PHYCHIPS

RED RCP
2015-11-05

Reader Control Protocol
User Manual

Rev. 1.0.6 2015-11-05

Contents

1 Revision History

Version	Date	Description
1.0.0	2015.03.05	Initial Release
1.0.1	2015.05.19	Add some functions
1.0.2	2015.05.28	Modified in section 4.28 Get Anti-Collision Mode Modified in section 4.29 Set Anti-Collision Mode
1.0.3	2015.06.10	Modified in section 3.2.2 Message code field Removed in section 4.18 Start Auto Read Removed in section 4.19 Stop Auto Read Added in section 4.18 Read Type C UII TID Modified in section 4.20 Read Type C Read Long Data Modified in section 4.27 Get Anti-Collision Mode Modified in section 4.28 Set Anti-Collision Mode Added in section 5 Use Case
1.0.4	2015.07.30	Modified in section 4.3 Get Region Modified in section 4.4 Set Region
1.0.5	2015.08.03	Added in Section.4.43 Get DTC Result
1.0.6	2015.11.05	Modified in section 4.18 Read Type C UII TID

2 Hardware Interface

It is possible to control PR9200 through UART, SPI or I2C. The interface type could be decided in PR9200 firmware. In order to use UART interface, the user should build firmware with UART RCP option. The UART format is described in the following section 2.1. SPI format is described in the following section 2.2 and the firmware should be built with SPI RCP option. Section 2.3 describes I2C format and the firmware should be built with I2C RCP option.

2.1 UART

The UART interface assigned to one channel; Pin P00 (RXD) and Pin P01 (TXD). The data is sent least significant bit (LSB) first. Signal format of data flow is shown in the figure below. Parameters for UART communication are 8 data bit, 1 stop bit, and no parity.

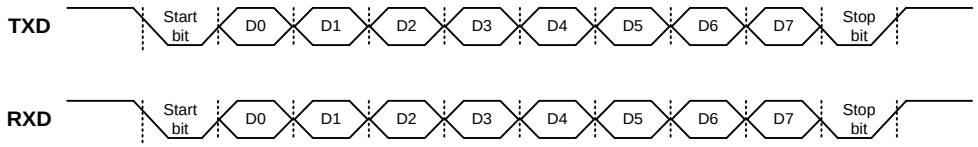


Figure 1 Serial Interface Format

2.2 SPI

PR9200 is operated as SPI slave and pins are assigned to P07 (SEL), P04 (SPI_TXD), P05 (SPI_RXD), P06 (SCK). The data is sent least significant bit (LSB) first. Signal format of data flow is shown in the figure below.

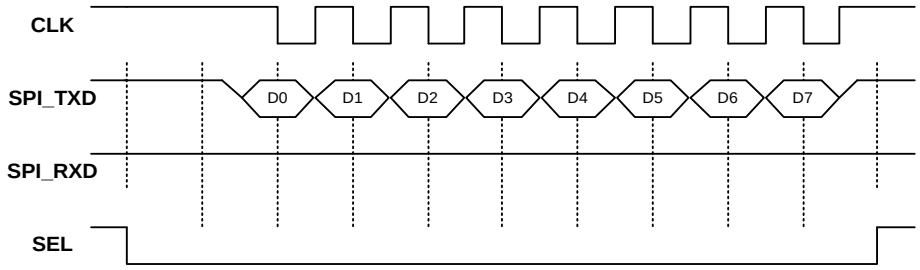


Figure 2 Master Write Mode

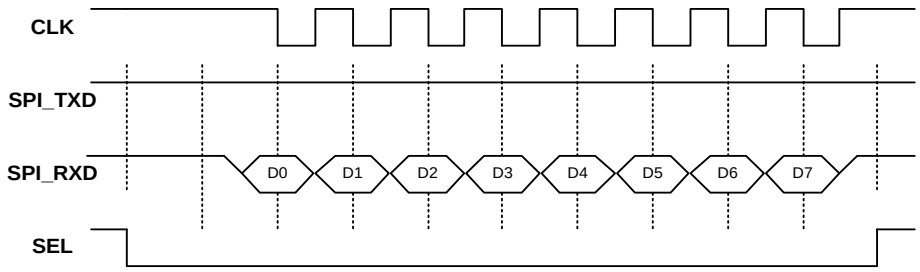


Figure 3 Master Read Mode

2.2.1 SPI Mode Switching

To switch between Master Write Mode and Master Read Mode in SPI interface, additional two bytes should be sent to PR9200. To retrieve a response and notification after the RCP command, the mode change bytes 0xBB and 0x0A are used. IRQ (P10) helps master determine time to send to mode change byte. Slave change IRQ to low when there is packet that slave response to master after command processing. When IRQ become low, Master send mode change bytes (0xBB, 0x0A) to slave. After retrieving the response, SPI interface shall be reverted to previous Master Writer Mode and IRQ return to high.

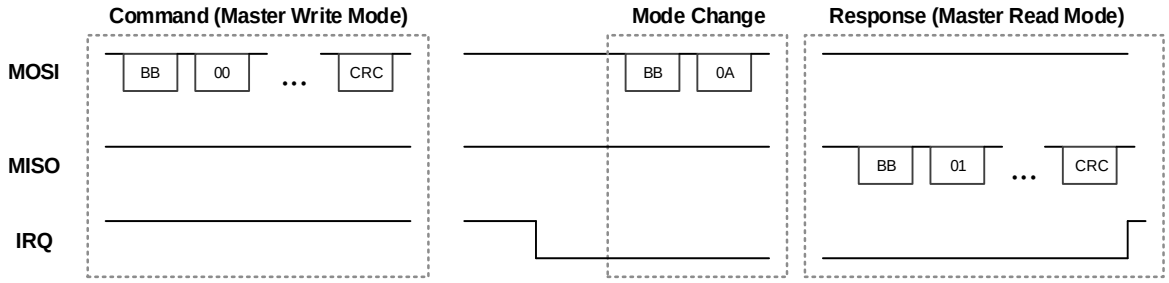


Figure 4 SPI command and response

Tag reading command should be sent to PR9200 before internal tag reading process. Finishing slave's tag reading, slave change IRQ to low. Then tag IDs could be retrieved using mode change bytes(0xBB, 0x0A). After retrieving tag IDs, SPI interface shall be reverted to previous Master Writer Mode. Also IRQ return to high.

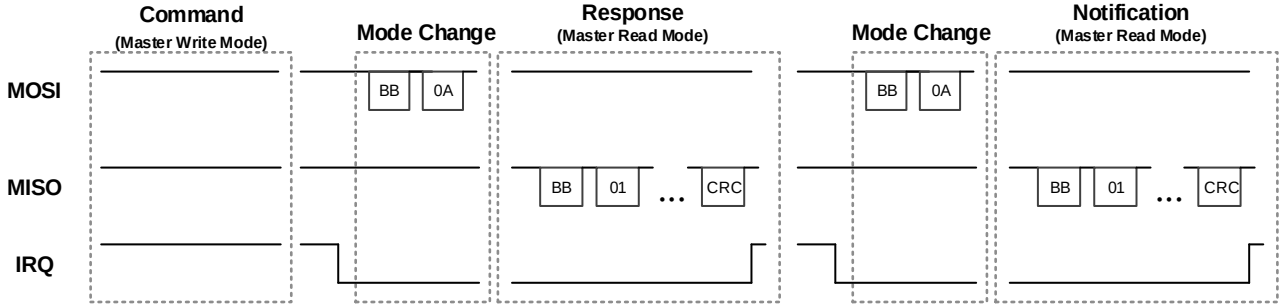


Figure 5 Reading tag IDs through SPI

2.3 I2C

PR9200 is operated as I2C slave and pins are assigned to P11 (SDA), P12 (SCK). The data is sent least significant bit (MSB) first. Signal format of data flow is shown in the figure below.

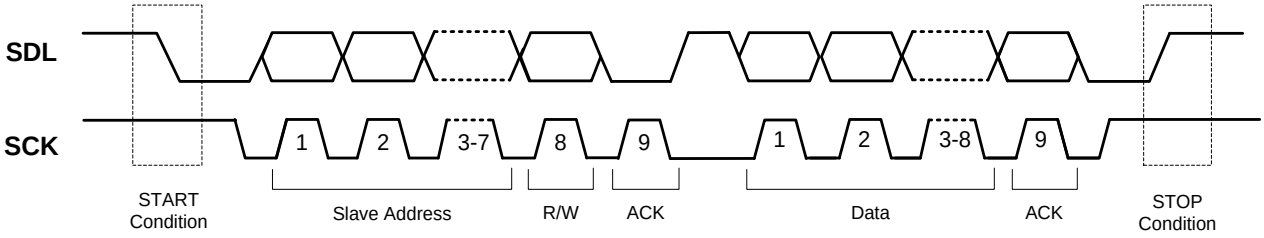


Figure 6 I2C Interface Format

2.3.1 I2C Mode Switching

I2C mode is determined according to mode bit of first byte. I2C mater sends to command and waits until IRQ become low. Slave change IRQ to low when there is packet that slave response to master after command processing.

When IRQ become low, Master receives response after switching mode bit to ‘read’. After retrieving the response, SPI interface shall be reverted to previous Master Writer Mode and IRQ return to high.

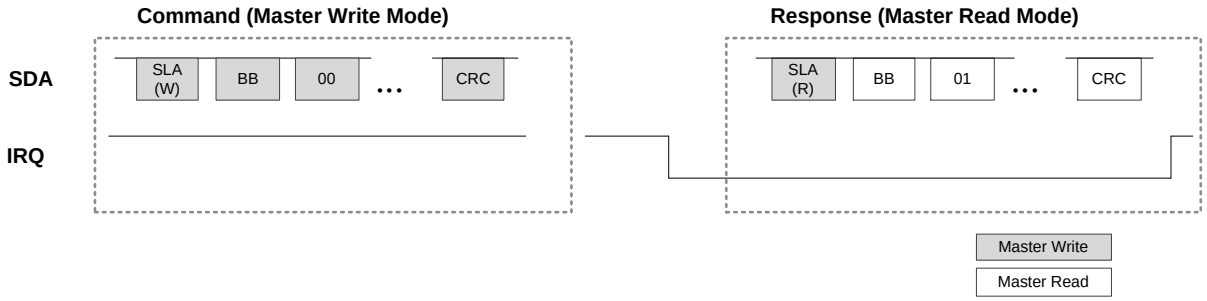


Figure 7 I2C command and response

3 RFID Reader Control Protocol overview

PR9200 UHF RFID reader is controlled through RCP(Reader Control Protocol,) which is using the UART serial interface. The RCP packet format is shown in the Figure 8 below. Preamble and end mark have constant values. 0xBB is used for preamble and 0x7E is used for end mark. Header consists of 3 fields: Message Type, Code, and Payload Length. Message Type field indicates packet types; command (0x00), response (0x01), notification (0x02). Code field is used to indicate control command type or response type. Payload Length field is used to inform PR9200 about payload length. Payload contains either data or control information.

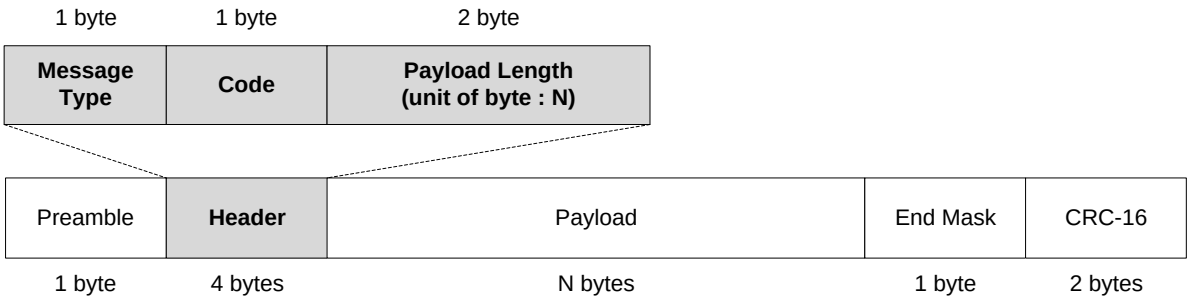


Figure 8 RCP packet format

RCP packet uses the big-endian convention. This means that high-order byte is filled first and low-order byte is filled last. In some cases, additional dummy bit 0s shall be added to pad out size of high-order byte.

3.1 Preamble and End Mark field

Preamble indicates the start of a RCP packet. Preamble has always the value 0xBB. End mark indicates the end of a RCP packet. End mark Preamble has always the value 0x7E. It is possible that a payload filed contains 0xBB or 0x7E (or both.) To tell these fields from other payload data, the header field has a payload length field.

3.2 Header Field

The header field is composed of 3 fields; message type, message codes, and payload length.

3.2.1 Message type field

The message type is used for indicating RCP packet type. Below table shows RCP packet types. Command packets are user-to-reader RCP packets. Response and notification RCP packets are reader-to-user RCP packets.

Type	Code value (HEX)
Command	0x00
Response	0x01
Notification	0x02
Reserved	0x03 to 0xFF

Table 1 Message Type

- Command and response

Command packets are used to control reader. After user sends a command packet to reader, a response packet is sent to user. All command packets have corresponding response packets.

- Notification

Unlike response packets, the notification packets are independently sent to user. In ‘Read Type C Tag ID Multiple’ mode, the notification packets have tag information and these packets are sent to user during reading round.

3.2.2 Message code field

Except for some commands, all packets may have two possible types; command and response packet. More details of using message code field follow next chapter.

Message code	Message Type	Code	UART	SPI	I²C
Set Reader Power Control	0x00 / 0x01	0x01	Yes	Yes	Yes
Get Reader Information	0x00 / 0x01	0x03	Yes	Yes	Yes
Get Region	0x00 / 0x01	0x06	Yes	Yes	Yes
Set Region	0x00 / 0x01	0x07	Yes	Yes	Yes
Set System Reset	0x00 / 0x01	0x08	Yes	Yes	Yes
Get Type C A/I Select Parameters	0x00 / 0x01	0x0B	Yes	Yes	Yes
Set Type C A/I Select Parameters	0x00 / 0x01	0x0C	Yes	Yes	Yes
Get Type C A/I Query Related Parameters	0x00 / 0x01	0x0D	Yes	Yes	Yes
Set Type C A/I Query Related Parameters	0x00 / 0x01	0x0E	Yes	Yes	Yes
Get current RF Channel	0x00 / 0x01	0x11	Yes	Yes	Yes
Set current RF Channel	0x00 / 0x01	0x12	Yes	Yes	Yes
Get FH and LBT Parameters	0x00 / 0x01	0x13	Yes	Yes	Yes
Set FH and LBT Parameters	0x00 / 0x01	0x14	Yes	Yes	Yes
Get Tx Power Level	0x00 / 0x01	0x15	Yes	Yes	Yes
Set Tx Power Level	0x00 / 0x01	0x16	Yes	Yes	Yes
RF CW signal control	0x00 / 0x01	0x17	Yes	Yes	Yes
Read Type C UII	0x00 / 0x01	0x22	Yes	Yes	Yes
Read Type C UII RSSI	0x02	0x23	Yes	Yes	Yes
Read Type C UII TID	0x00 / 0x01 / 0x02	0x25	Yes	Yes	Yes
Read Type C Tag Data	0x00 / 0x01	0x29	Yes	Yes	Yes
Read Type C Tag Long Data	0x00 / 0x01 / 0x02	0x2A	Yes	Yes	Yes
Get Session	0x00 / 0x01	0x2E	Yes	Yes	Yes
Set Session	0x00 /0x01	0x2F	Yes	Yes	Yes
Get Frequency Hopping Table	0x00 / 0x01	0x30	Yes	Yes	Yes
Set Frequency Hopping Table	0x00 / 0x01	0x31	Yes	Yes	Yes
Get Modulation	0x00 / 0x01	0x32	Yes	Yes	Yes
Set Modulation	0x00 / 0x01	0x33	Yes	Yes	Yes
Get Anti-Collision Mode	0x00 / 0x01	0x34	Yes	Yes	Yes
Set Anti-Collision Mode	0x00 / 0x01	0x35	Yes	Yes	Yes
Start Auto Read2	0x00 / 0x01 / 0x02	0x36	Yes	Yes	Yes
Stop Auto Read2	0x00 / 0x01	0x37	Yes	Yes	Yes
Start Auto Read RSSI	0x00 / 0x01 / 0x02	0x38	Yes	Yes	Yes
Stop Auto Read RSSI	0x00 / 0x01	0x39	Yes	Yes	Yes
Write Type C Tag Data	0x00 / 0x01	0x46	Yes	Yes	Yes
BlockWrite Type C Tag Data	0x00 / 0x01	0x47	Yes	Yes	Yes
BlockErase Type C Tag Data	0x00 / 0x01	0x48	Yes	Yes	Yes
Generic Transport	0x00/ 0x01	0x4D	Yes	Yes	Yes
BlockPermalock Type C Tag	0x00 / 0x01	0x83	Yes	Yes	Yes
Kill/Recom Type C Tag	0x00 / 0x01	0x65	Yes	Yes	Yes
Lock Type C Tag	0x00 / 0x01	0x82	Yes	Yes	Yes
Get Temperature	0x00 / 0x01	0xB7	Yes	Yes	Yes
Get RSSI	0x00 / 0x01	0xC5	Yes	Yes	Yes
Scan RSSI	0x00 / 0x01	0xC6	Yes	Yes	Yes
Get DTC Result	0x00 / 0x01	0xCA	Yes	Yes	Yes

Update Registry	0x00 / 0x01	0xD2	Yes	Yes	Yes
Get Registry Item	0x00 / 0x01	0xD4	Yes	Yes	Yes
Command Failure	0x01	0xFF	Yes	Yes	Yes
Set Optimum Frequency Hopping Table	0x00 / 0x01	0xE4	Yes	Yes	Yes
Get Frequency Hopping Mode	0x00 / 0x01	0xE5	Yes	Yes	Yes
Set Frequency Hopping Mode	0x00 / 0x01	0xE6	Yes	Yes	Yes
Get Tx Leakage RSSI Level for Smart hopping Mode	0x00 / 0x01	0xE7	Yes	Yes	Yes
Set Tx Leakage RSSI Level for Smart hopping Mode	0x00 / 0x01	0xE8	Yes	Yes	Yes

Table 2 Message codes

3.2.3 Payload length

The header is used to indicate the length of payload that is succeeding to payload length field. Payload length is expressed in 2 bytes.

3.3 Payload Field

Payload field contains either data or control information, depending on the packet type. For command packets, the control information is placed here. For response and notification packets, data information is placed here instead.

3.4 Cyclic Redundancy Check(CRC) Field

3.4.1 CRC General

The Command and Response use the same CRC-16 for verify a purity of message bits. The 16-bit CRC shall be calculated on all the message bits from the message type field to the end mark field. The Polynomial used to calculate the CRC is $X^{16}+X^{12}+X^5+1$ (initial value is 0xFFFF). The resulting CRC value shall be attached to the end of the packet (after End Mark filed) and transmitted. The most significant byte shall be transmitted first. The most significant bit of each byte shall be transmitted first.

An exemplary schematic diagram for a CRC-16 circuit is shown in below figure. The polynomial used to calculate the CRC-16, $X^{16}+X^{12}+X^5+1$, is the CRC-CCITT international standard, ITU recommendation X.25.

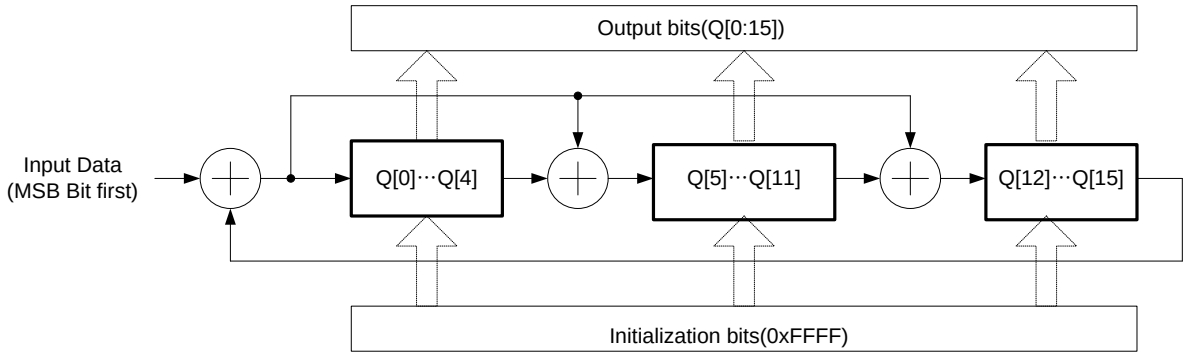


Figure 9 CRC-16 circuit

To calculate a CRC-16, first preload the entire CRC register(i.e. Q[15:0], Q15 is the MSB and Q0 is the LSB) with the value 0xFFFF. Second, clock the data bits to be encoded into the Input Data, MSB first. After clocking in all the data bits, Q[15:0] holds the CRC-16.

There are two methods to check a CRC-16

3.4.2 Inversion of incoming CRC bits by the receiving part.

First preload the entire CRC register(Q[15:0]) with the value 0xFFFF. Second, clock the received data bits into the Input Data, MSB first. Third, invert all bits of the received CRC-16, and clock the inverted CRC-16 bits into the Input Data, MSB first. The CRC-16 check passed if the value in Q[15:0]=0x1D0F

3.4.3 Non-inversion of incoming CRC bits by the receiving part.

First preload the entire CRC register(Q[15:0]) with the value 0xFFFF, then clock the received data and CRC-16 bits into the Input Data, MSB first. The CRC-16 check passed if the value in Q[15:0]=0x0000.

4 Details of Command, Response and Notification

4.1 Set Reader Power Mode

Set power mode.

4.1.1 Command

Message Type: Command (0x00)
Code: Set Reader Power Control (0x01)
Arguments
- Parameter (8-bit): SLEEP Mode (0x00), DEEPSLEEP Mode (0x01)

Example) Sleep mode

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x01	0x00	0x01	0x00	0x7E	0xNNNN

4.1.2 Response

Message Type: Response (0x01)
Code: Reader Power Control (0x01)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x01	0x00	0x01	0x00	0x7E	0xNNNN

*In order to change Sleep(or Deepsleep) mode to Normal mode, user must use external interrupt signal, P03. While module is Sleep mode, go down pin P03 to GND(Logic low) and you can exit Sleep mode.

4.2 Get Reader Information

Get basic information from the reader.

4.2.1 Command

Message Type: Command (0x00)
Code: Get Reader Information (0x03)
Arguments
- Model (0x00)
- S/N (0x01)
- Manufacturer (0x02)
- Frequency (0x03)
- Tag Type (0x04)

Example1) Reads reader manufacturer

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x03	0x00	0x01	0x02	0x7E	0xNNNN

Example2) Reads tag type

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x03	0x00	0x01	0x04	0x7E	0xNNNN

4.2.2 Response

Message Type: Response (0x01)
Code: Get Reader Information (0x03)
Arguments
- String (variable length)

Example1) Manufacturer = PHYCHIPS

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Argument		
0xBB	0x01	0x03	0x00	0x08	0x50 (P)	0x48 (H)	0x59 (Y)
Argument					End Mark	CRC-16	
0x43 (C)	0x48 (H)	0x49 (I)	0x50 (P)	0x53 (S)	0x7E	0xNNNN	

Example2) Tag Type = ISO 18000-6 Type B(0x01), ISO 18000-6 Type C(0x02)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Argument	End Mark	CRC-16
0xBB	0x01	0x03	0x00	0x01	0x02	0x7E	0xNNNN

4.3 Get Region

Get the current region. PR9200 uses individual channel table that depends on region. List of region code follows below.

4.3.1 Command

Message Type: Command (0x00)
Code: Get Region (0x06)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x06	0x00	0x00	0x7E	0xNNNN

4.3.2 Response

Message Type: Response (0x01)
Code: Get Region (0x06)
- Korea (0x11)
- North America (0x21)
- US (0x22)
- Europe (0x31)
- Japan (0x41)
- China1 (0x51)
- China2 (0x52)
- Brazil (0x61)

Example) Europe

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x06	0x00	0x01	0x31	0x7E	0xNNNN

4.4 Set Region

Set the current region. PR9200 uses individual channel table that depends on region. List of region code follows below.

4.4.1 Command

Message Type: Command (0x00)
Code: Set Region (0x07)
Arguments
- Korea (0x11)
- North America (0x21)
- US (0x22)
- Europe (0x31)
- Japan (0x41)
- China1 (0x51)
- China2 (0x52)
- Brazil (0x61)

Example) Europe

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x07	0x00	0x01	0x31	0x7E	0xNNNN

4.4.2 Response

Message Type: Response (0x01)
Code: Set Region (0x07)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x07	0x00	0x01	0x00	0x7E	0xNNNN

4.5 Set System Reset

Set the system level reset.

4.5.1 Command

Message Type: Command (0x00)
Code: Set System Reset (0x08)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x08	0x00	0x00	0x7E	0xNNNN

4.5.2 Response

Message Type: Response (0x01)
Code: Set System Reset (0x08)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x08	0x00	0x01	0x00	0x7E	0xNNNN

4.6 Get Type C A/I Select Parameters

Get 18000-6C air interface protocol command 'Select' parameters.

4.6.1 Command

Message Type: Command (0x00)
Code: Get Type C A/I Select Parameters (0x0B)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x0B	0x00	0x00	0x7E	0xNNNN

4.6.2 Response

Message Type: Response (0x01)
Code: Get Type C A/I Select Parameters (0x0B)
Arguments
- Target (3-bit): S0 (000), S1 (001), S2 (010), S3 (011), SL (100)
- Action (3-bit): Refer to ISO18000-6C.
- Memory Bank (2-bit): 00 RFU, 01 EPC, 10 TID, 11 User
- Pointer (32-bit): Starting mask address
- Length (8-bit): mask length bits
- Truncate (1-bit): Enable (1) and Disable (0)
- Reserve (7-bit): Reserved 0000000 value should be placed here.
- Mask (0~255 bits): Mask value

Example)
Target=S0, Action=assert SL or inventoried - > A, MB=User, Pointer = 0x000000FF,
Length =0x20, T=0, Mask = 11111111111111110000000000000000

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	T	A	M	Ptr (MSB)	
0xBB	0x01	0x0B	0x00	0x0B	000	000	11	0x00	0x00
	Ptr (LSB)	Length	T	Reserve	Mask (MSB)				Mask (LSB)
0x00	0xFF	0x20	0	0000000	0xFF	0xFF		0x00	0x00
End Mark	CRC-16								
0x7E	0xNNNN								

4.7 Set Type C A/I Select Parameters

Set 18000-6C air interface protocol command 'Select' parameters.

4.7.1 Command

Message Type: Command (0x00)
Code: Set Type C A/I Select Parameters (0x0C)
Arguments
- Target (3-bit): S0 (000), S1 (001), S2 (010), S3 (011), SL (100)
- Action (3-bit): Refer to ISO18000-6C.
- Memory Bank (2-bit): RFU (00), EPC (01), TID (10), User (11)
- Pointer (32-bit): Starting mask address
- Length (8-bit): mask length bits
- Truncate (1-bit): Enable (1) and Disable (0)
- Reserve (7-bit): Reserved 0000000 value should be placed here.
- Mask (0~255 bits): Mask value

Example)
Target=S0 where C, Action=assert SL ors inventoried - > A, MB=User, Pointer = 0x000000FF
Length=0x20, T=0, Mask=11111111111111110000000000000000

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	T	A	M	Ptr (MSB)	
0xBB	0x00	0x0C	0x00	0x0B	000	000	11	0x00	0x00
	Ptr (LSB)	Length	T	RFU	Mask(MSB)				Mask(LSB)
0x00	0xFF	0x20	0	0000000	0xFF	0xFF		0x00	0x00
End Mark	CRC-16								
0x7E	0xNNNN								

4.7.2 Response

Message Type: Response (0x01)
Code: Set Type C A/I Select Parameters (0x0C)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x0C	0x00	0x01	0x00	0x7E	0xNNNN

4.8 Get Type C A/I Query Parameters

Get 18000-6C air interface protocol command 'Query' parameters.

4.8.1 Command

Message Type: Command (0x00)
Code: Get Type C A/I Query Parameters (0x0D)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x0D	0x00	0x00	0x7E	0xNNNN

4.8.2 Response

Message Type: Response (0x01)
Code: Get Type C A/I Query Parameters (0x0D)
Arguments
- DR (1-bit): DR=8 (0), DR=64/3 (1)
- M (2-bit): M=1 (00), M=2 (01), M=4 (10), M=8 (11)
- TRext (1-bit): No pilot tone (0), Use pilot tone (1)
- Sel (2-bit): All (00 or 01), ~SL (10), SL (11)
- Session (2-bit): S0 (00), S1 (01), S2 (10), S3 (11)
- Target (1-bit): A (0), B (1)
- Q (4-bit): 0-15; the number of slots in the round.

Example) DR=8, M=1, TRext=Use pilot tone, Sel=All, Session=S0, Target=A, Q=4, No change to Q

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	DR	M	TR	Sel	S
0xBB	0x01	0x0D	0x00	0x02	0	00	1	00	00

T	Q	RSV	End Mark	CRC-16
0	0100	000	0x7E	0xNNNN

4.9 Set Type C A/I Query Parameters

Set 18000-6C air interface protocol command 'Query' parameters.

4.9.1 Command

Message Type: Command (0x00)
Code: Set Type C A/I Query Parameters (0x0E)
Arguments
- DR (1-bit): DR=8 (0), DR=64/3 (1)
- M (2-bit): M=1 (00), M=2 (01), M=4 (10), M=8 (11)
- TRext (1-bit): No pilot tone (0), Use pilot tone (1)
- Sel (2-bit): All (00 or 01), ~SL (10), SL (11)
- Session (2-bit): S0 (00), S1 (01), S2 (10), S3 (11)
- Target (1-bit): A (0), B (1)
- Q (4-bit): 0-15; the number of slots in the round.

Example) DR=8, M=1, TRext=Use pilot tone, Sel=All, Session=S0, Target=A, Q=4, No change to Q

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	DR	M	TR	Sel	S
0xBB	0x00	0x0E	0x00	0x02	0	00	1	00	00

T	Q	RSV	End Mark	CRC-16
0	0100	000	0x7E	0xNNNN

4.9.2 Response

Message Type: Response (0x01)
Code: Set Type C A/I Query Parameters (0x0E)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x0E	0x00	0x01	0x00	0x7E	0xNNNN

4.10 Get current RF Channel

Get RF channel. This command is valid only for non-FH mode.

4.10.1 Command

Message Type: Command (0x00)
Code: Get current RF Channel (0x11)
Arguments
- None

Example) Get current RF channel

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x11	0x00	0x00	0x7E	0xNNNN

4.10.2 Response

Message Type: Response (0x01)
Code: Get current RF Channel (0x11)
Arguments
- CN (8-bit): Channel Number. The range of channel number depends on regional settings
- CNO (8-bit): Channel number offset for miller subcarrier.

Example) Channel Number = 10

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	CN	CNO	End Mark
0xBB	0x01	0x11	0x00	0x02	0x0A	0x00	0x7E

CRC-16
0xNNNN

4.11 Set current RF Channel

Set RF channel. This command is valid only for non-FHSS mode.

4.11.1 Command

Message Type: Command (0x00)
Code: Set current RF Channel (0x12)
Arguments
- CN (8-bit): Channel number. The range of channel number depends on regional settings
- CNO (8-bit): Channel number offset for miller subcarrier.

Example) Channel Number = 10, Channel Number Offset = 0

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	CN	CNO	End Mark
0xBB	0x00	0x12	0x00	0x02	0x0A	0x00	0x7E

CRC-16
0xNNNN

4.11.2 Response

Message Type: Response (0x01)
Code: Set current RF Channel (0x12)
Arguments
- None

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x12	0x00	0x01	0x00	0x7E	0xNNNN

4.12 Get FH and LBT Parameters

Get FH and LBT control

4.12.1 Command

Message Type: Command (0x00)
Code: Get FH and LBT Parameters (0x13)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x13	0x00	0x00	0x7E	0xNNNN

4.12.2 Response

Message Type: Response (0x01)
Code: Get FH and LBT Parameters (0x13)
Arguments
- RT (16-bit): Read Time (1 = 1ms)
- IT (16-bit): Idle Time (1 = 1ms)
- CST (16-bit): Carrier Sense Time (1 = 1ms)
- RFL (16-bit): Target RF power level (-dBm x 10)
- FH (8-bit): enable (0x01 or over) / disable (0x00)
- LBT (8-bit): enable (0x01 or over) / disable (0x00)
- CW (8-bit): enable (0x01) / disable (0x00)

Example) Success, FH disable, LBT enable, RT 400ms, IT 100ms, CST 10ms, RFL -630 (-63.0 dBm)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	RT MSB	RT (LSB)	IT (MSB)
0xBB	0x01	0x13	0x00	0x0B	0x01	0x90	0x00
IT (LSB)	CST (MSB)	CST (LSB)	RFL (MSB)	RFL (LSB)	FH	LBT	CW
0x64	0x00	0x0A	0xFD	0x8A	0x00	0x01	0x00
End Mark	CRC-16						
0x7E	0xNNNN						

4.13 Set FH and LBT Parameters

Set FH and LBT Parameters

4.13.1 Command

Message Type: Command (0x00)
Code: Set FH and LBT Parameters (0x14)
Arguments
- RT (16-bit): Read Time (1 = 1ms)
- IT (16-bit): Idle Time (1 = 1ms)
- CST (16-bit): Carrier Sense Time (1 = 1ms)
- RFL (16-bit): Target RF power level (-dBm x 10)
- FH (8-bit): enable (0x01 or over) / disable (0x00)
- LBT (8-bit): enable (0x01 or over) / disable (0x00)
- CW (8-bit): enable (0x01) / disable (0x00)

Example1) FH enable (with LBT feature), RT 400ms, IT 100ms, CST 10ms, RFL -740 (-74.0 dBm)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	RT MSB	RT (LSB)	IT (MSB)
0xBB	0x00	0x14	0x00	0x0B	0x01	0x90	0x00
IT (LSB)	CST (MSB)	CST (LSB)	RFL (MSB)	RFL (LSB)	FH	LBT	CW
0x64	0x00	0x0A	0xFD	0x8A	0x01	0x01	0x00
End Mark	CRC-16						
0x7E	0xNNNN						

Example2) LBT enable (with FH feature), RT 400ms, IT 100ms, CST 10ms, RFL -740 (-74.0 dBm)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	RT MSB	RT (LSB)	IT (MSB)
0xBB	0x00	0x14	0x00	0x0B	0x01	0x90	0x00
IT (LSB)	CST (MSB)	CST (LSB)	RFL (MSB)	RFL (LSB)	FH	LBT	CW
0x64	0x00	0x0A	0xFD	0x8A	0x01	0x02	0x00
End Mark	CRC-16						
0x7E	0xNNNN						

4.13.2 Response

Message Type: Response (0x01)
Code: Set FH and LBT Parameters (0x14)
Arguments
- None

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x14	0x00	0x01	0x00	0x7E	0xNNNN

4.14 Get Tx Power Level

Get current, minimum, maximum Tx power level

4.14.1 Command

Message Type: Command (0x00)
Code: Get Tx Power Level (0x15)
Arguments
- None

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x15	0x00	0x00	0x7E	0xNNNN

4.14.2 Response

Message Type: Response (0x01)
Code: Get Tx Power Level (0x15)
Arguments
- PWR (16-bit): Current Tx Power
- Min PWR (16-bit): Min Tx Power
- Max PWR (16-bit): Max Tx Power

Example) PWR = 200 (20.0 dBm), Min PWR = 180 (18.0 dBm), Max PWR = 250 (25.0 dBm)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	PWR (MSB)	PWR (LSB)	Min PWR (MSB)
0xBB	0x01	0x15	0x00	0x02	0x00	0xC8	0x00

Min PWR (LSB)	Max PWR (MSB)	Max PWR (LSB)	End Mark	CRC-16
0xB4	0x00	0xFA	0x7E	0xNNNN

4.15 Set Tx Power Level

Set current Tx power level.

4.15.1 Command

Message Type: Command (0x00)
Code: Set Tx Power Level (0x16)
Arguments
- PWR (16-bit): Tx Power

Example) PWR = 200 (20.0 dBm)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	PWR (MSB)	PWR (LSB)	End Mark
0xBB	0x00	0x16	0x00	0x02	0x00	0xC8	0x7E

CRC-16
0xNNNN

4.15.2 Response

Message Type: Response (0x01)
Code: Set Tx Power Level (0x16)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x16	0x00	0x01	0x00	0x7E	0xNNNN

4.16 RF CW signal control

Turn the Continuous Wave (CW) signal on/off. This command packet is only valid for idle mode.

4.16.1 Command

Message Type: Command (0x00)
Code: RF CW signal control (0x17)
Arguments
- On (0xFF)
- Off (0x00)

Example) Turn RF CW signal on.

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x17	0x00	0x01	0xFF	0x7E	0xNNNN

4.16.2 Response

Message Type: Response (0x01)
Code: RF CW signal control (0x17)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x17	0x00	0x01	0x00	0x7E	0xNNNN

4.17 Read Type C UII

Read a EPC Block (PC + EPC)

4.17.1 Command

Message Type: Command (0x00)
Code: Read Type C UII (0x22)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x22	0x00	0x00	0x7E	0xNNNN

4.17.2 Response

Message Type: Response (0x01)
Code: Read Type C UII (0x22)
Arguments
- EPC Block (PC + EPC)

Example) PC = 0x3000, EPC = 0xE2003411B802011383258566

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	PC (MSB)	PC (LSB)	EPC (MSB)
0xBB	0x01	0x22	0x00	0x0E	0x30	0x00	0xE2
0x00	0x34	0x11	0xB8	0x02	0x01	0x13	0x83
		EPC (LSB)	End Mark	CRC-16			
0x25	0x85	0x66	0x7E	0xNNNN			

4.18 Read Type C UII TID

Start an automatic tag read operation, tag IDs with TID are sent back to user though notification packet.

4.18.1 Command

Message Type: Command (0x00)
Code: Read Type C UII TID (0x25)
Arguments
- MTNU: maximum number of tag to read
- MTIME: maximum elapsed time to tagging (sec)
- RC (16-bit): Repeat cycle (how many times reader perform inventory round).

Example) MTNU = 0, MTIME = 0, Repeat Cycle = 100

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	MTNU	MTIME	RC(MSB)
0xBB	0x00	0x25	0x00	0x04	0x00	0x00	0x00
RC(LSB)	End Mark	CRC-16					
0x64	0x7E	0xNNNN					

4.18.2 Response

Message Type: Response (0x01)
Code: Start Auto Read TID (0x25)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x25	0x00	0x01	0x00	0x7E	0xNNNN

4.18.3 Notification

Message Type: Notification (0x02)
Code: Read Type C UII TID (0x25)
Arguments
- EPC Block (PC + EPC)
- TID Block (Variable)

Example) PC = 0x3000, EPC = 0xE2003411B802011383258566, TID = 0xE2003411B8020113

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	PC(MSB)	PC(LSB)	EPC (MSB)
0xBB	0x02	0x25	0x00	0x16	0x30	0x00	0xE2
0x00	0x34	0x11	0xB8	0x02	0x01	0x13	0x83
		EPC (LSB)	TID (MSB)				
0x25	0x85	0x66	0xE2	0x00	0x34	0x11	0xB8
		TID (LSB)	End Mark	CRC-16			
0x02	0x01	0x13	0x7E	0xNNNN			

Message Type: Notification (0x02)
Code: Read Type C UII TID (0x25)
Arguments
- Read complete (0x1F)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x02	0x25	0x00	0x01	0x1F	0x7E	0xNNNN

4.19 Read Type C Tag Data

Read Type C tag data from specified memory bank.

4.19.1 Command

Message Type: Command (0x00)
Code: Read Type C Tag Memory (0x29)
Arguments
- AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filled to 0x00000000.
- UL (16-bit): Target tag's EPC length
- EPC (variable): Target tag's EPC
- MB (8-bit): Target memory bank; RFU (0x00), EPC (0x01), TID (0x02), User (0x03)
- SA (16-bit): Starting Address word pointer
- DL (16-bit): Data Length (Word Count)

Note: The Read Type C Tag Data command supports maximum 128 word.

Example)
Access Password = 0x00000000, UL = 12 (0x0C) byte,
EPC = 0xE2003411B802011526370494, Target memory bank = RFU, Start Address = 0x0000, Length = 4 word

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP (MSB)		
0xBB	0x00	0x29	0x00	0x17	0x00	0x00	0x00
AP (LSB)	UL (MSB)	UL (LSB)	EPC (MSB)				
0x00	0x00	0x0C	0xE2	0x00	0x34	0x11	0xB8
						EPC (LSB)	MB
0x02	0x01	0x15	0x26	0x37	0x04	0x94	0x00
SA (MSB)	SA (LSB)	DL (MSB)	DL (LSB)	End Mark	CRC-16		
0x00	0x00	0x00	0x04	0x7E	0xNNNN		

4.19.2 Response

Message Type: Response (0x01)
Code: Read Type C Tag Memory (0x29)
Arguments
- Tag memory contents (variable)

Example) RFU memory bank = 0x0000000000000000

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Argument		
0xBB	0x01	0x29	0x00	0x08	0x00	0x00	0x00
Argument					End Mark	CRC-16	
0x00	0x00	0x00	0x00	0x00	0x7E	0xNNNN	

4.20 Read Type C Tag Long Data

Read Type C tag data from specified memory bank. This is extended command of 4.20 Read Type C Tag Data. This command can be used to read over 128 word. In case word count is over 128 word, reader reads and reports 128 word at a time until all is reads.

4.20.1 Command

Message Type: Command (0x00)
Code: Read Type C Tag Long Data (0x2A)
Arguments
- AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filled to 0x00000000.
- UL (16-bit): Target tag's EPC length
- EPC (variable): Target tag's EPC
- MB (8-bit): Target memory bank; RFU (0x00), EPC (0x01), TID (0x02), User (0x03)
- SA (16-bit): Starting Address word pointer
- DL (16-bit): Data Length (Word Count)

Note: The Read Type C Tag Long Data command does not support a DL(Word Count) of "0".

Example)
Access Password = 0x00000000, UL = 12 (0x0C) byte,
EPC = 0xE2003411B802011526370494, Target memory bank = User, Start Address = 0x0000, Length = 200 word

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP (MSB)		
0xBB	0x00	0x2A	0x00	0x17	0x00	0x00	0x00
AP (LSB)	UL (MSB)	UL (LSB)	EPC (MSB)				
0x00	0x00	0x0C	0xE2	0x00	0x34	0x11	0xB8
						EPC (LSB)	MB
0x02	0x01	0x15	0x26	0x37	0x04	0x94	0x03
SA (MSB)	SA (LSB)	DL (MSB)	DL (LSB)	End Mark	CRC-16		
0x00	0x00	0x00	0xC8	0x7E	0xNNNN		

4.20.2 Response

Message Type: Response (0x01)
Code: Read Type C Tag Long Data (0x2A)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x2A	0x00	0x01	0x00	0x7E	0xNNNN

4.20.3 Notification

Message Type: Notification (0x02)
Code: Read Type C Tag Long Data (0x2A)
Arguments
- Start Address (16-bit): start address to report tag memory contents
- Word Count (8-bit): word count to report tag memory contents
- Tag memory contents (variable)

Example) First Notification: Start Address = 0x0000, Word Count = 0x80,
First Data Rx Data = 0xAAAAAAAAAAAAAAAAAAAAAA…..

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	SA (MSB)	SA (LSB)	Word Count

0xBB	0x02	0x2A	0x01	0x03	0x00	0x00	0x80
RxData (MSB)							
0xAA	0xAA	…	…	...	…	…	…
		RxData (LSB)	End Mark	CRC-16			
…	0xAA	0xAA	0x7E	0xNNNN			

Second Notification: Start Address = 0x0080, Word Count = 0x48,
Second Data Rx Data = 0xAAAAAAAAAAAAAAAAAAAA….

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	SA (MSB)	SA (LSB)	Word Count
0xBB	0x02	0x2A	0x00	0x93	0x00	0x80	0x48
RxData (MSB)							
0xAA	0xAA	…	…	...	…	…	…
		RxData (LSB)	End Mark	CRC-16			
…	0xAA	0xAA	0x7E	0xNNNN			

Message Type: Notification (0x02)
Code: Read Type C Tag Long Data (0x2A)
Arguments
- Read complete (0x1F)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x02	0x2A	0x00	0x01	0x1F	0x7E	0xNNNN

4.21 Get Session

Get current session.

4.21.1 Command

Message Type: Command (0x00)
Code: Get Session (0x2E)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x2E	0x00	0x00	0x7E	0xNNNN

4.21.2 Response

Message Type: Response (0x01)
Code: Get Session (0x2E)
Arguments
- Session (8-bit): S0(0x00), S1(0x01), S2(0x02), S3(0x03), Dev.mode(0xF0)

Example) S0

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Session	End Mark	CRC-16
0xBB	0x01	0x2E	0x00	0x01	0x00	0x7E	0xNNNN

4.22 Set Session

Set current session.

4.22.1 Command

Message Type: Command (0x00)
Code: Set Session (0x2F)
Arguments
- Session (8-bit): S0(0x00), S1(0x01), S2(0x02), S3(0x03), Dev.mode(0xF0)

Example) S0

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Session	End Mark	CRC-16
0xBB	0x00	0x2F	0x00	0x00	0x00	0x7E	0xNNNN

4.22.2 Response

Message Type: Response (0x01)
Code: Set Session (0x2F)
Arguments
- None

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x01	0x2F	0x00	0x00	0x7E	0xNNNN

4.23 Get Frequency Hopping Table

Get current frequency hopping table.

4.23.1 Command

Message Type: Command (0x00)
Code: Get Frequency Hopping Table (0x30)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x30	0x00	0x00	0x7E	0xNNNN

4.23.2 Response

Message Type: Response (0x01)
Code: Get Frequency Hopping Table (0x30)
Arguments
- Table Size (8-bit)
- Channel Number (variable)

Example) Table Size = 6, channel numbers = 47, 19, 20, 23, 46, 16

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Table Size	Argument	
0xBB	0x01	0x30	0x00	0x07	0x06	0x2F	0x13
Argument				End Mark	CRC-16		
0x14	0x17	0x2E	0x10	0x7E	0xNNNN		

4.24 Set Frequency Hopping Table

Set current frequency hopping table.

4.24.1 Command

Message Type: Command (0x00)
Code: Set Frequency Hopping Table (0x31)
Arguments
- Table Size (8-bit)
- Channel Numbers (variable)

Example)
Table Size = 6, channel numbers 47, 19, 20, 23, 46, 16

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Table Size	Argument	
0xBB	0x00	0x31	0x00	0x07	0x06	0x2F	0x13
Argument				End Mark	CRC-16		
0x14	0x17	0x2E	0x10	0x7E	0xNNNN		

4.24.2 Response

Message Type: Response (0x01)
Code: Set Frequency Hopping Table (0x31)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x31	0x00	0x01	0x00	0x7E	0xNNNN

4.25 Get Modulation Mode

Get current modulation mode. The modulation mode is combination Rx modulation type and BLF

4.25.1 Command

Message Type: Command (0x00)
Code: Get Modulation Mode (0x32)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x32	0x00	0x00	0x7E	0xNNNN

4.25.2 Response

Message Type: Response (0x01)
Code: Get Modulation Mode (0x32)
Arguments
- BLF (16-bit): backscatter link frequency
- RxMod (8-bit): data rate and modulation format
- DR (8-bit): divide ratio

	BLF	RxMod	DR		BLF	RxMod	DR
40KHz, FM0, DR=8	0x0028	0x00	0x00	40KHz, M4, DR=8	0x0028	0x02	0x00
80KHz, FM0, DR=8	0x0050	0x00	0x00	80KHz, M4, DR=8	0x0050	0x02	0x00
160KHz, FM0, DR=64/3	0x00A0	0x00	0x01	160KHz, M4, DR=64/3	0x00A0	0x02	0x01
250KHz, FM0, DR=64/3	0x00FA	0x00	0x01	250KHz, M4, DR=64/3	0x00FA	0x02	0x01
320KHz, FM0, DR=64/3	0x0140	0x00	0x01	320KHz, M4, DR=64/3	0x0140	0x02	0x01
640KHz, FM0, DR=64/3	0x0280	0x00	0x01	640KHz, M4, DR=64/3	0x0280	0x02	0x01
40KHz, M2, DR=8	0x0028	0x01	0x00	40KHz, M8, DR=8	0x0028	0x03	0x00
80KHz, M2, DR=8	0x0050	0x01	0x00	80KHz, M8, DR=8	0x0050	0x03	0x00
160KHz, M2, DR=64/3	0x00A0	0x01	0x01	160KHz, M8, DR=64/3	0x00A0	0x03	0x01
250KHz, M2, DR=64/3	0x00FA	0x01	0x01	250KHz, M8, DR=64/3	0x00FA	0x03	0x01
320KHz, M2, DR=64/3	0x0140	0x01	0x01	320KHz, M8, DR=64/3	0x0140	0x03	0x01
640KHz, M2, DR=64/3	0x0280	0x01	0x01	640KHz, M8, DR=64/3	0x0280	0x03	0x01

Example) BLF = 250KHz, RxMod = M8, DR = 64/3

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	BLF (MSB)	BLF (LSB)	RxMod
0xBB	0x01	0x32	0x00	0x04	0x00	0xFA	0x03
DR	End Mark	CRC-16					
0x01	0x7E	0xNNNN					

4.26 Set Modulation Mode

Set current modulation mode. The modulation mode is combination Rx modulation type and BLF

4.26.1 Command

Message Type: Command (0x00)
Code: Set Modulation Mode (0x33)
Arguments
- BLF (16-bit), RxMod (8-bit), DR (8-bit):

	BLF	RxMod	DR		BLF	RxMod	DR
40KHz, FM0, DR=8	0x0028	0x00	0x00	40KHz, M4, DR=8	0x0028	0x02	0x00
80KHz, FM0, DR=8	0x0050	0x00	0x00	80KHz, M4, DR=8	0x0050	0x02	0x00
160KHz, FM0, DR=64/3	0x00A0	0x00	0x01	160KHz, M4, DR=64/3	0x00A0	0x02	0x01
250KHz, FM0, DR=64/3	0x00FA	0x00	0x01	250KHz, M4, DR=64/3	0x00FA	0x02	0x01
320KHz, FM0, DR=64/3	0x0140	0x00	0x01	320KHz, M4, DR=64/3	0x0140	0x02	0x01
640KHz, FM0, DR=64/3	0x0280	0x00	0x01	640KHz, M4, DR=64/3	0x0280	0x02	0x01
40KHz, M2, DR=8	0x0028	0x01	0x00	40KHz, M8, DR=8	0x0028	0x03	0x00
80KHz, M2, DR=8	0x0050	0x01	0x00	80KHz, M8, DR=8	0x0050	0x03	0x00
160KHz, M2, DR=64/3	0x00A0	0x01	0x01	160KHz, M8, DR=64/3	0x00A0	0x03	0x01
250KHz, M2, DR=64/3	0x00FA	0x01	0x01	250KHz, M8, DR=64/3	0x00FA	0x03	0x01
320KHz, M2, DR=64/3	0x0140	0x01	0x01	320KHz, M8, DR=64/3	0x0140	0x03	0x01
640KHz, M2, DR=64/3	0x0280	0x01	0x01	640KHz, M8, DR=64/3	0x0280	0x03	0x01

Example) Manual, BLF = 250KHz, RxMod = M8, DR = 64/3

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Mod Mode	BLF (MSB)	BLF (LSB)
0xBB	0x00	0x33	0x00	0x05	0xFF	0x00	0xFA
RxMod	DR	End Mark	CRC-16				
0x03	0x01	0x7E	0xNNNN				

4.26.2 Response

Message Type: Response (0x01)
Code: Set Modulation Mode (0x33)
Arguments
- None

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x33	0x00	0x01	0x00	0x7E	0xNNNN

4.27 Get Anti-Collision Mode

Get Anti-collision algorithm.

4.27.1 Command

Message Type: Command (0x00)
Code: Get Anti-Collision Mode (0x34)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x34	0x00	0x00	0x7E	0xNNNN

4.27.2 Response

Message Type: Response (0x01)
Code: Get Anti-Collision Mode (0x34)
Arguments
- Anti-collision Mode (8-bit): fixed Q(0x00), dynamic Q(0x01)
- Q Start (8-bit)
- Q Max (8-bit)
- Q Min (8-bit)

Example) Anti-collision Mode: dynamic Q, Q Start: 4, Q Max:7, Q Min: 2

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Mode	Q Start	Q Max
0xBB	0x01	0x34	0x00	0x04	0x01	0x04	0x07
Q Min	End Mark	CRC-16					
0x02	0x7E	0xNNNN					

[Notice] Refer to application note “Anti-Collision Mode for multi-tag” for more detail

4.28 Set Anti-Collision Mode

Set Anti-collision algorithm.

4.28.1 Command

Message Type: Command (0x00)
Code: Set Anti-Collision Mode (0x35)
Arguments
- Anti-collision Mode (8-bit): fixed Q(0x00), dynamic Q(0x01)
- Q Start (8-bit)
- Q Max (8-bit)
- Q Min (8-bit)

Example) Anti-collision Mode: dynamic Q, Q Start: 4, Q Max:7, Q Min: 2

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Mode	Q Start	Q Max
0xBB	0x00	0x35	0x00	0x04	0x01	0x04	0x07

Q Min	End Mark	CRC-16
0x02	0x7E	0xNNNN

4.28.2 Response

Message Type: Response (0x01)
Code: Set Anti-Collision Mode (0x35)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x35	0x00	0x01	0x00	0x7E	0xNNNN

[Notice] Refer to application note “Anti-Collision Mode for multi-tag” for more detail

4.29 Start Auto Read2

Start an automatic tag read operation, tag IDs are sent back to user though notification packet.

4.29.1 Command

Message Type: Command (0x00)
Code: Start Auto Read2 (0x36)
Arguments
- Reserve: type B tag (0x01), type C Tag (0x02)
- MTNU: maximum number of tag to read
- MTIME: maximum elapsed time to tagging (sec)
- RC (16-bit): Repeat cycle (how many times reader perform inventory round).

Example) MTNU = 0, MTIME = 0, Repeat Cycle = 100

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Reserve	MTNU	MTIME
0xBB	0x00	0x36	0x00	0x05	0x02	0x00	0x00
RC(MSB)	RC(LSB)	End Mark	CRC-16				
0x00	0x64	0x7E	0xNNNN				

4.29.2 Response

Message Type: Response (0x01)
Code: Start Auto Read2 (0x36)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x36	0x00	0x01	0x00	0x7E	0xNNNN

4.29.3 Notification

Message Type: Notification (0x02)
Code: Read Type C UII (0x22)
Arguments
- EPC Block (PC + EPC)

Example) PC = 0x3000, EPC = 0xE2003411B802011383258566

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	PC(MSB)	PC(LSB)	EPC (MSB)
0xBB	0x02	0x22	0x00	0x0E	0x30	0x00	0xE2
0x00	0x34	0x11	0xB8	0x02	0x01	0x13	0x83
		EPC (LSB)	End Mark	CRC-16			
0x25	0x85	0x66	0x7E	0xNNNN			

Message Type: Notification (0x02)
Code: Start Auto Read2 (0x36)
Arguments
- Read complete (0x1F)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x02	0x36	0x00	0x01	0x1F	0x7E	0xNNNN

4.30 Stop Auto Read2

Stop an automatic read2 operation.

4.30.1 Command

Message Type: Command (0x00)
Code: Stop Auto Read2 (0x37)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x37	0x00	0x00	0x7E	0xNNNN

4.30.2 Response

Message Type: Response (0x01)
Code: Stop Auto Read2 (0x37)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x37	0x00	0x01	0x00	0x7E	0xNNNN

4.31 Start Auto Read RSSI

Start an automatic tag read operation, tag IDs with RSSI are sent back to user though notification packet.

4.31.1 Command

Message Type: Command (0x00)
Code: Start Auto Read RSSI (0x38)
Arguments
- Reserve: type B tag (0x01), type C Tag (0x02)
- MTNU: maximum number of tag to read
- MTIME: maximum elapsed time to tagging (sec)
- RC (16-bit): Repeat cycle (how many times reader perform inventory round).

Example) MTNU = 0, MTIME = 0, Repeat Cycle = 100

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Reserve	MTNU	MTIME
0xBB	0x00	0x38	0x00	0x05	0x02	0x00	0x00
RC(MSB)	RC(LSB)	End Mark	CRC-16				
0x00	0x64	0x7E	0xNNNN				

4.31.2 Response

Message Type: Response (0x01)
Code: Start Auto Read RSSI (0x38)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x38	0x00	0x01	0x00	0x7E	0xNNNN

4.31.3 Notification

Message Type: Notification (0x02)
Code: Read Type C UII RSSI (0x23)
Arguments
- EPC Block (PC + EPC)
- Tag RSSI (32-bit): GAIN_I (8-bit), GAIN_Q (8-bit), RSS_I (8-bit), RSSI_Q (8-bit)

Example) PC = 0x3000, EPC = 0xE2003411B802011383258566, TAG RSSI = -41.2

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	PC(MSB)	PC(LSB)	EPC (MSB)
0xBB	0x02	0x23	0x00	0x12	0x30	0x00	0xE2
0x00	0x34	0x11	0xB8	0x02	0x01	0x13	0x83
		EPC (LSB)	RSSI_I	RSSI_Q	GAIN_I	GAIN_Q	End Mark
0x25	0x85	0x66	0x14	0x28	0x79	0x89	0x7E
CRC-16							
0xNNNN							

Note: Tag RSSI calculation

$$\text{RFIN\_I}' = 20\log_{10}(RSSI\_I) - GAIN\_I - 63$$
$$\text{RFIN\_Q}' = 20\log_{10}(RSSI\_Q) - GAIN\_Q - 63$$
$$\text{RFIN\_I}'' = 10^{(\frac{RFIN\_I'}{20})}$$
$$\text{RFIN\_Q}'' = 10^{(\frac{RFIN\_Q'}{20})}$$

$$\text{RFIN} = \sqrt{(RFIN\_I'')^2 + (RFIN\_Q'')^2}$$
$$\text{TAG\_RSSI} = 20\log_{10}(RFIN)$$

Message Type: Notification (0x02)
Code: Start Auto Read RSSI (0x38)
Arguments
- Read complete (0x1F)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x02	0x38	0x00	0x01	0x1F	0x7E	0xNNNN

4.32 Stop Auto Read RSSI

Stop an automatic read operation.

4.32.1 Command

Message Type: Command (0x00)
Code: Stop Auto Read RSSI (0x39)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0x39	0x00	0x00	0x7E	0xNNNN

4.32.2 Response

Message Type: Response (0x01)
Code: Stop Auto Read RSSI (0x39)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x39	0x00	0x01	0x00	0x7E	0xNNNN

4.33 Write Type C Tag Data

Write type C tag data.

4.33.1 Command

Message Type: Command (0x00)
Code: Write Type C User Data (0x46)
Arguments
- AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filled to 0x00000000.
- UL (16-bit): Target tag's EPC length
- EPC (variable): Target tag's EPC
- MB (8-bit): Target memory bank; 0x00 Reserved, 0x01 EPC, 0x02 TID, 0x03 User
- SA (16-bit): Starting Address word pointer
- DL (16-bit): Data Length to write (Word Count)
- DT (variable): Data to write

Example)
Access Password = 0x00000000, UL = 12 (0x0C), EPC = 0xE2003411B802011526370494,
Target memory bank = RFU, Start Address = 0x0000, Data Length = 4 word, Data to write = 0x1234567800000000

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP (MSB)		
0xBB	0x00	0x46	0x00	0x1F	0x00	0x00	0x00
AP (LSB)	UL (MSB)	UL (LSB)	EPC (MSB)				
0x00	0x00	0x0C	0xE2	0x00	0x34	0x11	0xB8
						EPC (LSB)	MB
0x02	0x01	0x15	0x26	0x37	0x04	0x94	0x00
SA (MSB)	SA (LSB)	DL (MSB)	DL (LSB)	DT (MSB)			
0x00	0x00	0x00	0x04	0x12	0x34	0x56	0x78
			DT (LSB)	End Mark	CRC-16		
0x00	0x00	0x00	0x00	0x7E	0xNNNN		

4.33.2 Response

Message Type: Response (0x01)
Code: Write Type C User Data (0x46)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x46	0x00	0x01	0x00	0x7E	0xNNNN

4.34 BlockWrite Type C Tag Data

Blockwrite type C tag data.

4.34.1 Command

Message Type: Command (0x00)
Code: BlockWrite Type C User Data (0x47)
Arguments
- AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filled to 0x00000000.
- UL (16-bit): Target tag's EPC length
- EPC (variable): Target tag's EPC
- MB (8-bit): Target memory bank; 0x00 Reserved, 0x01 EPC, 0x02 TID, 0x03 User
- SA (16-bit): Starting Address word pointer
- DL (16-bit): Data Length to write (Word Count)
- DT (variable): Data to write

Example)
Access Password = 0x00000000, UL = 12 (0x0C), EPC = 0xE2003411B802011526370494,
Target memory bank = RFU, Start Address = 0x0000, Data Length = 4 word, Data to write = 0x1234567800000000

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP (MSB)		
0xBB	0x00	0x47	0x00	0x1F	0x00	0x00	0x00
AP (LSB)	UL (MSB)	UL (LSB)	EPC (MSB)				
0x00	0x00	0x0C	0xE2	0x00	0x34	0x11	0xB8
						EPC (LSB)	MB
0x02	0x01	0x15	0x26	0x37	0x04	0x94	0x00
SA (MSB)	SA (LSB)	DL (MSB)	DL (LSB)	DT (MSB)			
0x00	0x00	0x00	0x04	0x12	0x34	0x56	0x78
			DT (LSB)	End Mark	CRC-16		
0x00	0x00	0x00	0x00	0x7E	0xNNNN		

4.34.2 Response

Message Type: Response (0x01)
Code: BlockWrite Type C User Data (0x47)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x47	0x00	0x01	0x00	0x7E	0xNNNN

4.35 BlockErase Type C Tag Data

Block erases type C tag data.

4.35.1 Command

Message Type: Command (0x00)
Code: BlockErase Type C Tag Data (0x48)
Arguments
- AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filled to 0x00000000.
- UL (16-bit): Target tag's EPC length
- EPC (variable): Target tag's EPC
- MB (8-bit): Target memory bank; 0x00 RFU, 0x01 EPC, 0x02 TID, 0x03 User
- SA (16-bit): Starting Address word pointer
- DL (16-bit): Data Length (Word Count)

Example)
Access Password = 0x00000000, UL = 12 (0x0C) byte, EPC = 0xE2003411B802011526370494,
Target memory bank = RFU, Start Address = 0x0000, Length = 4 word

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP (MSB)		
0xBB	0x00	0x48	0x00	0x17	0x00	0x00	0x00
AP (LSB)	UL (MSB)	UL (LSB)	EPC (MSB)				
0x00	0x00	0x0C	0xE2	0x00	0x34	0x11	0xB8
						EPC (LSB)	MB
0x02	0x01	0x15	0x26	0x37	0x04	0x94	0x00
SA (MSB)	SA (LSB)	DL (MSB)	DL (LSB)	End Mark	CRC-16		
0x00	0x00	0x00	0x04	0x7E	0xNNNN		

4.35.2 Response

Message Type: Response (0x01)
Code: BlockErase Type C Tag Data (0x48)
Arguments
- Success

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x48	0x00	0x01	0x00	0x7E	0xNNNN

4.36 BlockPermalock Type C Tag

BlockPermalock type C tag.

4.36.1 Command

Message Type: Command (0x00)
Code: BlockPermalock Type C Tag (0x83)
Arguments
- AP (32-bit): Access Password if target memory bank was password protected. Otherwise, set AP filled to 0x00000000.
- UL (16-bit): Target tag's EPC length
- EPC (variable): Target tag's EPC
- RFU (8-bit): 0x00
- R/L (8-bit): Read/Lock bit; 0x00 Read, 0x01 Permalock
- MB (8-bit): Target memory bank; 0x00 Reserved, 0x01 EPC, 0x02 TID, 0x03 User
- BP (16-bit): Mask starting address, specified in units of 16 blocks
- BR (8-bit): Mask range, specified in units of 16 blocks
- Mask (variable): Mask value

Example)
Access Password = 0x11111111, UL = 12 (0x0C), EPC = 0xE2003411B802011526370494, RFU = 0x00, Read/Lock bit = Lock (0x01), Target memory bank = User memory (0x03), Block Pointer = 0x0000, Block Range = 1, Mask value = 0xFFFF

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP (MSB)		
0xBB	0x00	0x83	0x00	0x0F	0x11	0x11	0x11
AP (LSB)	UL (MSB)	UL (LSB)	EPC (MSB)				
0x11	0x00	0x0C	0xE2	0x00	0x34	0x11	0xB8
						EPC (LSB)	RFU
0x02	0x01	0x15	0x26	0x37	0x04	0x94	0x00
R/L	MB	BP (MSB)	BP (LSB)	BR	Mask	Mask	End Mark
0x01	0x03	0x00	0x00	0x01	0xFF	0xFF	0x7E
CRC-16							
0xNNNN							

4.36.2 Response

Message Type: Response (0x01)
Code: BlockPermalock Type C Tag (0x83)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x83	0x00	0x01	0x00	0x7E	0xNNNN

4.37 Kill Type C Tag

Kill a Tag.

4.37.1 Command

Message Type: Command (0x00)
Code: Kill Type C Tag (0x65)
Arguments
- KP (32-bit): Kill Password. If KP filed set to 0x00000000, 'Kill Type C Tag' command do not work. The target tag ignores it.
- UL (16-bit): Target tag's EPC length
- EPC (variable): Target tag's EPC
- Recom (8-bit): Recommissioning bits

Example)
Kill Password =0x87654321, UL = 12 (0x0C) byte, EPC = 0xE2003411B802011526370494, Recom = 0x00

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	KP (MSB)		
0xBB	0x00	0x65	0x00	0x13	0x87	0x65	0x43
KP (LSB)	UL (MSB)	UL (LSB)	EPC (MSB)				
0x21	0x00	0x0C	0xE2	0x00	0x34	0x11	0xB8
						Recom	EPC (LSB)
0x02	0x01	0x15	0x26	0x37	0x04	0x00	0x94
End Mark	CRC-16						
0x7E	0xNNNN						

4.37.2 Response

Message Type: Response (0x01)
Code: Kill Type C Tag (0x65)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x65	0x00	0x01	0x00	0x7E	0xNNNN

4.38 Lock Type C Tag

Lock an indicated memory bank in the tag.

4.38.1 Command

Message Type: Command (0x00)
Code: Lock Type C Tag (0x82)
Arguments
- AP (32-bit): Access Password if memory bank was password protected. Otherwise, set AP filled to 0x00000000.
- UL (16-bit): Target tag's EPC length
- EPC (variable): Target tag's EPC
- LD (24-bit): Lock mask and action flags. Pad 4-bit zeros (dummy) to the left of 20-bit lock mask and associated action flags.

Example)
Access Password = 0x00000000, UL = 12(0x0C) byte, EPC = 0xE2003411B802011526370494, Lock mask and action flags = 0x080200 {Binary: 0000 (dummy) + 1000000000 (mask) + 1000000000 (lock data)}

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP (MSB)		
0xBB	0x00	0x82	0x00	0x15	0x00	0x00	0x00
AP (LSB)	UL (MSB)	UL (LSB)	EPC (MSB)				
0x00	0x00	0x0C	0xE2	0x00	0x34	0x11	0xB8
						EPC (LSB)	LD (MSB)
0x02	0x01	0x15	0x26	0x37	0x04	0x94	0x08
	LD (LSB)	End Mark	CRC-16				
0x02	0x00	0x7E	0xNNNN				

4.38.2 Response

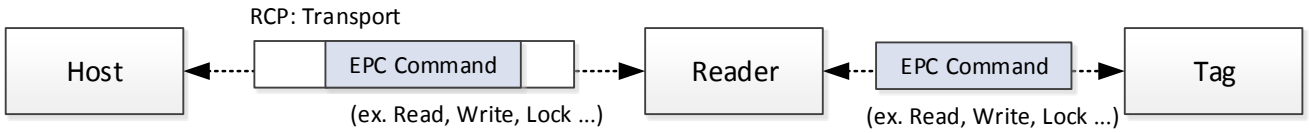
Message Type: Response (0x01)
Code: Lock Type C Tag (0x82)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x82	0x00	0x01	0x00	0x7E	0xNNNN

4.39 Generic Transport

Transport command is used to send a generic EPC command and receive a generic EPC response.



The generic EPC command is wrapped in payload filed of Transport command. Reader extracts only generic EPC command from received transport command and transmits it to Tag.

4.39.1 Command

Message Type: Command (0x00)
Code: Generic Transport Command (0x4D)
Arguments
- TS (8-bit): Transmission state defines which EPC state the readers transmits the command in.
This parameter shall also contain a flag to append the current EPC handle and calculate the CRC16 to the generic command payload.

	RN16/Handle	CRC	Transmission State
# of bits	1	1	6
Description	0: No RN16/handle 1: append RN16/handle	0: No CRC 1: append CRC	0: No Action 1: Select 2: Query 3: Reply 4: Acknowledged 5: Open 6: Secured (include Access)

- AP (32-bit): Access password. Set to 0x00000000 when the Generic command is to be transmitted in the EPC Secure state.
- RM (8-bit): RM specifies the EPC encoding and bit rate of the response. 0 value means to use current communication settings. Otherwise, RM value can be encoded 0001ammt. Where a is the DR bit, mm are M bits, and t is TRext bit.
- UL (16-bit): Target tag's EPC length
- EPC (variable): Target tag's EPC
- SZ (16-bit): Generic command length in bits (excluding the EPC handle and CRC16)
- GC (variable): Generic command payload (length equal to SZ divided by 8 rounded up)

Example) TS = 0xC3 (append RN16and CRC, Reply), Access Password = 0x00000000, RM = 0x00(current setting), UL = 12 (0x0C), EPC = 0x0080B0403C000000120A8A67,
SZ = 18 (0x0012), GC = 0xE00100 (GetSensorData Command)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	TS	AP (MSB)	
0xBB	0x00	0x4D	0x00	0x19	0xC3	0x00	0x00
	AP (LSB)	RM	UL (MSB)	UL (LSB)	EPC (MSB)		
0xBB	0x00	0x00	0x00	0x0C	0x00	0x08	0xB0
0x40	0x3C	0x00	0x00	0x00	0x12	0x0A	0x8A
EPC (LSB)	SZ (MSB)	SZ (LSB)	GC (MSB)		GC (LSB)	End Mark	CRC-16
0x67	0x00	0x12	0xE0	0x01	0x00	0x7E	0xNNNN

4.39.2 Response

Message Type: Response (0x01)
Code: Generic Transport Command (0x4D)
Arguments

- SZ (16-bit): Generic response length in bits (including the header, handle and CRC)
- GR (variable): Generic response contents (length equal to SZ divided by 8 rounded up)

Example) SZ = 0x0061(97 bits), GR = 0x20800000000000000DF4C1BECD

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	SZ(MSB)	SZ(LSB)	GR (MSB)
0xBB	0x01	0x4D	0x00	0x08	0x00	0x61	0x20
0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x0D
			GR (LSB)	End Mark	CRC-16		
0xF4	0xC1	0BE	0xCD	0x7E	0xNNNN		

4.40 Get Temperature

Get current temperature

4.40.1 Command

Message Type: Command (0x00)
Code: Get Temperature (0xB7)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0xB7	0x00	0x00	0x7E	0xNNNN

4.40.2 Response

Message Type: Response (0x01)
Code: Get Temperature (0xB7)
Arguments
- Temp (8-bit): Current temperature

Example) 24 °C

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Temp	End Mark	CRC-16
0xBB	0x01	0xB7	0x00	0x01	0x18	0x7E	0xNNNN

4.41 Get RSSI

Get RSSI level

4.41.1 Command

Message Type: Command (0x00)
Code: Get RSSI level (0xC5)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0xC5	0x00	0x00	0x7E	0xNNNN

4.41.2 Response

Message Type: Response (0x01)
Code: Get RSSI level (0xC5)
Arguments
- RSSI (16-bit): RSSI level (-dBm x 10, decimal value)

Example) RSSI = 900 (-90.0 dBm)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	RSSI (MSB)	RSSI (LSB)	End Mark
0xBB	0x01	0xC5	0x00	0x02	0x03	0x84	0x7E

CRC-16
0xNNNN

4.42 Scan RSSI

Scan RSSI level on all channels

4.42.1 Command

Message Type: Command (0x00)
Code: Scan RSSI (0xC6)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0xC6	0x00	0x00	0x7E	0xNNNN

4.42.2 Response

Message Type: Response (0x01)
Code: Scans RSSI (0xC6)
Arguments
- CHS (8-bit): Start channel number
- CHE (8-bit): Stop channel number
- CHB (8-bit): Best channel (lowest RSSI channel)
- RSSI1 (8-bit): RSSI level on CHS (-dBm)
- RSSI2 (8-bit): RSSI level on CHS + 1 (-dBm)
 ….
- RSSI[N] (8-bit): RSSI level on CHE (-dBm)

Example) CHS = 7, CHE = 20, CHB = 7, RSSI0 = 90 (-90dBm), RSSI1 = 87 (-87), …

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	CHS	CHE	CHB
0xBB	0x01	0xC6	0x00	0x11	0x07	0x14	0x07
RSSI1	RSSI2	RSSI3	...	RSSI13	RSSI14	End Mark	CRC-16
5A	57	57		5A	5A	0x7E	0xNNNN

4.43 Get DTC Result

Scan RSSI level on all channels

4.43.1 Command

Message Type: Command (0x00)
Code: Get DTC Result (0xCA)
Arguments
- None

Example)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0xCA	0x00	0x00	0x7E	0xNNNN

4.43.2 Response

Message Type: Response (0x01)
Code: Get DTC Result (0xCA)
Arguments
- IDT(8-bit): inductor number for digital tune
- DTC1(8-bit): digital tunable capacitor 1
- DTC2(8-bit): digital tunable capacitor 2
- RSSI(8-bit): leakage RSSI value to check leakage cancellation results
- State(8-bit): state number of leakage cancellation algorithm

Example) Example) IDT = 2, DTC1 = 23, DTC2 = 1, RSSI = 38, State = 1

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	IDT	DTC1	DTC2
0xBB	0x01	0xCA	0x00	0x05	0x02	0x17	0x01
RSSI	State	End Mark	CRC-16				
0x26	0x01	0x7E	0xNNNN				

4.43.3 Notification

Message Type: Notification (0x02)
Code: Get DTC Result (0xCA)
Arguments
- IDT(8-bit): inductor number for digital tune
- DTC1(8-bit): digital tunable capacitor 1
- DTC2(8-bit): digital tunable capacitor 2
- RSSI(8-bit): leakage RSSI value to check leakage cancellation results
- State(8-bit): state number of leakage cancellation algorithm
- CC(8-bit): Current Channel

Example) IDT = 2, DTC1 = 23, DTC2 = 1, RSSI = 38, State = 1, Current Channel = 10

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	IDT	DTC1	DTC2
0xBB	0x02	0xCA	0x00	0x06	0x02	0x17	0x01
RSSI	State	CC	End Mark	CRC-16			
0x26	0x01	0x0A	0x7E	0xNNNN			

4.44 Update Registry

Sets Registry Update function

4.44.1 Command

Message Type: Command (0x00)
Code: Update Registry (0xD2)
Arguments
- Arg (8-bit): Store (0x01)

Example) Store data into Registry

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0xD2	0x00	0x01	0x01	0x7E	0xNNNN

4.44.2 Response

Message Type: Response (0x01)
Code Update Registry (0xD2)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xD2	0x00	0x01	0x00	0x7E	0xNNNN

4.45 Get Registry Item

Gets Registry items

4.45.1 Command

Message Type: Command (0x00)
Code: Get Registry Item (0xD4)
Arguments
- Registry Version (0x0000)
- Firmware Date (0x0001)
- Band (0x0002)
- Tx power (0x0003)
- FH/LBT (0x0004)
- Anti-collision Mode (0x0005)
- Modulation Mode (0x0006)
- Query(Q) (0x0007)
- Frequency Hopping Table (0x0008)
- Tx Power Table (0x0009)

Example) Get Registry version

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	ADD (MSB)	ADD (LSB)	End Mark
0xBB	0x00	0xD4	0x00	0x02	0x00	0x00	0x7E
CRC-16							
0xNNNN							

4.45.2 Response

Message Type: Response (0x01)
Code: Get Registry Item (0xD4)
Arguments
- Active (8-bit): Registry items status; Inactive (0x00), Read-Only (0xBC), Active (0xA5)
- Data (Variable)

Example) Registry Version = 1

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Active	Data	End Mark
0xBB	0x01	0xD4	0x00	0x02	0x00	0x01	0x7E
CRC-16							
0xNNNN							

4.46 Set Optimum Frequency Hopping Table

Set Optimum Frequency Hopping Table.
When the reader's antenna size is not large enough, you cannot use all channel of your band.
If you read tag's information at the channel outside antenna bandwidth, the read range will be quite decreased
This command help you search good channels within your band and set optimized frequency hopping table.
When you execute this command, reader find optimized channel automatically.

4.46.1 Command

Message Type: Command (0x00)
Code: Set Optimum Frequency Hopping Table (0xE4)
Arguments
- None

Example) Set Optimum Frequency Hopping Table

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0xE4	0x00	0x00	0x7E	0xNNNN

4.46.2 Response

Message Type: Response (0x01)
Code: Set Optimum Frequency Hopping Table (0xE4)
Arguments
- Start (0x00), Finish (0x01)

Example) Start

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xE4	0x00	0x01	0x00	0x7E	0xNNNN

Example) Finish

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xE4	0x00	0x01	0x01	0x7E	0xNNNN

4.47 Get Frequency Hopping Mode

Get Frequency Hopping Mode
Reader can set two types of Frequency hopping table: normal mode and SH (Smart hopping) mode.

Reader use all frequency channel of your operation band in normal mode.
In SH (Smart Hopping) mode, you use the specified frequency hopping table selected by “Set Optimum Frequency Hopping Table” command.

4.47.1 Command

Message Type: Command (0x00)
Code: Get Frequency Hopping Mode (0xE5)
Arguments
- None

Example) Get Frequency Hopping Mode

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0xE5	0x00	0x00	0x7E	0xNNNN

4.47.2 Response

Message Type: Response (0x01)
Code: Get Frequency Hopping Mode (0xE5)
Arguments
- Frequency Hopping Mode (0x00: Normal Mode, 0x01: Smart Hopping Mode)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xE5	0x00	0x01	0x00	0x7E	0xNNNN

4.48 Set Frequency Hopping Mode

Set Frequency hopping mode
Normal mode use all the frequency channel of your band for frequency hopping
Smart hopping mode use the hopping table selected by “Set Optimum Frequency Hopping Table”.

4.48.1 Command

Message Type: Response (0x00)
Code: Set Frequency Hopping Mode (0xE6)
Arguments

- Frequency Hopping Mode (0x00: Normal Mode, 0x01: Smart Hopping Mode)

Example) Set Normal Mode

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0xE6	0x00	0x01	0x00	0x7E	0xNNNN

4.48.2 Response

Message Type: Response (0x01)
Code: Set Frequency Hopping Mode (0xE6)
Arguments
- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xE6	0x00	0x01	0x00	0x7E	0xNNNN

4.49 Get Tx Leakage RSSI Level for Smart hopping Mode

For Smart hopping mode, reference value of Tx Leakage RSSI is needed to select the good channel according to antenna while “Set Optimum Frequency Hopping Table”.

4.49.1 Command

Message Type: Response (0x00)
Code: Get Tx Leakage RSSI level for smart hopping mode (0xE7)
Arguments
-None

Example) Get tx leakage RSSI level for Smart hopping mode

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0xE7	0x00	0x00	0x7E	0xNNNN

4.49.2 Response

Message Type: Response (0x01)
Code: Get Tx Leakage RSSI level for Smart hopping Mode (0xE7)
Arguments
-Reference Tx Leakage RSSI Level (1~255)

Example) Reference Tx leakage RSSI Level = 50

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xE7	0x00	0x01	0x32	0x7E	0xNNNN

4.50 Set Tx Leakage RSSI Level for Smart hopping Mode

Set Tx Leakage RSSI Level for Smart hopping mode.
This value is reference level to select channel for Smart hopping mode.
If this value is too small, the number of channel can be used may be reduced. If you want to use more channels with Smart hopping mode, increase this value. Default value is 50.

4.50.1 Command

Message Type: Response (0x00)
Code: Set Tx Leakage RSSI level for smart hopping mode (0xE8)
Arguments
-Reference Tx Leakage RSSI Level (1~255)

Example) Set Reference Tx leakage RSSI Level to 50

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0xE8	0x00	0x01	0x32	0x7E	0xNNNN

4.50.2 Response

Message Type: Response (0x01)
Code: Set Tx Leakage RSSI level for smart hopping mode (0xE8)
Arguments
-Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xE8	0x00	0x01	0x00	0x7E	0xNNNN

4.51 Command failure

Response to invalid command

Message Type: Response (0x01)
Code: Command failure (0xFF)
Arguments
- Error code (8-bit)

Error Code	Description
0x01	Failure of Reader power control
0x02	Failure of Reader control
0x03	Failure to get the Reader Information
0x07	Failure to get region
0x08	Failure to set region
0x09	Failure to read the tag memory
0x0A	Failure of automatic read operation
0x0B	Automatic read in operation
0x0C	Cannot stop automatic read
0x0D	Not in automatic read mode
0x0E	Invalid parameter
0x10	Failure to write data
0x11	Failure to erase data
0x12	Failure to kill a tag
0x13	Failure to lock a tag
0x15	No tag detected
0x17	Not supported command
0x18	Undefined command
0x19	Failure to reset Reader
0xFF	CRC error

- Command code (8-bit)
- Sub Error Code (8-bit)

Category	Error Code	Description
0x00 – 0x0F: EPC G2v2 Error Message	0x01	Not supported
	0x02	Insufficient privileges
	0x03	Memory overrun
	0x04	Memory locked
	0x05	Crypto suite error
	0x06	Command not encapsulated
	0x07	ResponseBuffer overflow
	0x08	Security timeout
	0x0B	Insufficient power
	0x0F	Non-specific error
0x10 – 0x7F: Vendor Specific Error	0x11	Sensor Scheduling configuration
	0x12	Tag Busy
	0x13	Measurement type not supported
0x80 – 0x8F: Protocol Error	0x80	No tag detected
	0x81	Handle acquisition failure
	0x82	Access password failure
0x90 – 0x9F: Modem Error	0x90	CRC error
	0x91	Rx Timeout
0xA0 – 0xAF: Registry	0xA0	Registry update failure
	0xA1	Registry erase failure
	0xA2	Registry write failure
	0xA3	Registry not exist
0xB0 – 0xBF: Peripheral	0xB0	UART failure
	0xB1	SPI failure
	0xB2	I2C failure
	0xB3	GPIO failure
0xC0 – 0xDF: Reserved		

0xE0 – 0xFF: Custom Error	0xE0	Not supported command
	0xE1	Undefined command
	0xE2	Invalid parameter
	0xE3	Too high parameter
	0xE4	Too low parameter
	0xE5	Failure automatic read operation
	0xE6	Not automatic read mode
	0xE7	Failure to get last response
	0xE8	Failure to control test
	0xE9	Failure to reset Reader
	0xEA	Rfidblock control failure

Example) Invalid parameter

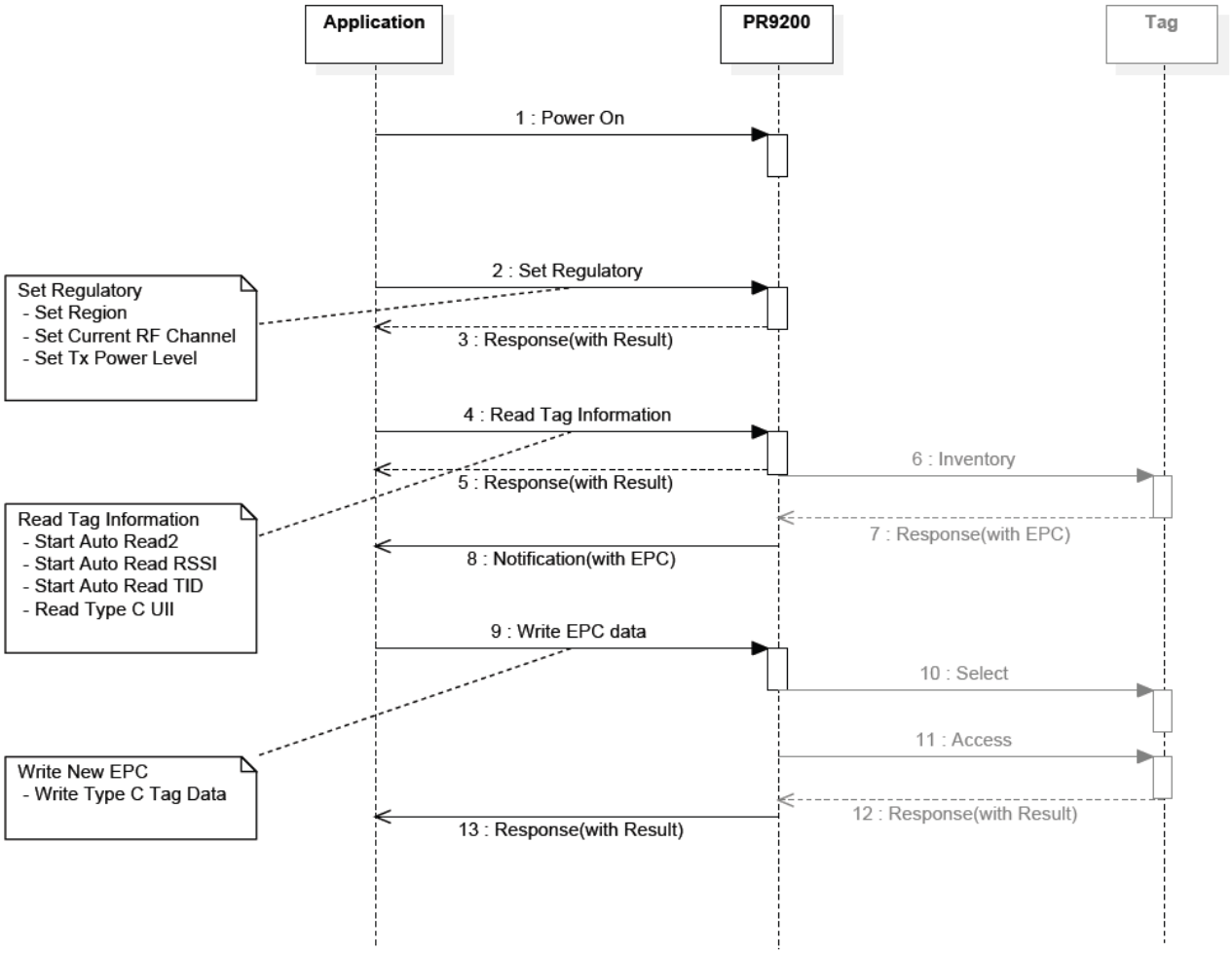
Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Error Code	Cmd. Code	Sub Error Code
0xBB	0x01	0xFF	0x00	0x01	0x0E	0x07	0xE2
End Mark	CRC-16						
0x7E	0xNNNN						

5 Use Case

5.1 Change to the new EPC

Replace it with a new EPC.

5.1.1 Command Sequence



5.1.2 Command Example

[Command] Set Region
Region = US

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x07	0x00	0x01	0x21	0x7E	0xNNNN

[Response] Set Region
Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x07	0x00	0x01	0x00	0x7E	0xNNNN

[Command] Set Current RF Channel
Channel Number = 10, Channel Number Offset = 0

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	CN	CNO	End Mark

0xBB	0x00	0x12	0x00	0x02	0x0A	0x00	0x7E
CRC-16							
0xNNNN							

[Response] Set Current RF Channel
Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x12	0x00	0x01	0x00	0x7E	0xNNNN

[Command] Set Tx Power Level
PWR = 200 (20.0 dBm)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	PWR (MSB)	PWR (LSB)	End Mark
0xBB	0x00	0x16	0x00	0x02	0x00	0xC8	0x7E
CRC-16							
0xNNNN							

[Response] Set Tx Power Level
Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x16	0x00	0x01	0x00	0x7E	0xNNNN

[Command] Start Auto Read 2
MTNU = 0, MTIME = 0, Repeat Cycle = 100

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Reserve	MTNU	MTIME
0xBB	0x00	0x36	0x00	0x05	0x02	0x00	0x00
RC(MSB)	RC(LSB)	End Mark	CRC-16				
0x00	0x64	0x7E	0xNNNN				

[Response] Start Auto Read 2
Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x36	0x00	0x01	0x00	0x7E	0xNNNN

[Notification] Start Auto Read 2
PC = 0x3000, EPC = 0xE2003411B802011383258566

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	PC(MSB)	PC(LSB)	EPC (MSB)
0xBB	0x02	0x22	0x00	0x0E	0x30	0x00	0xE2
0x00	0x34	0x11	0xB8	0x02	0x01	0x13	0x83
		EPC (LSB)	End Mark	CRC-16			
0x25	0x85	0x66	0x7E	0xNNNN			

[Command] Write Type C Tag Data
Access Password = 0x00000000, UL = 12 (0x0C), EPC = 0xE2003411B802011383258566,

Target memory bank = EPC, Start Address = 0x0002, Data Length = 6 word,
Data to write = 0xE2003411B802011383258566

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	AP (MSB)		
0xBB	0x00	0x46	0x00	0x23	0x00	0x00	0x00
AP (LSB)	UL (MSB)	UL (LSB)	EPC (MSB)				
0x00	0x00	0x0C	0xE2	0x00	0x34	0x11	0xB8
						EPC (LSB)	MB
0x02	0x01	0x13	0x83	0x25	0x85	0x66	0x01
SA (MSB)	SA (LSB)	DL (MSB)	DL (LSB)	DT (MSB)			
0x00	0x02	0x00	0x06	0xE2	0x00	0x34	0x11
							DT (LSB)
0xB8	0x02	0x01	0x13	0x83	0x25	0x85	0x77
End Mark	CRC-16						
0x7E	0xNNNN						

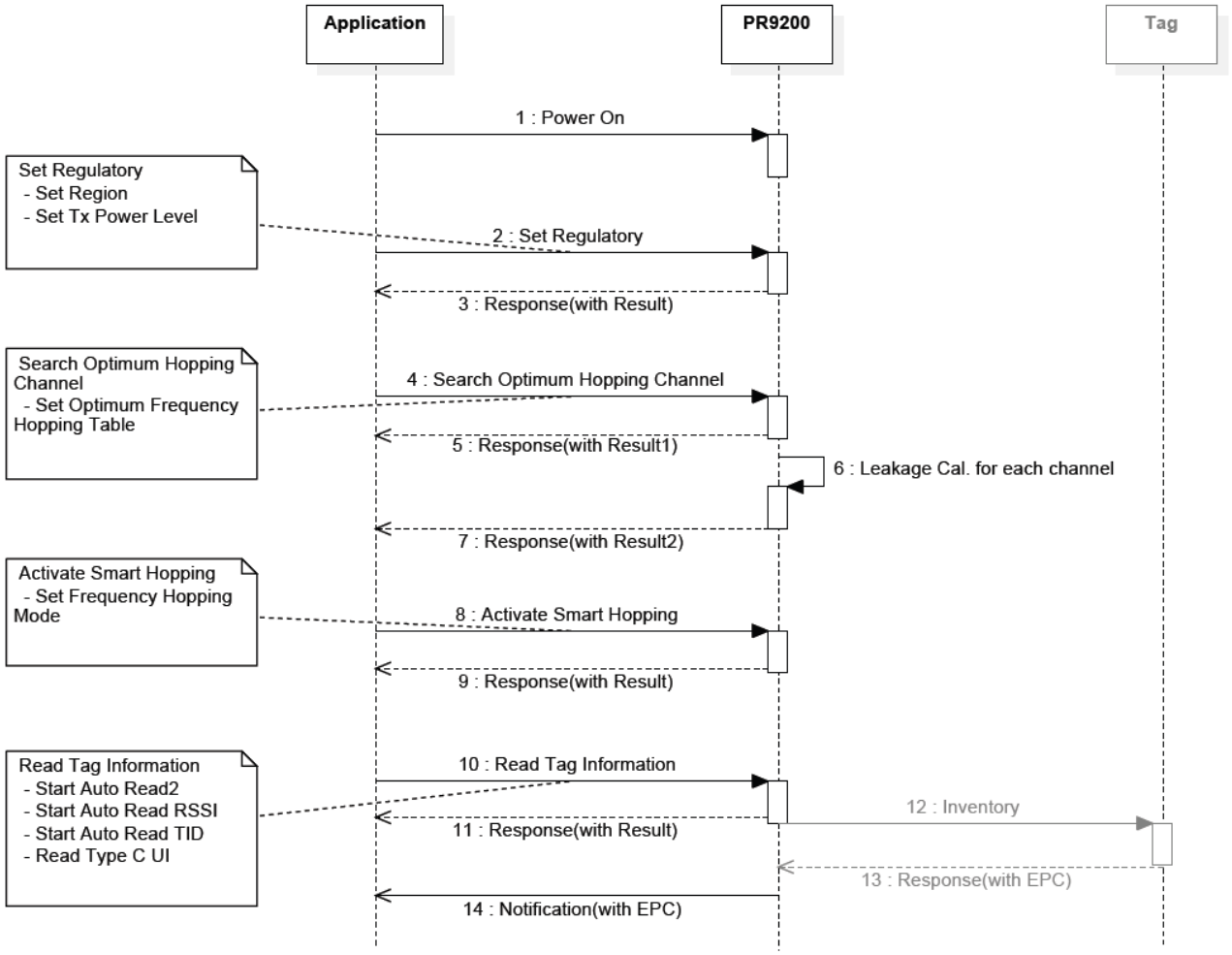
[Response] Write Type C Tag Data
Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x46	0x00	0x01	0x00	0x7E	0xNNNN

5.2 Activate the Smart-hopping Table

Use a good channel selected by the RSSI value in the entire channel.

5.2.1 Command Sequence



5.2.2 Command Example

[Command] Set Region
Region = US

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0x07	0x00	0x01	0x21	0x7E	0xNNNN

[Response] Set Region
Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x07	0x00	0x01	0x00	0x7E	0xNNNN

[Command] Set Tx Power Level
PWR = 200 (20.0 dBm)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	PWR (MSB)	PWR (LSB)	End Mark
0xBB	0x00	0x16	0x00	0x02	0x00	0xC8	0x7E
CRC-16							

0xNNNN

[Response] Set Tx Power Level
Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x16	0x00	0x01	0x00	0x7E	0xNNNN

[Command] Set Optimum Frequency Hopping Table

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	End Mark	CRC-16
0xBB	0x00	0xE4	0x00	0x00	0x7E	0xNNNN

[Response] Set Optimum Frequency Hopping Table
Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xE4	0x00	0x01	0x00	0x7E	0xNNNN

[Command] Set Frequency Hopping Mode
Smart Hopping Mode

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0xE6	0x00	0x01	0x01	0x7E	0xNNNN

[Response] Set Frequency Hopping Mode
Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xE6	0x00	0x01	0x00	0x7E	0xNNNN

[Command] Start Auto Read 2
MTNU = 0, MTIME = 0, Repeat Cycle = 100

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Reserve	MTNU	MTIME
0xBB	0x00	0x36	0x00	0x05	0x02	0x00	0x00
RC(MSB)	RC(LSB)	End Mark	CRC-16				
0x00	0x64	0x7E	0xNNNN				

[Response] Start Auto Read 2
Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0x36	0x00	0x01	0x00	0x7E	0xNNNN

[Notification] Start Auto Read 2
PC = 0x3000, EPC = 0xE2003411B802011383258566

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	PC(MSB)	PC(LSB)	EPC (MSB)
0xBB	0x02	0x22	0x00	0x0E	0x30	0x00	0xE2
0x00	0x34	0x11	0xB8	0x02	0x01	0x13	0x83
		EPC (LSB)	End Mark	CRC-16			

0x25	0x85	0x66	0x7E	0xNNNN

6 References

- ISO/IEC 18000-6 "Information technology - Radio frequency identification (RFID) for item management - Part6: Parameters for air interface communications at 860MHz to 960MHz"
- EPC™ "Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz"
- TTA, MRFS-1-06-R1-v1.0,”Mobile RFID Reader Control Protocol”

6 Address Information

PHYCHIPS Inc.
#104, 187 Techno 2-ro, Yuseong-gu, Daejeon, Korea (Yongsan-dong, Migun Technoworld 2)
http://www.phychips.com
sales@phychips.com
+82-42-864-2402
+82-42-864-2403