RRU9813M UHF RFID Reader User’s Manual v1.10

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 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....11
        8.2.3 Write Data....12
        8.2.4 Write EPC....14
        8.2.5 Kill Tag....14
        8.2.6 Lock....15
        8.2.7 BlockErase....17
        8.2.8 Set Privacy with Mask Pattern....18
        8.2.9 Set Privacy without Mask Pattern....19
        8.2.10 Reset Privacy....20
        8.2.11 Check Privacy....20
        8.2.12 EAS Configure....20
        8.2.13 EAS Alarm....21
        8.2.14 Single Tag Inventory....22
        8.2.15 Block Write....22
        8.2.16 Read Monza4QT parameters....24
        8.2.17 Set Monza4QT parameters....25
        8.2.18 Extension Read Data....26
        8.2.19 Extension Write Data....27
        8.2.20 Buffer Inventory....29
    8.3 READER-DEFINED COMMAND....30
        8.3.1 Get Reader Information....30
        8.3.2 Set Region....32
        8.3.3 Set Address....32
        8.3.4 Set InventoryScanTime....33
        8.3.5 Set Baud Rate....33
        8.3.6 Set RF Power....34
        8.3.7 Beep Setting....34

8.3.8 Set GPIO....34
8.3.9 Get GPIO Status....35
8.3.10 Get Reader Serial number....35
8.3.11 Set Tag Custom function....35
8.3.12 Set Buffer EPC/TID length....36
8.3.13 Get Buffer EPC/TID length....36
8.3.14 Get Buffer data....37
8.3.15 Clear Buffer....37
8.3.16 Get Buffer tag number....38

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	Single Tag Inventory	0x0f	The function is used to get one tag’s EPC value in the field.
15	Block Write	0x10	The function is used to write multiple words in a Tag’s Reserved, EPC, TID, or User memory.
16	Read Monza4Qt	0x11	The function is used to get Monza4QT work parameters.
17	Set Monza4Qt	0x12	The function is used to Set Monza4QT work parameters.
18	Extension Read	0x15	The function is used to read part or all of a Tag’s Password, EPC, TID, or User memory.
19	Extension Write	0x16	The function is used to write several words in a Tag’s Reserved, EPC, TID, or User memory.
20	Buffer Inventory	0x18	The function is used to get tag count in the buffer

4.2 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	Set GPIO	0x46	The function is used to set GPIO status.
8	Get GPIO Status	0x47	The function is used to read GPIO status.
9	Get seria Number	0x4C	The function is used to get reader‘s seria number.
10	Set tag custom Function	0x3a	The function is used to set tag’s defined function..
11	Beep Setting	0x40	The function is used to set read’s beep status
12	Set buffer EPC/TID length	0x70	The function is used to set buffer EPC/TID length
13	Get buffer EPC/TID length	0x71	The function is used to get buffer EPC/TID length
14	Get buffer data	0x72	The function is used to get buffer data
15	Clear buffer	0x73	The function is used to clear buffer data
16	Get tag number	0x74	The function is used to get tag ‘s number in the buffer.

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.
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

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.

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	0xXX	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	0x80	0xXX

Parameter explanation:

QValue: One byte, Q value, range is 0-15. Q value setting should be field label quantity is approximately equal to 2 Q.

Session: One byte, Session value,range is 0-3.
0x00:S0;
0x01:S1;
0x02:S2;
0x03:S3;

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.

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,0x80, Other values are reserved.

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	0x01	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.

Ant: It describes from which antenna the tag EPC is collected,this module is 0x01.

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 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	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 0.

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 Single Tag Inventory

This command is used to get one tag's EPC value in the field.

Command:

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

Response:

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

Num: one byte. It specifies the number of tag inventoried and should always be 1 if tag detected.

Ant: It describes from which antenna the tag EPC is collected,this module is 0x01.

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

8.2.15 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

8.2.16 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:diabled;bit1=1:enabled.

8.2.17 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:diabled;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 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	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:diabled;bit1=1:enabled.

8.2.18 Extension 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.

8.2.19 Extension 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 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	0x16	0x00	-	LSB	MSB

8.2.20 Buffer Inventory

This command is used to identify the tabbed operation, at the same time will save tag data in buffer.

Diffent with 8.2.1 "office" command, legibility in predefined office hours continuous plate tag, the tag data stored in the internal cache area (buffer can be set to EPC maximum length 128 - bit or 496 bit two formats), office at the end of the office time arrival, and returns the buffer zone and the office of the total number of tags in tag number (the same tag read will be many times repeatedly counts). Allows users to cache the data extraction, clear the cache and query cache area commands such as tag number to access the data in the cache

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	0x18	-	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	0x80	0xXX

Parameter explanation:

QValue: One byte, Q value, range is 0-15. Q value setting should be field label quantity is approximately equal to 2 Q.

Session: One byte, Session value,range is 0-3.

0x00:S0;

0x01:S1;

0x02:S2;

0x03:S3;

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.

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,0x80, Other values are reserved.

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

Remark:

3 *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.*

4 *The* Target, Ant, ScanTime *is optional parameters,these parameters must be used to together.*

Response:

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

Parameter explanation:

BufferCount: 2 bytes, Total number of tag record in buffer, same EPC/TID data tag will be deemed to be the same. If not clear the buffer, tag number for many times the number of office operation accumulation

TagNum: 2 bytes,Read tag times current query, do not distinguish whether read many times with same tag.

8.3 READER-DEFINED COMMAND

8.3.1 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	
0x11	0xXX	0x21	0x00	Version, Type, Tr_Type, dmaxfre, dminfre, Power, Scntm, Ant, Beepen, Reserved, Reserved	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 0x0f for RRU9813M
Tr_Type	1	It specifies the protocol the reader supports. 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 8.3.2.
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 8.3.2.
Power	1	It specifies the reader's RF output power. The value range is 0 to 26.
Scntm	1	It specifies the InventoryScanTime. Please also refer to 8.2.1
Ant	1	RFU
Beepen	1	It specifies the Beep status.
Reserved	1	RFU
CheckAnt	1	RFU

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
…	…	…	…	…
1	1	1	1	RFU

8.3.2 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		
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:

MaxFre(Bit7)	MaxFre(Bit6)	MinFre(Bit7)	MinFre(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
…	…	…	…	…
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: Fs = 902.75 + N * 0.5 (MHz), N∈ [0, 49].
Korean band: Fs = 917.1 + N * 0.2 (MHz), N∈ [0, 31].
EU band: Fs = 865.1 + N*0.2(MHz) N∈[0, 14]。

8.3.3 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.*

8.3.4 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

8.3.5 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.*

8.3.6 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

8.3.7 Beep Setting

The command is used to set beep status.

Command:

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

Parameter explanation:

BeepEn: one byte.Bit0=0:Close;Bit0=1:Open.

Bit1-Bit7: reserved.default is 0;

Response:

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

8.3.8 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 2 output GPIOs' (OUT1~OUT2) level. Bit0~Bit1 correspond to OUT1~OUT2.

Response:

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

8.3.9 Get GPIO Status

The command is used to get the current 8 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 2 output GPIOs' (OUT1~OUT2) level and 2 input GPIOs' (INT1~INT2) level. Bit4~Bit5 correspond to OUT1~OUT2,Bit0~Bit1 correspond to INT1~INT2.

8.3.10 Get Reader Serial number

The command is used to get Reader serial number.

Command:

Len	Adr	Cmd	Data[]	CRC-16	
0x04	0xXX	0x4C	-	LSB	MSB

Response:

Len	Adr	reCmd	Status	Data[]	CRC-16
				SerialNo	

0x09	0xXX	0x4c	0x00	4bytes	LSB	MSB

Parameter explanation:

SerialNo: 4 bytes.Reader serial number.

8.3.11 Set Tag Custom function

The command is used to set tag custom function.

Command:

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

Parameter explanation:

InlayType: tag Type,range is 0-254.

Default is 0,not specify type.

Value 1 is enabled Monza4QT Peek function.

Response:

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

Parameter explanation:

InlayType: tag Type,range is 0-254.

8.3.12 Set Buffer EPC/TID length

The command is used to set the EPC/TID's length of tag in the buffer.

Command:

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

Parameter explanation:

SaveLen: one byte.EPC/TID length.

0x00:128bits;

0x01:496bits;

Response:

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

8.3.13 Get Buffer EPC/TID length

The command is used to get the EPC/TID’s length of tag in the buffer.

Command:

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

Parameter explanation:

None.

Response:

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

SaveLen: one byte.EPC/TID length.

0x00:128bits;

0x01:496bits;

8.3.14 Get Buffer data

The command is used to get data from buffer.

Command:

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

Parameter explanation:

None.

Response:

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

Status:

Value	Comment
0x01	Tag inventory operation is completely finished and all 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.

Num: the tag number of this single response.

EPC Data: the EPC/TID data in the buffer.

EPC-n				
Ant	**Len**	**EPC/TID**	**RSSI**	**Count**
0xXX	**0xXX**	**nBytes**	**0xXX**	**0xXX**

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.

Len :EPC/TID length of this tag.

EPC/TID: the tag's EPC/TID data.length is **Len**.

RSSI: the signal strength of the tag for the first Read time

Count: The number of the successful reading tag, the value 0xFF said number greater than or equal to 255 times.

8.3.15 Clear Buffer

The command is used to clear tag data in the buffer.

Command:

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

Response:

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

8.3.16 Get Buffer tag number.

The command is used to get tag number in the buffer.

Command:

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

Parameter explanation:

None.

Response:

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

Count: 2 bytes.tag number in the buffer.