
ENum	EPC	QTcontrol1	QTcontrol0	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	Variable	0xXX	0xXX	4Bytes	0xXX	2Bytes	0xXX	Variable

Parameter explanation:

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem** , **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by ENum with most significant word first and most significant byte in a word first.

QTcontrol1: reserved, 0x00;

QTcontrol0: Tag work parameters.

Bit0: mirror page.bit0=0:private;bit0=1:public.

Bit1: enabled distance protect or not.bit1=0:disabled;bit1=1:enabled.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password if operation needed or set Pwd all 0s.

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of MaskData is MaskLen/8. If MaskLen can not be divided exactly

by 8, the length of MaskData is $\text{int}(\text{MaskLen}/8)+1$ with 0 padding in the least significant bit of last byte of MaskData.

*Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** are vacant.*

Response:

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

Parameter explanation:

QTcontrol: Tag work parameters.

Bit0: mirror page.bit0=0:private;bit0=1:public.

Bit1: enabled distance protect or not.bit1=0:disabled;bit1=1:enabled.

8.2.17 Extended Read Data

The command is used to read part or all of a Tag's Password, EPC, TID, or User memory.

Command:

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

Data as follows:

Data[]									
ENum	EPC	Mem	WordPtr	Num	Pwd	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	Variable	0xXX	2Bytes	0xXX	4Byte	0xXX	2Bytes	0xXX	Variable

Parameter explanation:

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by ENum with most significant word first and most significant byte in a word first.

Mem: One byte. It specifies the target memory of the operation with 0x00 for Password memory, 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

WordPtr: Two byte. It specifies the starting word address for the operation with 0x0000 for the first word, 0x0001 for the second word and so on.

Num: One byte. It specifies the number of words to read. The value range is from 1 to 120. Other value will incur a parameters error.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of Pwd. Please remember to set Pwd the correct password if operation needed or set Pwd all 0s.

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of MaskData is MaskLen/8. If MaskLen can not be divided exactly by 8, the length of MaskData is int(MaskLen/8)+1 with 0 padding in the least significant bit of last byte of MaskData.

*Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** are vacant.*

Response:

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

Parameter explanation:

Word1, Word2.....: read out data in word units with most significant byte of a word first. Word1 is the content of the start-address-defined word in target memory and so on.

8.2.18 Extended Write Data

The command is used to write several words into a Tag's Reserved, EPC, TID or User memory.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x016	-	LSB	MSB

Data as follows:

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

Parameter explanation:

WNum: One byte. It specifies the number of words to be written into a tag. The value can not be 0 and should be equal to the word length of **Wdt**.

ENum: one byte length indicator in word unit (1 word = 2 bytes).

When ENum's value is in 0~15, it represents the **EPC** length. And in this case, the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** shall all be absent.

When **ENum** is 0xff, there shall be **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** and no **EPC** presented. The **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** construct a mask pattern for the operation.

Other value for **ENum** is prohibited and will incur a parameter error.

EPC: tag's whole EPC code. Its word length is defined by **ENum** with most significant word first and most significant byte in a word first.

Mem: One byte. It specifies the target memory of the operation with 0x00 for Password memory, 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

WordPtr: Two byte. It specifies the starting word address for the operation with 0x0000 for the first word, 0x0001 for the second word and so on.

Wdt: the data to be written into a tag. The word length of **Wdt** should be equal to **WNum** and the data should be arranged as most significant word first and most significant byte in a word first.

Pwd: Four bytes access password. The most significant byte (MSB) of access password is the MSB of **Pwd**. Please remember to set **Pwd** the correct password if operation needed or set **Pwd** all 0s.

MaskMem: One byte. It specifies the target memory when applying mask pattern with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address in target memory when applying mask pattern. The value range is from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern.

MaskData: mask pattern data. The length of **MaskData** is $\text{MaskLen}/8$. If **MaskLen** can not be divided exactly by 8, the length of **MaskData** is $\text{int}(\text{MaskLen}/8)+1$ with 0 padding in the least significant bit of last byte of **MaskData**.

*Notes: The operation actually uses the tag's whole EPC code as the mask pattern when the **MaskMem**, **MaskAdr**, **MaskLen** and **MaskData** are vacant.*

Response:

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

8.2.19 Mix Inventory

The command is used to inventory tags in the field, then get their EPC values. And read specified tag data at the same time.

This command is executed within predefined **InventoryScanTime** limit. The default **InventoryScanTime** is 1 second (20*100ms). The allowable value range is from 3*100ms to 255*100ms.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
0xXX	0xXX	0x019	-	LSB	MSB

Data as follows:

Data[]					
QValue	Session	MaskMem	MaskAdr	MaskLen	MaskData
0xXX	0xXX	0xXX	2Bytes	0xXX	Variable

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

Parameter explanation:

QValue: One byte, Q value,

Bit7: Statistical data packets flag;

0: not send the flag packets when query finished.

1: send the flag packets when query finished.

Bit6-Bit0: range is 0-15. Q value setting should be field tag quantity is approximately equal to 2 Q. other value return parameter error.

Session: One byte, Session value,

0x00:S0;

0x01:S1;

0x02:S2;

0x03:S3;

0xFF: Auto config session. (Only EPC query effective).

other value return parameter error.

MaskMem: One byte. It specifies the target memory when applying inventory mask with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address of the mask pattern data. The value ranges from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern data. The value ranges from 0 to 255.

MaskData: Mask pattern data. The byte length of the MaskData is MaskLen/8. If MaskLen is not 8bits integer times, the length of MaskData should be $\text{int}[\text{MaskLen}/8]+1$ with 0 patching in the low significant location.

ReadMem: One byte. It specifies the target memory of the operation with 0x00 for Password memory, 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

ReadAdr: Two byte. It specifies the start word of read data. 0x0000 is first word, 0x0001 is second word, and so on.

ReadLen: One byte. It specifies the length of word to read.the value ranges form 1 to 120.

Pwd: Four byte. It specifies the access-password of tag.

Target(Optional parameters):One byte,

0x00: Target value is A;

0x01: Target value is B;

Other values are reserved.

Ant(Optional parameters):One byte,antenna number this time to query.

0x80: antenna 1;

0x81: antenna 2;

0x82: antenna 3;

0x83: antenna 4.

ScanTime(Optional parameters): One byte,reader will set max scan time to ScanTime*100ms.

Remark:

- ① *The MaskMem , MaskAdr, MaskLen and MaskData can be all vacant. That means tags will be inventoried without mask pattern.*
- ② *The Target, Ant, ScanTime is optional parameters,these parameters must be used to together.*

(1).When Status is 0x26

Response:

Len	Adr	reCmd	Status	Data[]			CRC-16	
				Ant	ReadRate	TotalCount		
0x0C	0xXX	0x19	0x26	0xXX	2 Bytes	4 Bytes	LSB	MSB

Parameter explanation:

Ant: It describes from which antenna the tag EPC is collected. For example, **Ant** = 0000 0100b means ANT3 and **Ant** = 0000 1000b means ANT4, etc.

ReadRate: 2 Bytes, Tag identification rate (the number of successfully read tag/SEC). Don't distinguish whether read many times with same tag

TotalCount: Total count of read tag,Don't distinguish whether read many times with same tag.

(2).When Status is not 0x26

Response:

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

Parameter explanation:

Status:

Value	Comment
0x01	Tag inventory operation is completely finished in defined InventoryScanTime and all tags' EPC data are returned.
0x02	Tag inventory operation is not completely finished when defined InventoryScanTime overflows and some tags' EPC data are returned.
0x03	The number of the inventoried tags is too much and not all tags' EPC data can be returned within this single response data block. Other response data blocks are followed.
0x04	Tag inventory operation is finished since the number of the inventoried tags reaches the reader process limit. All inventoried tags' EPC data are returned.
0xF8	If Qvalue's bit8=1.when inventory finished,return this code, it's rate packet

Ant: It describes from which antenna the tag EPC is collected. For example, **Ant** = 0000 0100b means ANT3 and **Ant** = 0000 1000b means ANT4, etc.

Num: The number of tag detected.

Data Packet: Inventoried tag's data. **Packet-1** is the first packet data, **Packet-2** is the second packet data, and so on. Each packet as follows:

PacketParam	Len	Data	RSSI
0xXX	0xXX	Variable	0xXX

Packet explanation:

PacketParam: Packet parameters, 1 byte.

Bit7: packet types.

0 - this packet with information for the tag EPC;

1 - this packet with information for the tag Data (the data of tag and the last read EPC is same one).

Bit6 ~ Bit0: packet sequence number. Packet number ranges from 0 to 127. Each mix-inventory, packet number will start from 0, per upload a packet, the packet sequence number will automatically add 1, when the packet number more than 127, will start from 0

Len: the length of Data.

Data: Tag's EPC or Data.

RSSI: the signal strength when read the tag.

8.3 18000-6B COMMAND

8.4 READER-DEFINED COMMAND

8.4.1 Get Reader Information

The command is used to get the reader's information such as firmware version, reader type code, supporting protocol, RF power, work frequency band, InventoryScanTime etc.

Command:

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

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x10	0xXX	0x21	0x00	Version, Type, Tr_Type, dmaxfre, dminfre, Power, Scntm, Ant1, BeepEn, Ant2, CheckAnt	LSB	MSB

Parameter explanation:

Parameter	Length(Byte)	Connect
Version	2	The first byte specifies the main version and the second byte specifies the subversion.
Type	1	The reader type code with 0x8C-UHFREADER288A12,
Tr_Type	1	It specifies the protocol the reader supports. For UHFREADER288, two bits (bit1&bit0) are set to 1 with bit1=1 for ISO18000-6C and bit0=1 for ISO18000-6B.
dmaxfre	1	Bit7~Bit6 is used to indicate frequency band and Bit5~Bit0 is used to specify the maximum frequency point. As to the frequency band definition, please refer to the following table. Please also refer to 8.4.2.
dminfre	1	Bit7~Bit6 is used to indicate frequency band and Bit5~Bit0 is used to specify the minimum frequency point. As to the frequency band definition, please refer to the following table. Please also refer to 8.4.2.
Power	1	It specifies the reader's RF output power. The value range is 0 to 30 with 30 for 30dbm.
Scntm	1	It specifies the InventoryScanTime. Please also refer to 8.2.1
Ant1	1	Antenna 1-8 config information
BeepEn	1	It specifies the beep is enabled or not.
Ant2	1	Antenna 9-12 config information
CheckAnt	1	It specifies the Antanna checked before query,please also refer to 8.4.13

Frequency Band Table:

dmaxfre(Bit7)	dmaxfre(Bit6)	dminfre(Bit7)	dminfre(Bit6)	Region Frequency Band
0	0	0	0	RFU
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	RFU
0	1	1	0	Ukraine band
0	1	1	1	Peru band
1	0	0	0	Chinese band1
...

1	1	1	1	RFU
---	---	---	---	-----

8.4.2 Set Region

The command is used to set the reader's working frequency band and the maximum and minimum frequency point in the band.

Command:

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

Parameter explanation:

MaxFre: one byte. Bit7~Bit6 is used to indicate frequency band and Bit5~Bit0 is used to specify the maximum frequency point. As to the frequency band definition, please refer to the following table.

MinFre: one byte. Bit7~Bit6 is used to indicate frequency band and Bit5~Bit0 is used to specify the minimum frequency point. As to the frequency band definition, please refer to the following table.

Please note that **MaxFre** should be greater than **MinFre**.

Frequency Band Table:

dmaxfre(Bit7)	dmaxfre(Bit6)	dminfre(Bit7)	dminfre(Bit6)	Region Frequency Band
0	0	0	0	RFU
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	RFU
0	1	1	0	Ukraine band
0	1	1	1	Peru band
1	0	0	0	Chinese band1
...
1	1	1	1	RFU

Response:

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

Various Region Frequency Band Calculation:

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

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

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

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

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

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

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

8.4.3 Set Address

This command is used to set the reader's address.

Command:

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

Parameter explanation:

Address: one byte. It specifies the reader's address and the value range is 0 to 254. 255 is the broadcasting address.

Response:

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

Notes: The Adr in response is the old address and not the new address.

8.4.4 Set InventoryScanTime

The command is used to set the InventoryScanTime.

Command:

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

Parameter explanation:

Scantime: one byte. It specifies the InventoryScan Time. Please also refer to 8.2.1.

Response:

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

8.4.5 Set Baud Rate

The command is used to change the communication baud rate.

Command:

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

Parameter explanation:

BaudRate: one byte. It specifies the communication baud rate as following table.

BaudRate	Bps
0	9600bps
1	19200 bps
2	38400 bps
5	57600 bps (default)
6	115200 bps

Response:

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

Notes: The response is still using the old baud rate. After this, later communication will use new baud rate.

8.4.6 Set RF Power

The command is used to adjust the RF output power of the reader.

Command:

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

Parameter explanation:

Pwr: one byte. It specifies the RF output power. The value range is from 0 to 30 with 30 for around 30dbm.

Response:

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

8.4.7 BuzzerAndLED Control

The command is used to control Buzzer and LED action.

Command:

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

Parameter explanation:

ActiveT: one byte. It specifies the time duration that the LED and the buzzer are on. The time duration is **ActiveT***50ms and the default **ActiveT** value is 0.

SilentT: one byte. It specifies the time duration that the LED and the buzzer are off. The time duration is **SilentT** *50ms and the default **SilentT** value is 0.

Times: one byte. It specifies the repeat times that the LED and buzzer go on and off. The default **Times** value is 0.

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x33	0x00	-	LSB	MSB

8.4.8 Set Work Mode

The command is used to set the work mode of the reader. The reader supports 4 work modes that are answer mode, auto-running mode, low-level-active trigger mode and high-level-active trigger mode. When the reader is in auto-running mode, it can only receive and executive reader-defined commands.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
			Reader_mode		
0x05	0xXX	0x35	0xXX	LSB	MSB

Parameter explanation:

Reader_mode:

Bit2	Bit1	Bit0	Work Mode
0	0	0	Answer Mode
0	0	1	Auto-running Mode
0	1	0	Low-level-active Trigger Mode
0	1	1	High-level-active Trigger Mode

Bit3~Bit7: reserved.

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x35	0x00	-	LSB	MSB

Output Data Format in auto-running mode and trigger mode:

1. Tag inventory related operation:

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

Data as follows:

Data[]				
DiscTime	LastTime	Count	Ant	EPC/TID ID
4Bytes	4Bytes	2Bytes	0xXX	Variable

Parameter explanation:

DiscTime: 4 bytes. It specifies the time stamp when the tag is firstly detected.

DiscTime					
Year	Month	Date	Hour	Minute	Second
Bit31~Bit26	Bit25~Bit22	Bit21~Bit17	Bit16~Bit12	Bit11~Bit6	Bit5~Bit0

Year: 6bits. It ranges from 0 to 63 for year 2000 to 2063.

Month: 4bits. It ranges from 1 to 12 for Jan. to Dec.

Date: 5bits. It ranges from 1 to 31 for day 1 to day 31.

Hour: 5bits. It ranges from 0 to 23.

Minute: 6bits. It ranges from 0 to 59.

Second: 6bits. It ranges from 0 to 59.

LastTime: 4 bytes. It specifies the time stamp when the tag is latest detected.

LastTime					
Year	Month	Date	Hour	Minute	Second
Bit31~Bit26	Bit25~Bit22	Bit21~Bit17	Bit16~Bit12	Bit11~Bit6	Bit5~Bit0

Year: 6bits. It ranges from 0 to 63 for year 2000 to 2063.

Month: 4bits. It ranges from 1 to 12 for Jan. to Dec.

Date: 5bits. It ranges from 1 to 31 for day 1 to day 31.

Hour: 5bits. It ranges from 0 to 23.

Minute: 6bits. It ranges from 0 to 59.

Second: 6bits. It ranges from 0 to 59.

Count: 2 bytes. It specifies the times the tag has been detected during time interval from **DiscTime** to **LastTime**.

Ant: two byte. It specifies which antenna or antennae have detected the tag. It uses BCD coding. For example, **Ant** as **0000 0110 0000 1000**b means antenna 4,10 and antenna 11 have detected the tag.

EPC/TID ID: Detected tag's EPC/TID data. The first byte is EPC/TID length and the following bytes are EPC/TID data with most significant word of EPC/TID transmitted first and the most significant byte of a word also transmitted first.

2. Check Tag's EAS operation

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

When EAS alarm is activated, the built-in relay will pick up for about 3s.

Note: When EAS checking is arranged before a tag inventory operation, the reader will not give out above response message even an EAS alarm activated but the relay can still act and pick up for about 3s.

8.4.9 Get System Parameters

The command is used to get the current set parameters of the reader for tag operation and work mode configuration.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x36	-	LSB	MSB

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x32	0xXX	0x36	0x00	Reader_mode, Accuracy, RepCondition, RepPauseTime, ReadPauseTim,	LSB	MSB

				TagProtocol, MaskMem, MaskAdr, MaskLen, MaskData, TriggerTime, AdrTID, LenTID		
--	--	--	--	---	--	--

Parameter explanation:

Reader_mode: one byte. It specifies the current work mode. Please also refer to 8.4.8.

Accuracy: one byte. It specifies the EAS checking sensitivity and accuracy. Please also refer to 8.4.10.

RepCondition, RepPauseTime: one byte each. They specify the response style parameters when the reader in auto-running and trigger mode. Please also refer to 8.4.12.

ReadPauseTim: one byte. It specifies the inventory interval when the reader in auto-running and trigger mode. Please also refer to 8.4.13.

TagProtocol: one byte. It specifies the tag type when the reader runs a tag related operation command. Please also refer to 8.4.14.

MaskMem, MaskAdr, MaskLen, MaskData: Mask pattern parameters. Please also refer to 8.4.11.

TriggerTime: one byte. It specifies the trigger delay time. Please also refer to 8.4.25.

AdrTID, LenTID: one byte each. They specify TID-inventory parameters. Please also refer to 8.4.26.

Note: **TriggerTime, AdrTID, LenTID** are available for readers with firmware version 1.20 or above.

8.4.10 Set EAS Sensitivity

The command is used to set the EAS checking sensitivity and accuracy. It also defines the relay action when an EAS alarm is activated.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
			Accuracy		
0x05	0xXX	0x37	0xXX	LSB	MSB

Parameter explanation:

Accuracy: one bit (bit7). It specifies the relay action when an EAS alarm is activated.

=1 pick-up for about 3s.

=0 no action.

bit6~bit0: EAS alarm accuracy definition. The value range is from 0 to 8. Smaller value means higher sensitivity but lower accuracy. The default value is 8.

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x37	0x00	-	LSB	MSB

8.4.11 Set Mask

The command is used to set the mask pattern for the G2 tag related operation when the reader is in auto-running and trigger mode.

Command:

Len	Adr	Cmd	Data[]				CRC-16	
			MaskMem	MaskAdr	MaskLen	MaskData		
0xXX	0xXX	0x3b	0xXX	2Bytes	0xXX	Variable	LSB	MSB

Parameter explanation:

MaskMem: One byte. It specifies the target memory when applying inventory mask with 0x01 for EPC memory, 0x02 for TID memory and 0x03 for User memory. Other values are reserved.

MaskAdr: Two bytes. It specifies the start bit address of the mask pattern data. The value ranges from 0 to 16383.

MaskLen: One byte. It specifies the bit length of the mask pattern data. The value ranges from 0 to 255.

MaskData: Mask pattern data. The byte length of the MaskData is MaskLen/8. If MaskLen is not 8bits integer times, the length of MaskData should be int[MaskLen/8]+1 with 0 patching in the low significant location.

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x3b	0x00	-	LSB	MSB

8.4.12 Set Response Style of Auto-running Mode

The command is used to set the response style when the reader is in auto-running and trigger mode.

Command:

Len	Adr	Cmd	Data[]		CRC-16	
			RepCondition	RepPauseTime		
0x06	0xXX	0x3c	0xXX	0xXX	LSB	MSB

Parameter explanation:

RepCondition: one byte. It specifies the response style:

- 0x00: Command notification. The reader will collect all the tags' information and store them into its inner buffer. The reader will send back the buffer data when it receives the **Get Tag Buffer Info** command from the host.
- 0x01: Timer notification. The reader will give out tag's information it collects during a pre-defined period **RepPauseTime**.
- 0x02: Add-in notification. The reader will give out a message when it gets a new tag's information.
- 0x03: Delete notification. The reader will give out a message data of a tag that is going to be deleted from inner buffer when the buffer is full and the reader need to empty room for new-coming tag data.

0x04: Change notification. The reader will give out a message when the reader gets a new tag data or deletes a tag data.

Other values are reserved.

RepPauseTime: one byte. It specifies the time period when the reader runs in a Timer notification style.

Notes: The reader's inner buffer can hold up to 97 tags' data and it adopts a FIFO strategy when the 98th and later tag data comes.

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x3c	0x00	-	LSB	MSB

8.4.13 Set Inventory Interval

The command is used to set the time interval when the reader does consecutive inventory operations in auto-running and trigger mode.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
			ReadPauseTim		
0x05	0xXX	0x3d	0xXX	LSB	MSB

Parameter explanation:

RepPauseTime: one byte. It specifies the time interval of consecutive inventory operations.

ReadPauseTim	Time interval
0x00	10ms
0x01	20ms
0x02	30ms
0x03	50ms
0x04	100ms
Other value	Reserved

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x3d	0x00	-	LSB	MSB

8.4.14 Select Tag Type

The command is used to define the tag type when the reader runs a tag related operation command in

auto-running mode and trigger mode.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
			TagProtocol		
0x05	0xXX	0x3e	0xXX	LSB	MSB

Parameter explanation:

TagProtocol: one byte. It specifies the tag type and operation.

TagProtocol	Result
0x00	G2 tag EPC inventory
0x01	G2 tag EPC inventory and EAS checking
0x02	G2 tag EAS checking
0x10	G2 tag TID-inventory
0x11	G2 tag TID-inventory and EAS checking
0x80	ISO18000-6B tag inventory
Other value	Reserved

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x3e	0x00	-	LSB	MSB

8.4.15 Set Antenna Multiplexing

The command is used to set the antenna multiplexing scenario.

Command:

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

Parameter explanation:

Ant: one byte. It specifies the antenna multiplexing scenario. The reader has 4 antenna ports and each antenna port can be individually enable or disable. At least one antenna port should be enabled.

Bit0: =0: ANT1 disabled.

=1: ANT1 enabled.

Bit1: =0: ANT2 disabled.

=1: ANT2 enabled.

Bit2: =0: ANT3 disabled.

=1: ANT3 enabled.

Bit3: =0: ANT4 disabled.

=1: ANT4 enabled.

Bit4~Bit7: Reserved.

Response:

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

8.4.16 Set Beep Notification

The command is used to enable/disable beep notification.

Command:

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

Parameter explanation:

BeepEn: Bit0. It specifies the status of beep notification.

=0: Beep notification disabled.

=1: Beep notification enabled. The inner buzzer will beep once whenever a tag related operation succeeds.

Bit1~Bit7: Reserved.

Response:

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

8.4.17 Set Real-time Clock

The command is used to set the time of the inner real-time clock.

Command:

Len	Adr	Cmd	Data[]						CRC-16		
			Year	Month	Date	Hour	Minute	Second			
0x0a	0xXX	0x41	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	LSB	MSB

Parameter explanation:

Year: one byte. It ranges from 0 to 63 for year 2000 to 2063.

Month: one byte. It ranges from 1 to 12 for Jan. to Dec.

Date: one byte. It ranges from 1 to 31 for day 1 to day 31.

Hour: one byte. It ranges from 0 to 23.

Minute: one byte. It ranges from 0 to 59.

Second: one byte. It ranges from 0 to 59.

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x41	0x00	-	LSB	MSB

Notes: The time to be set shall not be earlier than any time stamp of the tags stored in the reader's inner buffer.

8.4.18 Get Time

The command is used to get the current time of the reader's inner real-time clock.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x42	-	LSB	MSB

Response:

Len	Adr	reCmd	Status	Data[]						CRC-16	
				Year	Month	Date	Hour	Minute	Second	LSB	MSB
0x0b	0xXX	0x42	0x00	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX	LSB	MSB

Parameter explanation:

Year: one byte. It ranges from 0 to 63 for year 2000 to 2063.

Month: one byte. It ranges from 1 to 12 for Jan. to Dec.

Date: one byte. It ranges from 1 to 31 for day 1 to day 31.

Hour: one byte. It ranges from 0 to 23.

Minute: one byte. It ranges from 0 to 59.

Second: one byte. It ranges from 0 to 59.

8.4.19 Get Tag Buffer Info

The command is used to fetch the tag information stored in the reader's inner buffer.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x43	-	LSB	MSB

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0xXX	0xXX	0x43	0xXX	-	LSB	MSB

Parameter explanation:

Status: one byte. If it is 0x00, it means no data is available. If it is 0x03, it means the data is presented in the response block but not all data has been fetched. Other data is available in following response blocks.

Data[]: the structure of **Data[]** is:

Data[]				
DiscTime	LastTime	Count	Ant	EPC-ID
4Bytes	4Bytes	2Bytes	0xXX	Variable

DiscTime: 4 bytes. It specifies the time stamp when the tag is firstly detected.

DiscTime					
Year	Month	Date	Hour	Minute	Second
Bit31~Bit26	Bit25~Bit22	Bit21~Bit17	Bit16~Bit12	Bit11~Bit6	Bit5~Bit0

Year: 6bits. It ranges from 0 to 63 for year 2000 to 2063.

Month: 4bits. It ranges from 1 to 12 for Jan. to Dec.

Date: 5bits. It ranges from 1 to 31 for day 1 to day 31.

Hour: 5bits. It ranges from 0 to 23.

Minute: 6bits. It ranges from 0 to 59.

Second: 6bits. It ranges from 0 to 59.

LastTime: 4 bytes. It specifies the time stamp when the tag is latest detected.

LastTime					
Year	Month	Date	Hour	Minute	Second
Bit31~Bit26	Bit25~Bit22	Bit21~Bit17	Bit16~Bit12	Bit11~Bit6	Bit5~Bit0

Year: 6bits. It ranges from 0 to 63 for year 2000 to 2063.

Month: 4bits. It ranges from 1 to 12 for Jan. to Dec.

Date: 5bits. It ranges from 1 to 31 for day 1 to day 31.

Hour: 5bits. It ranges from 0 to 23.

Minute: 6bits. It ranges from 0 to 59.

Second: 6bits. It ranges from 0 to 59.

Count: 2 bytes. It specifies the times the tag has been detected during time interval from **DiscTime** to **LastTime**.

Ant: one byte. It specifies which antenna or antennae have detected the tag. It uses BCD coding. For example, **Ant** as 00000101b means antenna 1 and antenna 3 have detected the tag.

EPC/TID ID: Detected tag's EPC/TID data. The first byte is EPC/TID length and the following bytes are EPC/TID data with most significant word of EPC/TID transmitted first and the most significant byte of a word also transmitted first.

8.4.20 Clear Tag Buffer

The command is used to clear the reader's inner buffer.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x44	-	LSB	MSB

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x44	0x00	-	LSB	MSB

8.4.21 Set Relay

The command is used to control the built-in relay.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
			RelayTime		
0x05	0xXX	0x45	0xXX	LSB	MSB

Parameter explanation:

RelayTime: one byte. It specifies the relay pick-up time. The value is 1~255 for (1~255)*50ms pick-up time.

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x45	0x00	-	LSB	MSB

8.4.22 Set GPIO

The command is used to set the GPIO output TTL level. The default is TTL high level.

Command:

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

Parameter explanation:

OutputPin: one byte. It specifies the 4 output GPIOs' (OUT1~OUT4) level. Bit4~Bit7 correspond to OUT1~OUT4.

Response:

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

8.4.23 Get GPIO Status

The command is used to get the current 4 GPIOs output status.

Command:

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

Response:

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

Parameter explanation:

OutputPin: one byte. It specifies the 4 output GPIOs' (OUT1~OUT4) level. Bit4~Bit7 correspond to OUT1~OUT4.

8.4.24 Set Trigger Delay Time

The command is used to set the trigger delay time. When the reader in trigger mode, it will keep working for a pre-defined time period even if the trigger source has been removed.

Command:

Len	Adr	Cmd	Data[]		CRC-16	
			TriggerTime			
0x05	0xXX	0x49	0xXX		LSB	MSB

Parameter explanation:

TriggerTime: one byte. It specifies the trigger delay time. The value range is 0~255 for (0~255)*1s trigger delay time.

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x49	0x00	-	LSB	MSB

Notes: The command is only available for readers with firmware version 1.20 or above.

8.4.25 Set TID-Inventory Parameters

The command is used to set TID-Inventory parameters. Please also refer to 8.2.1.

Command:

Len	Adr	Cmd	Data[]		CRC-16	
			AdrTID	LenTID		
0x06	0xXX	0x4a	0xXX	0xXX	LSB	MSB

Parameter explanation:

AdrTID: One byte. It specifies the start word address in TID memory when doing the TID-inventory.

LenTID: One byte. It specifies the number of words when doing the TID-inventory. The range is 0~ 15.

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x4a	0x00	-	LSB	MSB

Notes: The command is only available for readers with firmware version 1.20 or above.

8.4.26 Get Reader Serial Number

The command is used to get reader's serial number.

Command

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

Parameter explanation:

None.

Response:

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

SerialNo:reader's serial number,4bytes.*Notes: The command is only available for readers with firmware version 1.20 or above.***8.4.27 Change AT Mode**

The command is used to change network module to AT mode or exit..

Command

Len	Adr	Cmd	Data[]	CRC-16	
			ATMode		
0x05	0xXX	0x4d	0xXX	LSB	MSB

Parameter explanation:**ATMode:** One byte.network module mode select.

0x01,AT mode;

0x00,exit AT mode.

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x4d	0x00	-	LSB	MSB

*Notes: The command is only available for readers with firmware version 1.20 or above.***8.4.28 AT Transparent Command**

The command is used to send transparent cmd to network module.

Command

Len	Adr	Cmd	Data[]		CRC-16	
			TimeOut	ATCmd		
0x05+N	0xXX	0x4e	0xXX	N bytes	LSB	MSB

Parameter explanation:**TimeOut:** One byte.the cmd operation timeout(1~255)*100ms.**ATCmd:** Nbytes.the datas of user send to network module.**Response:**

Len	Adr	reCmd	Status	Data[]	CRC-16	
				ATResp		
0x05+N	0xXX	0x4e	0x00	N bytes	LSB	MSB

ATResp: Nbytes.the datas of network module response.

Notes: The command is only available for readers with firmware version 1.20 or above.

8.4.29 Tag custom function

The command is used to set enable some type tag custom work mechanism, to achieve a specific function.

Command

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

Parameter explanation:

InlayType: One byte.range is 0~254

The default value is 0, do not specify a tag type.

Value 1 enable the Monza4QT Peek function of tag (tag status temporary from public to private), it will affect the tag reading data, writing data, the piece of writing, write protect words and write the EPC number etc

Response:

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

InlayType: tag type.0~255..

Notes: The command is only available for readers with firmware version 1.20 or above.

8.4.30 Set antenna checked

The command is used to set the antenna checked before query tag.

Command:

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

Parameter explanation:

CheckAnt: one byte.check antenna flag.

0x00:Disabled;

0x01:Enabled;

Response:

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

8.4.31 Set Write Power

The command is used to set power parameter when do write operation

Command:

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

Parameter explanation:

WritePower: one byte. power parameters.

Bit7: whether enable WritePower

0 -Disable;

1 -Enable.

Bit6 ~ Bit0:Power, range is 0 to 30.

Response:

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

8.4.32 Get Write Power

The command is used to Get power parameter when do write operation

Command:

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

Response:

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

Parameter explanation:

WritePower: one byte. power parameters.

Bit7: whether enable WritePower

0 -Disable;

1 -Enable.

Bit6 ~ Bit0:Power, range is 0 to 30.

8.4.33 Get or Set Rerty times

The command is used to get or set retry times when write failed.

Command:

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

Parameter explanation:

HeartBeatTime: one byte. Retry times parameters.

Bit7: Get/set retry times flag.

0 - Get write retry times;

1 - Set write retry times.

Bit6 ~ Bit0: a heartbeat packet interval.

Bit6~Bit3: Reserved.

If Bit7 = 0, Bit2 ~ Bit0 will be ignored;

If Bit7 = 1, Bit2 ~ Bit0 is retry times when write operation failed.range is 0 to 7.

Response:

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

8.4.35 Set detect multiple tag parameter

The command is used to set the detecting parameters when the reader is in auto-running or trigger mode in auto-running or trigger mode,the detecting command would not be seted through these,just depend on the parameters in the command which would be sent, See section 8.2.1

Command

Len	Adr	Cmd	Data[]		CRC-16	
			QValue	Session		
0x06	0xXX	0x81	0xXX	0xXX	LSB	MSB

Parameter explanation:

QValue: one bits, It is the initial Q used to detec EPC tag. The Q should be seted to meet the requirements that 2Q venue should be approximately equal to the number of tags. The range is 0~15, when wrong Q turn up in the command, error message will be sent back.

Session: one bits, It is used to detec EPC tag.

0x00: Session ues S0;

0x01: Session use S1;

0x02: Session use S2;

0x03: Session use S3.

Keep other venue, when wrong Session turn up in the command, error message will be sent back

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16	
0x05	0xXX	0x81	0x00	-	LSB	MSB

8.4.36 Get detect multiple tag parameter

The command is used to get the detecting parameters when detect multiple tag in auto-running or trigger mode

Command

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x82	-	LSB	MSB

Parameter explanation:

None.

Response:

Len	Adr	recmd	Status	Data[]		CRC-16	
				QValue	Sessionl	LSB	MSB
0x07	0xXX	0x82	0x00	0xXX	0xXX	LSB	MSB

8.4.37 R2000 module ROM update

The Command is used to change reader to update ROM of R2000-module mode, when reader Response this command, reader is upgrade mode, and Exit this mode requires the reader to power off and on.

Command

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x84	-	LSB	MSB

Parameter explanation:

None.

Response:

Len	Adr	recmd	Status	Data[]	CRC-16	
0x05	0xXX	0x84	0x00	-	LSB	MSB