



## SDK Manual

ARETE POP

Smart RFID Dongle Reader



## Notice

### Registered Trademark



ARETE mobile is launched by PHYCHIPS Inc. for mobile device and currently registered worldwide with its ARETE mobile trade mark legally protected by law.

© 2013 PHYCHIPS Inc. All rights reserved.

This User Guide is produced by PHYCHIPS Inc. and protected by Copyright Laws.  
Please note that there may be possible mistake or omission of information in this document.

### Applicable Device

This document is applicable for iOS 6.0 and Android 2.3.3 above

## Contents

<b>1</b>	<b>iPhone Library.....</b>	<b>7</b>
1.1	Add libAreteAudio .....	7
1.2	APIs .....	10
1.2.1	Overview .....	10
1.2.2	AudioMgr.....	10
1.2.3	Rcp Api.....	11
<b>2</b>	<b>Android Library.....</b>	<b>23</b>
2.1	Add libAreteAudio .....	23
2.2	APIs .....	26
2.2.1	Overview .....	26
2.2.2	Rcp Api.....	26
<b>3</b>	<b>Reader Control Protocol .....</b>	<b>37</b>
3.1	RFID Reader Control Protocol overview .....	37
3.1.1	Preamble and End Mark field.....	37
3.1.2	Header Field.....	37
3.1.3	Payload Field .....	39
3.1.4	Cyclic Redundancy Check(CRC) Field.....	39
3.2	Get Reader Information.....	40
3.2.1	Command.....	40
3.2.2	Response .....	40
3.3	Get Region .....	41
3.3.1	Command.....	41
3.3.2	Response .....	41
3.4	Set Region.....	42
3.4.1	Command.....	42
3.4.2	Response .....	42
3.5	Get Type C A/I Select Parameters.....	43
3.5.1	Command.....	43
3.5.2	Response .....	43
3.6	Set Type C A/I Select Parameters .....	44
3.6.1	Command.....	44
3.6.2	Response .....	44

3.7	Get Type C A/I Query Parameters .....	45
3.7.1	Command .....	45
3.7.2	Response .....	45
3.8	Set Type C A/I Query Parameters .....	46
3.8.1	Command .....	46
3.8.2	Response .....	46
3.9	Get current RF Channel .....	47
3.9.1	Command .....	47
3.9.2	Response .....	47
3.10	Set current RF Channel .....	48
3.10.1	Command .....	48
3.10.2	Response .....	48
3.11	Get FH and LBT Parameters .....	49
3.11.1	Command .....	49
3.11.2	Response .....	49
3.12	Set FH and LBT Parameters .....	50
3.12.1	Command .....	50
3.12.2	Response .....	50
3.13	Get Tx Power Level .....	51
3.13.1	Command .....	51
3.13.2	Response .....	51
3.14	Set Tx Power Level .....	52
3.14.1	Command .....	52
3.14.2	Response .....	52
3.15	RF CW signal control .....	53
3.15.1	Command .....	53
3.15.2	Response .....	53
3.16	Read Type C Tag Data .....	54
3.16.1	Command .....	54
3.16.2	Response .....	54
3.17	Get Frequency Hopping Table .....	55
3.17.1	Command .....	55
3.17.2	Response .....	55
3.18	Set Frequency Hopping Table .....	56
3.18.1	Command .....	56

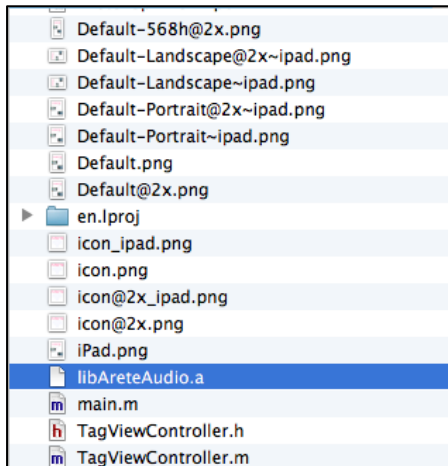
3.18.2	Response .....	56
3.20	Start Auto Read2.....	57
3.20.1	Command.....	57
3.20.2	Response .....	57
3.20.3	Notification.....	57
3.21	Stop Auto Read2.....	59
3.21.1	Command.....	59
3.21.2	Response .....	59
3.22	Write Type C Tag Data .....	60
3.22.1	Command.....	60
3.22.2	Response .....	60
3.23	Kill Type C Tag .....	61
3.23.1	Command.....	61
3.23.2	Response .....	61
3.24	Lock Type C Tag .....	62
3.24.1	Command.....	62
3.24.2	Response .....	62
3.25	Set Beep On .....	63
3.25.1	Command.....	63
3.25.2	Response .....	63
3.26	Get Temperature .....	64
3.26.1	Command.....	64
3.26.2	Response .....	64
3.27	Get RSSI .....	65
3.27.1	Command.....	65
3.27.2	Response .....	65
3.28	Update Registry .....	66
3.28.1	Command.....	66
3.28.2	Response .....	66
3.29	Erase Registry .....	67
3.29.1	Command.....	67
3.29.2	Response .....	67
3.30	Get Registry Item.....	68
3.30.1	Command.....	68
3.30.2	Response .....	68

3.31	Get ADC value.....	69
3.31.1	Command.....	69
3.31.2	Response .....	69
3.32	Command failure .....	70
4	Customer Service .....	71

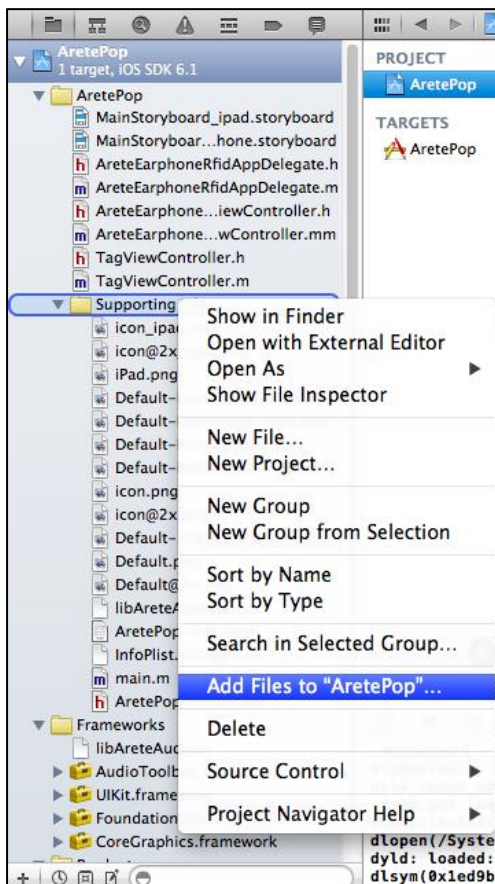
## 1 iPhone Library

### 1.1 Add libAreteAudio

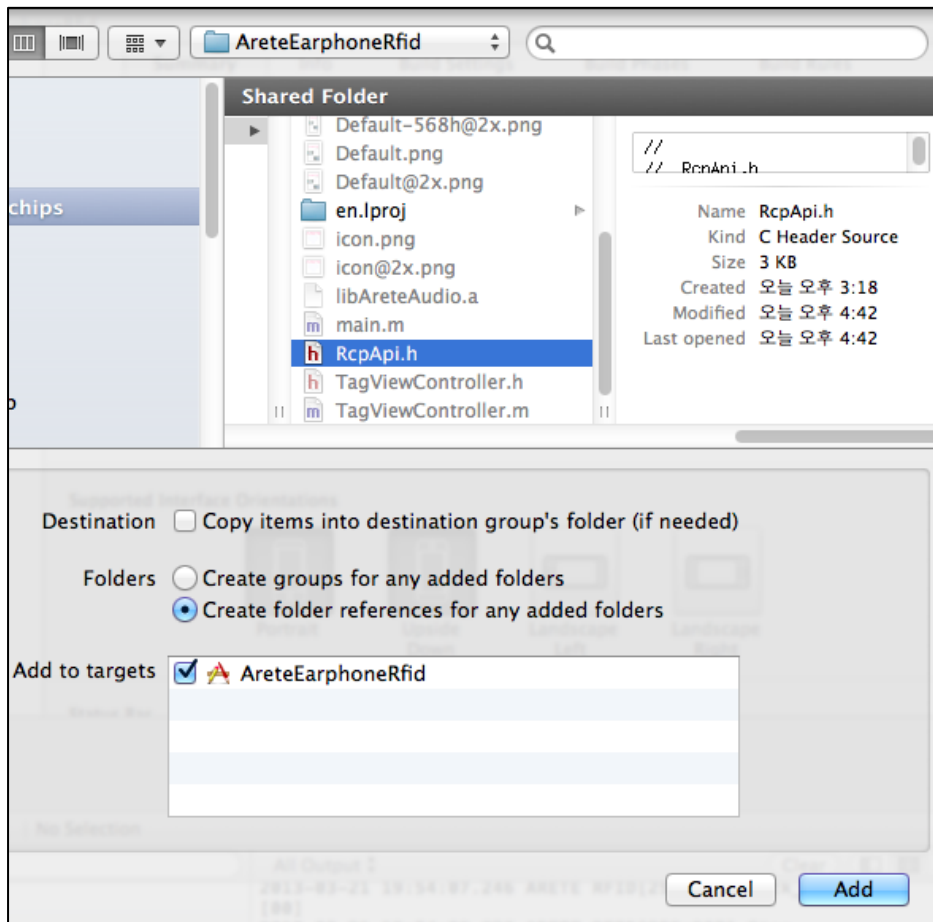
Extract the library zip file to the folder where the project has been created.



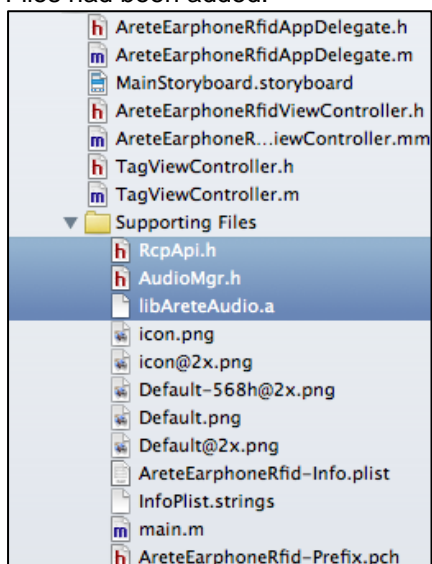
Select add files to your project.



Select library files and click add.



Files had been added.

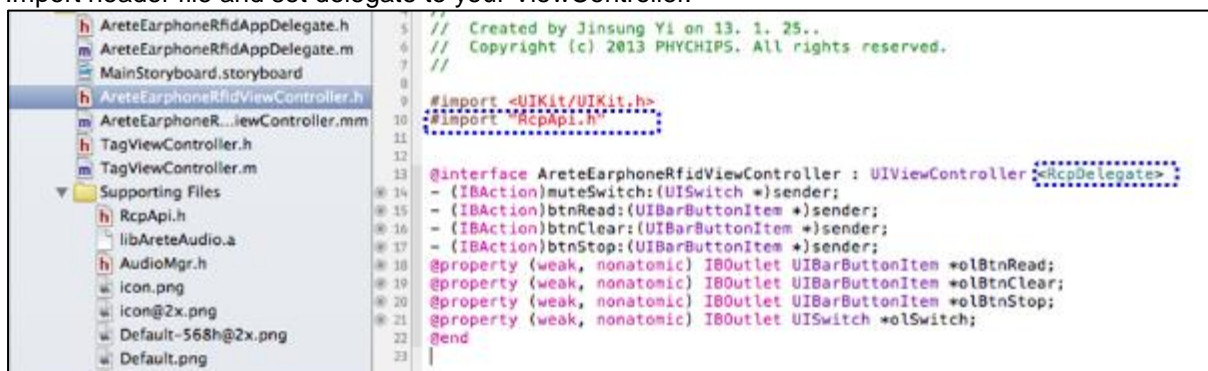




Add library file to Linked Frameworks and Libraries.



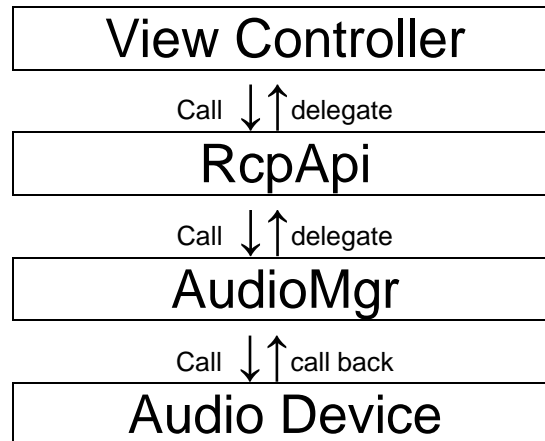
Import header file and set delegate to your ViewController.



## 1.2 APIs

### 1.2.1 Overview

API classes are layered as shown below. AudioMgr is converting Byte Array to Audio Signal, or vice versa. RcpApi generates RCP Packet Byte Array



### 1.2.2 AudioMgr

- Inherits  
@interface AudioMgr : NSObject
- Property  
@property (nonatomic, weak) id<AudioMgrDelegate> delegate;
- Delegate  
@protocol AudioMgrDelegate <NSObject>  
- (int)receive:(NSData \*)data;  
- (void) plugStatusChanged:(NSInteger)status;  
@end
- Instance Method

Method	- (id)init
Description	Returns a newly initialized API.
Parameters	None
Return Value	A newly initialized API object.

Method	- (BOOL)open
Description	Returns a Boolean value that indicating whether audio device is initialized.
Parameters	None
Return Value	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)isOpened</b>
<b>Description</b>	Returns a Boolean value that indicating whether audio device is initialized.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Open NO: Close

<b>Method</b>	<b>- (void)close</b>
<b>Description</b>	Close audio device.
<b>Parameters</b>	None
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)send:(NSData*)data</b>
<b>Description</b>	Returns a Boolean value that indicating whether :(NSData*)data is forwarded Audio Device.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@required <b>- (int)receive:(NSData *)data;</b>
<b>Description</b>	You can override this method to perform additional tasks associated with processing raw RCP packet.
<b>Parameters</b>	Byte Array: RCP packet
<b>Return Value</b>	Always return 0

<b>Method</b>	@required <b>- (void) plugStatusChanged:(NSInteger)status;</b>
<b>Description</b>	You can override this method to perform additional tasks associated with getting headset plug status.
<b>Parameters</b>	(NSInteger)status: Unplugged(0), Plugged(1)
<b>Return Value</b>	None

### 1.2.3 Rcp Api

- Inherits  
@interface RcpApi : NSObject <AudioMgrDelegate>
- Property  
@property (nonatomic, assign) BOOL isConnected;  
@property (nonatomic, weak) id<RcpDelegate> delegate;
- Delegate  
@optional  
- (void)pcEpcReceived:(NSData \*)pcEpc;

- (void)plugged:(BOOL)plug;
- (void)readerConnected;
- (void)ackReceived:(uint8\_t)commandCode;
- (void)errReceived:(uint8\_t)errCode;
- (void)readerInfoReceived:(NSData \*)data;
- (void)regionReceived:(uint8\_t)region;
- (void)selectParamReceived:(NSData \*)selParam;
- (void)queryParamReceived:(NSData \*)qryParam;
- (void)channelReceived:(uint8\_t)channel channelOffset:(uint8\_t)channelOffset;
- (void)fhLbtReceived:(NSData \*)fhLb;
- (void)txPowerLevelReceived:(uint8\_t)power;
- (void>tagMemoryReceived:(NSData \*)data;
- (void)hoppingTableReceived:(NSData \*)table;
- (void)modulationParamReceived:(uint8\_t)param;
- (void)anticolParamReceived:(uint8\_t)param;
- (void)tempReceived:(uint8\_t)temp;
- (void)rssiReceived:(uint16\_t)rssi;
- (void)registryItemReceived:(NSData \*)item; @end
- (void)adcReceived:(NSData \*)data;

● Instance Method

<b>Method</b>	- (id)init
<b>Description</b>	Returns a newly initialized API.
<b>Parameters</b>	None
<b>Return Value</b>	A newly initialized API object.

<b>Method</b>	- (BOOL)open
<b>Description</b>	Returns a Boolean value that indicating whether audio device is initialized.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	- (BOOL)isOpened
<b>Description</b>	Returns a Boolean value that indicating whether audio device is initialized.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Open NO: Close

<b>Method</b>	@optional - (void)plugged:(BOOL)plug
<b>Description</b>	You can override this method to perform additional tasks associated with headset plug status
<b>Parameters</b>	YES: Plugged NO: Unplugged
<b>Return Value</b>	None

<b>Method</b>	@optional <b>- (void)readerConnected</b>
<b>Description</b>	You can override this method to perform additional tasks associated with checking reader connection.
<b>Parameters</b>	None
<b>Return Value</b>	None

<b>Method</b>	@optional <b>- (void)ackReceived:(uint8_t)commandCode</b>
<b>Description</b>	You can override this method to perform additional tasks associated with non-parameter RCP acknowledge. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	commandCode: called RCP command code
<b>Return Value</b>	None

<b>Method</b>	@optional <b>- (void)errReceived:(uint8_t)errCode</b>
<b>Description</b>	You can override this method to perform additional tasks associated with RCP error. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	errCode: RCP error code.
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)startReadTags:(uint8_t)mtnu mtime:(uint8_t)mtime repeatCycle:(uint16_t)repeatCycle</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to start an automatic tag read operation during the inventory round. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	mtnu: maximum number of tag to read mtime: maximum elapsed time to tagging (sec) repeatCycle: Repeat cycle (how many times reader perform inventory round)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional <b>- (void)pcEpcReceived:(NSData *)pcEpc</b>
<b>Description</b>	You can override this method to perform additional tasks associated with processing PC and EPC.
<b>Parameters</b>	Byte Array: PC + EPC
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)stopReadTags</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to stop an automatic read2 operation. To learn more about RCP, please refer to chapter 3.

<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)getRegion</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get the current region. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional <b>- (void)regionReceived:(uint8_t)region</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving region.
<b>Parameters</b>	- Korea (0x11) - US (0x21) - Europe (0x31) - Japan (0x41) - China1 (0x51) - China2 (0x52)
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)getReaderInfo:(uint8_t)infoType</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get the reader info. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- MODEL(0x01) - SN(0x02) - STATUS(0xB0) - INFO(0xB1)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional <b>- (void)readerInfoReceived:(NSData *)data</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving reader information parameters.
<b>Parameters</b>	Byte Array: RCP select packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)setRegion:(uint8_t)region</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set the current region. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- Korea (0x11) - US (0x21)

	<ul style="list-style-type: none"> <li>- Europe (0x31)</li> <li>- Japan (0x41)</li> <li>- China1 (0x51)</li> <li>- China2 (0x52)</li> </ul>
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)getSelectParam</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get 18000-6C air interface protocol command 'Select' parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional <b>- (void)selectParamReceived:(NSData *)selfParam</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving select parameters.
<b>Parameters</b>	Byte Array: RCP select packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)setSelectParam:(uint8_t)target          action:(uint8_t)action          memoryBank:(uint8_t)memoryBank          pointer:(uint32_t)pointer          length:(uint8_t)length          truncate:(uint8_t)truncate          mask:(NSData *)mask</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set 18000-6C air interface protocol command 'Select' parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- target: S0 (0), S1 (1), S2 (2), S3 (3), SL (4)</li> <li>- action: Refer to ISO18000-6C.</li> <li>- memoryBank: RFU (0), EPC (1), TID (2), User (3)</li> <li>- pointer: Starting mask address</li> <li>- length: mask length bits</li> <li>- truncate: Enable (1) and Disable (0)</li> <li>- mask: Mask value</li> </ul>
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)getQueryParam</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get 18000-6C air interface protocol command 'Query' parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional <b>- (void)queryParamReceived:(NSData *)qryParam</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving query parameters.
<b>Parameters</b>	Byte Array: RCP packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)setQueryParam:(uint8_t)dr m:(uint8_t)m trext:(uint8_t)trext sel:(uint8_t)sel session:(uint8_t)session target:(uint8_t)target q:(uint8_t)q</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set 18000-6C air interface protocol command 'Query' parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- dr: DR=8 (0), DR=64/3 (1)</li> <li>- m: M=1 (0), M=2 (1), M=4 (2), M=8 (3)</li> <li>- trext: No pilot tone (0), Use pilot tone (1)</li> <li>- sel: All (0 or 1), ~SL (2), SL (3)</li> <li>- session (2-bit): S0 (0), S1 (1), S2 (2), S3 (3)</li> <li>- target (1-bit): A (0), B (1)</li> <li>- q: 0-15; the number of slots in the round.</li> </ul>
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)getChannel</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get RF channel. This method is valid only for non-FH mode. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional <b>- (void)channelReceived:(uint8_t)channel channelOffset:(uint8_t)channelOffset</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving channel.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- channel: Channel number. The range of channel number depends on regional settings</li> <li>- channelOffset: Channel number offset for miller subcarrier.</li> </ul>
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)setChannel:(uint8_t)channel channelOffset:(uint8_t)channelOffset</b>
---------------	---



<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set RF channel. This method is valid only for non-FH mode. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- channel: Channel number. The range of channel number depends on regional settings - channelOffset: Channel number offset for miller subcarrier.
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)getFhLbtParam</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get FH and LBT control parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional <b>- (void)fhLbtReceived:(NSData *)fhLb</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving FH and LBT control parameters.
<b>Parameters</b>	Byte Array: RCP packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)setFhLbtParam:(uint16_t)readTime idleTime:(uint16_t)idleTime carrierSenseTime:(uint16_t) carrierSenseTime rfLevel:(uint16_t)rfLevel frequencyHopping:(uint8_t)frequencyHopping listenBeforeTalk:(uint8_t)listenBeforeTalk continuousWave:(uint8_t)continuousWave;</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set FH and LBT control parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)getOutputPowerLevel</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get current output power level. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional
---------------	-----------

	<b>- (void)txPowerLevelReceived:(uint8_t)power</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving current output power level
<b>Parameters</b>	- power : output power in dBm x 10
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)setOutputPowerLevel:(uint16_t)power</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set current output power level. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- power : output power in dBm x 10
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)setRfCw:(uint8_t)on</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to turn the Continuous Wave (CW) signal on/off. This command method is only valid for idle mode. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- On (0xFF) - Off (0x00)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)readFromTagMemory:(uint32_t)accessPassword epc:(NSData*)epc memoryBank:(uint8_t)memoryBank startAddress:(uint16_t)startAddress dataLength:(uint16_t)dataLength;</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to read tag memory from specified memory bank. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- accessPassword: Access Password if target memory bank was password protected. Otherwise, set AP filed to 0x00000000. - epc: Target tag's EPC - memoryBank: Target memory bank; RFU (0x00), EPC (0x01), TID (0x02), User (0x03) - startAddress: Starting Address word pointer - dataLength: Data Length (Word Count)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional <b>- (void)tagMemoryReceived:(NSData *)data</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving tag memory data from specified memory bank.
<b>Parameters</b>	Byte Array: Tag Memory Data.
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)getFreqHoppingTable</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get current frequency hopping table. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional <b>- (void)hoppingTableReceived:(NSData *)table</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving hopping table.
<b>Parameters</b>	Byte Array: RCP packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)setFreqHoppingTable:(uint8_t)tableSize channels:(NSData*)channels;</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set current frequency hopping table. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- tableSize (8-bit) - channels (variable)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)writeToTagMemory:(uint32_t)accessPassword epc:(NSData*)epc memoryBank:(uint8_t)memoryBank startAddress:(uint16_t)startAddress dataToWrite:(NSData*)dataToWrite;</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set current frequency hopping table. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- accessPassword: Access Password if target memory bank was password protected. Otherwise, set AP filed to 0x00000000. - epc: Target tag's EPC - memoryBank: Target memory bank; 0x00 Reserved, 0x01 EPC, 0x02 TID, 0x03 User - startAddress: Starting Address word pointer - dataToWrite: Data to write
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)killTag:(uint32_t)killPassword epc:(NSData*)epc</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to kill a Tag. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- killPassword: Kill Password. If KP filed set to 0x00000000, 'Kill Type C Tag' command do not work. The target tag ignores it.

	- epc: Target tag's EPC
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	- <b>(BOOL)lockTagMemory:(uint32_t)accessPassword epc:(NSData*)epc lockData:(uint32_t)lockData</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to lock an indicated memory bank in the tag. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- accessPassword: Access Password if memory bank was password protected. Otherwise, set AP filed to 0x00000000. - epc: Target tag's EPC - lockData: Lock mask and action flags. Pad 12-bit zeros (dummy) to the left of 20-bit lock mask and associated action flags.
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	- <b>(BOOL)getTemperature</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get current temperature. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional - <b>(void)tempReceived:(uint8_t)temp</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving reader temperature.
<b>Parameters</b>	temp: current temperature
<b>Return Value</b>	None

<b>Method</b>	- <b>(BOOL)getRssi</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get RSSI level. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional - <b>(void)rssiReceived:(uint16_t)rssi</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving RSSI
<b>Parameters</b>	RSSI: dBm x -10
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)updateRegistry:(uint8_t)update</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set Registry Update function. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- update: Store (0x01)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)eraseRegistry:(uint8_t)erase</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set Registry Erase function. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- erase: Erase (0xFF)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>- (BOOL)getRegistryItem:(uint16_t)registryItem</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get Registry items. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- Registry Version (0x0000)</li> <li>- Firmware Date (0x0001)</li> <li>- Band (0x0002)</li> <li>- Tx power (0x0003)</li> <li>- FH/LBT (0x0004)</li> <li>- Anti-collision Mode (0x0005)</li> <li>- Modulation Mode (0x0006)</li> <li>- Query(Q) (0x0007)</li> <li>- Frequency Hopping Table (0x0008)</li> <li>- Tx Power Table (0x0009)</li> </ul>
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	@optional <b>- (void)registryItemReceived:(NSData *)item</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving tag memory data from specified memory bank.
<b>Parameters</b>	Byte Array: RCP packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>- (BOOL)setBeep:(uint8_t)on</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to turn beep on/off. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- On (0xFF)</li> <li>- Off (0x00)</li> </ul>
<b>Return Value</b>	YES: Success NO: Failure

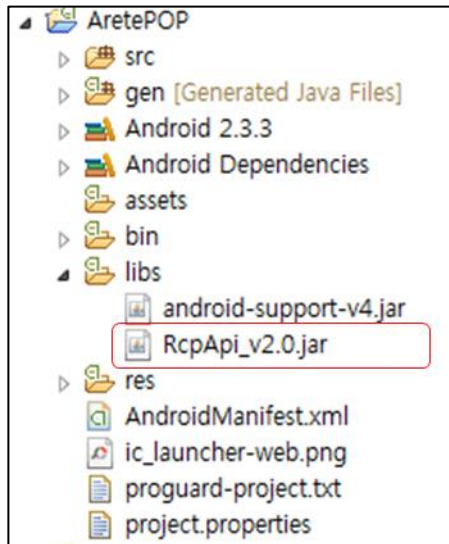
<b>Method</b>	<b>- (void)adcReceived:(NSData*)data</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving ADC values.
<b>Parameters</b>	Byte Array: RCP packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

## 2 Android Library

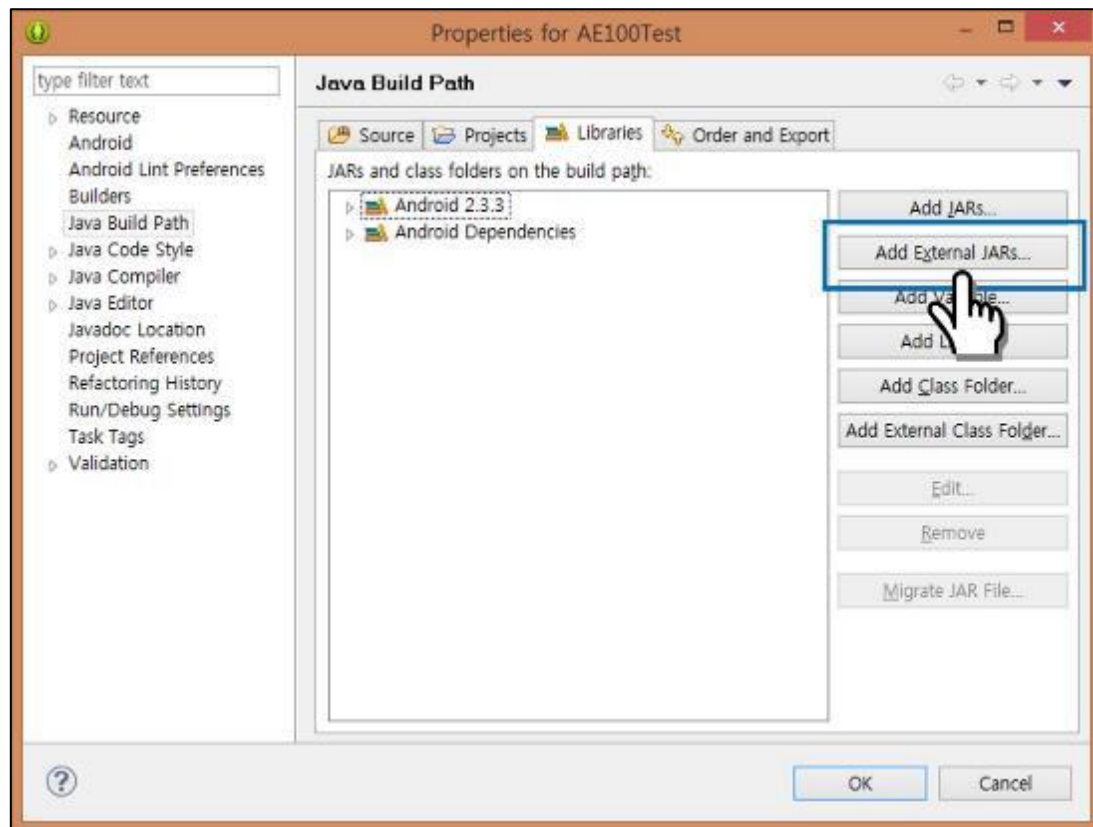
### 2.1 Add libAreteAudio

Copy the library file to the folder where the project has been created.

Add library file to your project.

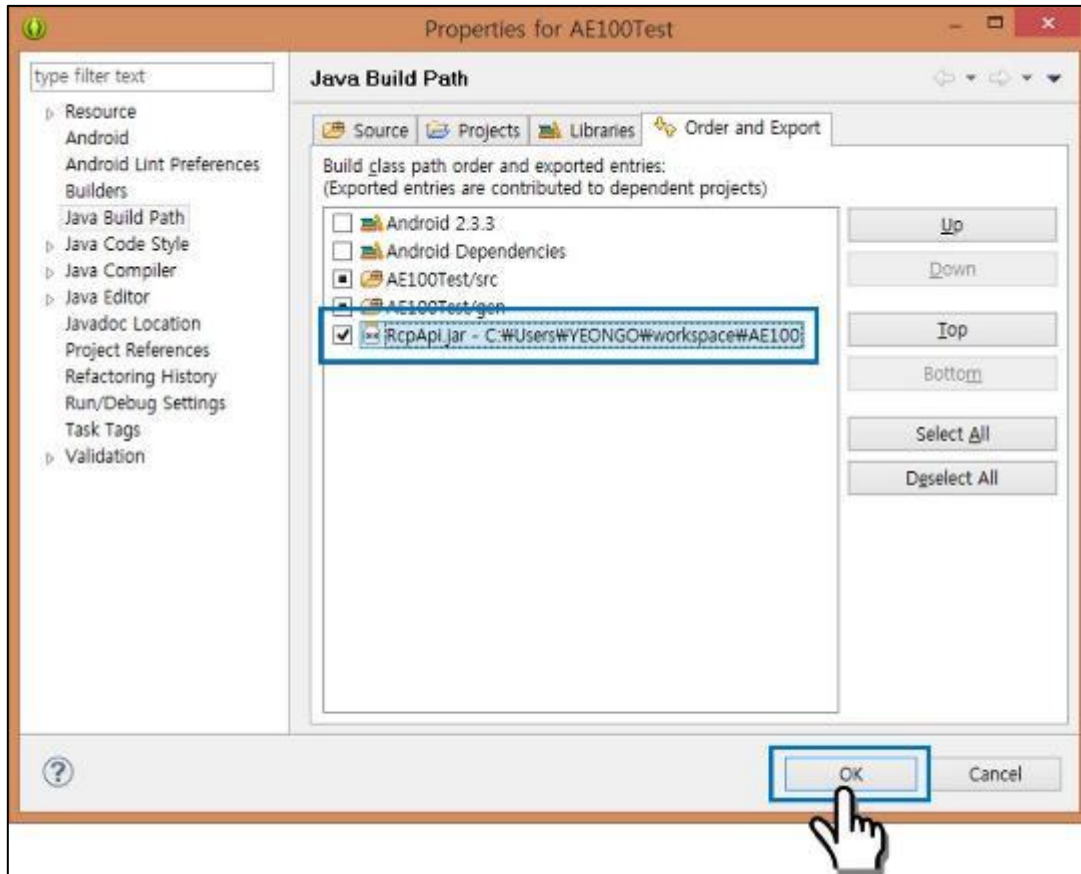


Select library files and click add.



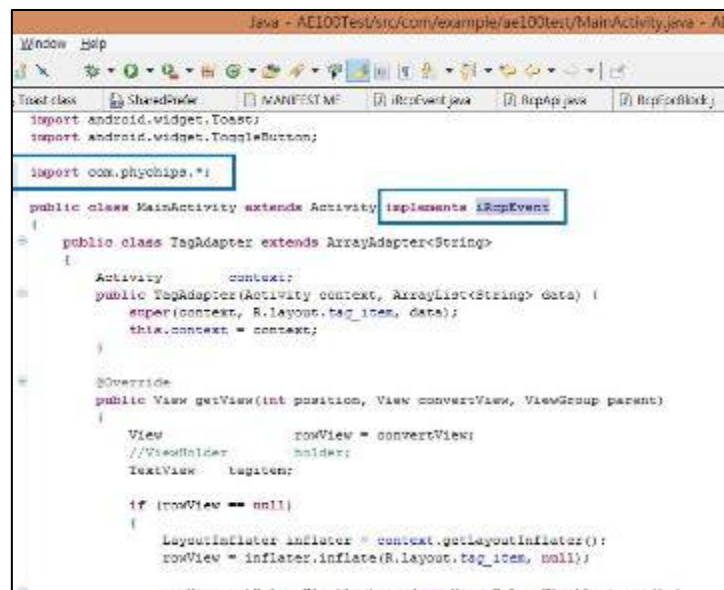
In "Order and Export" tab, check the library and click OK





Files had been added.

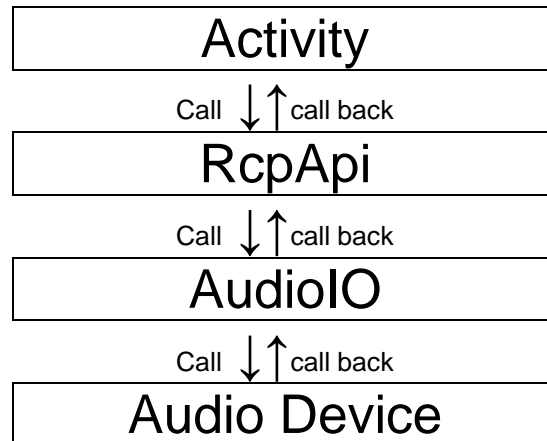
Import class and set delegate to your Activity.



## 2.2 APIs

### 2.2.1 Overview

API classes are layered as shown below. AudioIO is converting Byte Array to Audio Signal, or vice versa. RcpApi generates RCP Packet Byte Array



### 2.2.2 Rcp Api

- Inherits

None

- Property

Boolean isOpen;

- Interface

```

void onTagReceived(int[] data);
void onReaderInfoReceived(int[] data);
void onRegionReceived(int[] data);
void onSelectParamReceived(int[] data);
void onQueryParamReceived(int[] data);
void onChannelReceived(int[] data);
void onFhLbtReceived(int[] data);
void onTxPowerLevelReceived(int[] data);
void onTagMemoryReceived(int[] data);
void onHoppingTableReceived(int[] data);
void onModulationParamReceived(int[] data);
void onAnticolParamReceived(int[] data);
void onTempReceived(int[] data);
void onRssiReceived(int[] data);
void onRegistryItemReceived(int[] data);
void onResetReceived(int[] data);
void onSuccessReceived(int[] data);
void onFailureReceived(int[] data);
void onBeepStateReceived(int[] dest);
void onBatteryStateReceived(int[] dest);
void onTestFerPacketReceived(int[] dest);
  
```

- Class Method

<b>Method</b>	<b>void open()</b>
<b>Description</b>	Open audio device.
<b>Parameters</b>	None
<b>Return Value</b>	None

<b>Method</b>	<b>boolean isOpen()</b>
<b>Description</b>	Returns a Boolean value that indication whether audio device is initialized..
<b>Parameters</b>	None
<b>Return Value</b>	YES : Open NO : Close

<b>Method</b>	<b>void close()</b>
<b>Description</b>	Close audio device.
<b>Parameters</b>	None
<b>Return Value</b>	None

<b>Method</b>	<b>void onSuccessReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with non-parameter RCP acknowledge. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	data: success code
<b>Return Value</b>	None

<b>Method</b>	<b>void onFailureReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with RCP error. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	data: RCP error code.
<b>Return Value</b>	None

<b>Method</b>	<b>boolean startReadTags(int max_tags, int max_time, int repeat_cycle)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to start an automatic tag read operation during the inventory round. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	max_tags: maximum number of tag to read max_time: maximum elapsed time to tagging (sec) repeat_cycle: Repeat cycle (how many times reader perform inventory round)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onTagReceived(int[] data)</b>
---------------	---------------------------------------

<b>Description</b>	You can override this method to perform additional tasks associated with processing PC and EPC.
<b>Parameters</b>	data: PC + EPC
<b>Return Value</b>	None

<b>Method</b>	<b>boolean stopReadTags()</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to stop an automatic read2 operation. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean getRegion()</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get the current region. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onRegionReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving region.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- Korea (0x11)</li> <li>- US (0x21)</li> <li>- Europe (0x31)</li> <li>- Japan (0x41)</li> <li>- China1 (0x51)</li> <li>- China2 (0x52)</li> </ul>
<b>Return Value</b>	None

<b>Method</b>	<b>boolean setRegion(int region)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set the current region. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- Korea (0x11)</li> <li>- US (0x21)</li> <li>- Europe (0x31)</li> <li>- Japan (0x41)</li> <li>- China1 (0x51)</li> <li>- China2 (0x52)</li> </ul>
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean getReadinfor(int type)</b>
<b>Description</b>	Return a Boolean value that indicating whether RCP command is forward reader to get the reader info. To learn more about RCP. Please refer to chapter 3.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- MODEL (0x01)</li> <li>- SN (0x02)</li> <li>- STATUS (0x03)</li> <li>- INFO (0x04)</li> </ul>
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void readerinfoReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving reader information parameters.
<b>Parameters</b>	Byte Array : Rcp selected packet response payload. To learn more about RCP. Please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>boolean getTypeCSelectParam()</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get 18000-6C air interface protocol command 'Select' parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onSelectParamReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving select parameters.
<b>Parameters</b>	data: RCP select packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>boolean setTypeCSelectParam(RcpTypeCSelect param)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set 18000-6C air interface protocol command 'Select' parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- target: S0 (0), S1 (1), S2 (2), S3 (3), SL (4)</li> <li>- action: Refer to ISO18000-6C.</li> <li>- memoryBank: RFU (0), EPC (1), TID (2), User (3)</li> <li>- pointer: Starting mask address</li> <li>- length: mask length bits</li> <li>- truncate: Enable (1) and Disable (0)</li> <li>- mask: Mask value</li> </ul>
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean getTypeCQueryParam()</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get 18000-6C air interface protocol command 'Query' parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onQueryParamReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving query parameters.
<b>Parameters</b>	data: RCP packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>boolean setTypeCQueryParam(RcpTypeCQuery param)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set 18000-6C air interface protocol command 'Query' parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- dr: DR=8 (0), DR=64/3 (1)</li> <li>- m: M=1 (0), M=2 (1), M=4 (2), M=8 (3)</li> <li>- trext: No pilot tone (0), Use pilot tone (1)</li> <li>- sel: All (0 or 1), ~SL (2), SL (3)</li> <li>- session (2-bit): S0 (0), S1 (1), S2 (2), S3 (3)</li> <li>- target (1-bit): A (0), B (1)</li> <li>- q: 0-15; the number of slots in the round.</li> </ul>
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean getCurrChannel()</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get RF channel. This method is valid only for non-FH mode. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onChannelReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving channel.
<b>Parameters</b>	data: Channel number and offset. The range of channel number depends on regional settings. Channel number offset for miller subcarrier.
<b>Return Value</b>	None

<b>Method</b>	<b>boolean setCurrChannel(RcpChannelInfo param)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded

	reader to set RF channel. This method is valid only for non-FH mode. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- channel: Channel number. The range of channel number depends on regional settings - channelOffset: Channel number offset for miller subcarrier.
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean getFhLbtParam()</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get FH and LBT control parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onFhLbtReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving FH and LBT control parameters.
<b>Parameters</b>	data: RCP packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>boolean setFhLbtParam(RcpFhLbtParam param)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set FH and LBT control parameters. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- readtime: duration of the reading process - idletime: duration of non-transmission interval - sensetime: during sense time, RSSI measurement process is done. Minimum RSSI measurement time is 3.2ms - powerlevel: Target RF power level (dBm x 100) - fhmode: enable(0x01) / disable(0x00) - lbtmode: enable(0x01) / disable(0x00) - cwmode: enable(0x01) / disable(0x00)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean getOutputPowerLevel()</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get current output power level. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onTxPowerLevelReceived(int[] data)</b>
---------------	--

<b>Description</b>	You can override this method to perform additional tasks associated with receiving current output power level
<b>Parameters</b>	data: output power in dBm x 10
<b>Return Value</b>	None

<b>Method</b>	<b>boolean setOutputPowerLevel(int power_level)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set current output power level. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- power_level : output power in dBm x 10
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean setRfCw(int control)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to turn the Continuous Wave (CW) signal on/off. This command method is only valid for idle mode. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- On (0xFF) - Off (0x00)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean readFromTagMemory(RcpTypeCTag param)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to read tag memory from specified memory bank. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- accessPassword: Access Password if target memory bank was password protected. Otherwise, set AP filed to 0x00000000. - epc: Target tag's EPC - memoryBank: Target memory bank; RFU (0x00), EPC (0x01), TID (0x02), User (0x03) - startAddress: Starting Address word pointer - dataLength: Data Length (Word Count)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onTagMemoryReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving tag memory data from specified memory bank.
<b>Parameters</b>	data: Tag Memory Data.
<b>Return Value</b>	None

<b>Method</b>	<b>boolean getFreqHoppingTable()</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get current frequency hopping table. To learn more about RCP, please refer to chapter 3.



<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onHoppingTableReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving hopping table.
<b>Parameters</b>	data: RCP packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>boolean setFreqHoppingTable(RcpFreqHoppingTable hopTable)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set current frequency hopping table. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- tableSize (8-bit) - channels (variable)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean writeToTagMemory(RcpTypeCTag param)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set current frequency hopping table. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- accessPassword: Access Password if target memory bank was password protected. Otherwise, set AP filed to 0x00000000. - epc: Target tag's EPC - memoryBank: Target memory bank; 0x00 Reserved, 0x01 EPC, 0x02 TID, 0x03 User - startAddress: Starting Address word pointer - dataToWrite: Data to write
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean killTag(RcpTypeCTag param)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to kill a Tag. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- killPassword: Kill Password. If KP filed set to 0x00000000, 'Kill Type C Tag' command do not work. The target tag ignores it. - epc: Target tag's EPC
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean lockTagMemory(RcpTypeCTag param)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to lock an indicated memory bank in the tag. To learn more about RCP, please refer to chapter 3.

<b>Parameters</b>	<ul style="list-style-type: none"> <li>- accessPassword: Access Password if memory bank was password protected. Otherwise, set AP filed to 0x00000000.</li> <li>- epc: Target tag's EPC</li> <li>- lockData: Lock mask and action flags. Pad 12-bit zeros (dummy) to the left of 20-bit lock mask and associated action flags.</li> </ul>
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean getTemperature()</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get current temperature. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onTempReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving reader temperature.
<b>Parameters</b>	data: current temperature
<b>Return Value</b>	None

<b>Method</b>	<b>boolean getRssi()</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get RSSI level. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	None
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onRssiReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving RSSI
<b>Parameters</b>	data: RSSI dBm x -10
<b>Return Value</b>	None

<b>Method</b>	<b>boolean updateRegistry(int action)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set Registry Update function. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- update: Store (0x01)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean eraseRegistry(int action)</b>
---------------	--

<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to set Registry Erase function. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	- erase: Erase (0xFF)
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>boolean getRegistry(int item)</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to get Registry items. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>- Registry Version (0x0000)</li> <li>- Firmware Date (0x0001)</li> <li>- Band (0x0002)</li> <li>- Tx power (0x0003)</li> <li>- FH/LBT (0x0004)</li> <li>- Anti-collision Mode (0x0005)</li> <li>- Modulation Mode (0x0006)</li> <li>- Query(Q) (0x0007)</li> <li>- Frequency Hopping Table (0x0008)</li> <li>- Tx Power Table (0x0009)</li> </ul>
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>void onRegistryItemReceived(int[] data)</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving tag memory data from specified memory bank.
<b>Parameters</b>	data: RCP packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

<b>Method</b>	<b>Bool nativeRcp(byte[] packet)</b>
<b>Description</b>	This Directly send the command packet to RFID reader. It returns the response packet in byte array. Auto read / stop are not supported.
<b>Parameters</b>	byte[] packet
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>Bool getBeep ()</b>
<b>Description</b>	Returns a Boolean value that indicating whether RCP command is forwarded reader to turn beep on/off. To learn more about RCP, please refer to chapter 3.
<b>Parameters</b>	byte[] packet
<b>Return Value</b>	YES: Success NO: Failure

<b>Method</b>	<b>Bool setBeep (Boolean state)</b>
<b>Description</b>	beep set on/off.

<b>Parameters</b>	True : on / false : off
<b>Return Value</b>	YES: Success NO: Failure

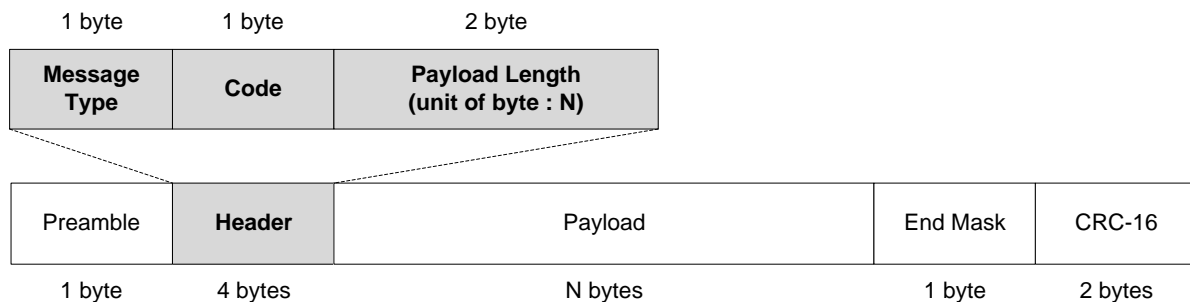
<b>Method</b>	<b>void onBeepStateReceived(int[] data);</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving beep state.
<b>Parameters</b>	Data : beep state( 0x01 : on, 0x00 : off)
<b>Return Value</b>	None

<b>Method</b>	<b>void onTestFerPacketReceived(int[] dest);</b>
<b>Description</b>	You can override this method to perform additional tasks associated with receiving ADC values.
<b>Parameters</b>	Byte Array: RCP packet response payload. To learn more about RCP, please refer to chapter 3.
<b>Return Value</b>	None

### 3 Reader Control Protocol

#### 3.1 RFID Reader Control Protocol overview

ARETE POP 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 1 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 1 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.

##### 3.1.1 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 field contains 0xBB or 0x7E (or both.) To tell these fields from other payload data, the header field has a payload length field.

##### 3.1.2 Header Field

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

##### 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.

### 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
Get Reader Information	0x00 / 0x01	0x03
Get Region	0x00 / 0x01	0x06
Set Region	0x00 / 0x01	0x07
Get Type C A/I Select Parameters	0x00 / 0x01	0x0B
Set Type C A/I Select Parameters	0x00 / 0x01	0x0C
Get Type C A/I Query Related Parameters	0x00 / 0x01	0x0D
Set Type C A/I Query Related Parameters	0x00 / 0x01	0x0E
Get current RF Channel	0x00 / 0x01	0x11
Set current RF Channel	0x00 / 0x01	0x12
Get FH and LBT Parameters	0x00 / 0x01	0x13
Set FH and LBT Parameters	0x00 / 0x01	0x14
Get Tx Power Level	0x00 / 0x01	0x15
Set Tx Power Level	0x00 / 0x01	0x16
RF CW signal control	0x00 / 0x01	0x17
Read Type C Tag Data	0x00 / 0x01	0x29
Get Frequency Hopping Table	0x00 / 0x01	0x30
Set Frequency Hopping Table	0x00 / 0x01	0x31
Start Auto Read2	0x00 / 0x01 / 0x02	0x36
Stop Auto Read2	0x00 / 0x01	0x37
Write Type C Tag Data	0x00 / 0x01	0x46
Kill/Recom Type C Tag	0x00 / 0x01	0x65
Lock Type C Tag	0x00 / 0x01	0x82
Set Beep On	0x00 / 0x01	0xAB
Get Temperature	0x00 / 0x01	0xB7
Get RSSI	0x00 / 0x01	0xC5
Update Registry	0x00 / 0x01	0xD2
Erase Registry	0x00 / 0x01	0xD3
Get Registry Item	0x00 / 0x01	0xD4
Get ADC	0x00 / 0x01	0xDD
Command Failure	0x01	0xFF

**Table 2 Message codes**

**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.

**3.1.3 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.

**3.1.4 Cyclic Redundancy Check(CRC) Field**

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.

### 3.2 Get Reader Information

Get basic information from the reader.

#### 3.2.1 Command

Message Type: Command (0x00)

Code: Get Reader Information (0x03)

Arguments

- Model (0x00)
- S/N (0x01)
- STATUS(0xB0)
- INFO(0xB1)

Example) Reads reader manufacturer

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

#### 3.2.2 Response

Message Type: Response (0x01)

Code: Get Reader Information (0x03)

Arguments

- String (variable length)

Example) 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	



### 3.3 Get Region

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

#### 3.3.1 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

#### 3.3.2 Response

Message Type: Response (0x01)

Code: Get Region (0x06)

- Korea (0x11)

- US (0x21)

- Europe (0x31)

- Japan (0x41)

- China1 (0x51)

- China2 (0x52)

Example) Europe

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

### 3.4 Set Region

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

#### 3.4.1 Command

Message Type: Command (0x00)

Code: Set Region (0x07)

Arguments

- Korea (0x11)
- US (0x21)
- Europe (0x31)
- Japan (0x41)
- China1 (0x51)
- China2 (0x52)

Example) Europe

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

#### 3.4.2 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

### 3.5 Get Type C A/I Select Parameters

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

#### 3.5.1 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

#### 3.5.2 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

Length: 0x20, Mask: 00000000000000000000000000000000										
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									

### 3.6 Set Type C A/I Select Parameters

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

#### 3.6.1 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=11111111111111111000000000000000

length: 0x20, T: 0, Mask: 00000000000000000000000000000000										
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	00000000	0xFF	0xFF			0x00	0x00
End Mark	CRC-16									
0x7E	0xNNNN									

#### 3.6.2 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

### 3.7 Get Type C A/I Query Parameters

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

#### 3.7.1 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

#### 3.7.2 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							

### 3.8 Set Type C A/I Query Parameters

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

#### 3.8.1 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							

#### 3.8.2 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

### 3.9 Get current RF Channel

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

#### 3.9.1 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

#### 3.9.2 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							

### 3.10 Set current RF Channel

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

#### 3.10.1 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							

#### 3.10.2 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



### 3.11 Get FH and LBT Parameters

Get FH and LBT control

#### 3.11.1 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

#### 3.11.2 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) / disable (0x00)
- LBT (8-bit): enable (0x01) / 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						

### 3.12 Set FH and LBT Parameters

Set FH and LBT Parameters

#### 3.12.1 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) / disable (0x00)
- LBT (8-bit): enable (0x01) / disable (0x00)
- CW (8-bit): enable (0x01) / disable (0x00)

Example) FH disable, LBT enable, 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	0x00	0x01	0x00
End Mark	CRC-16						
0x7E	0xNNNN						

#### 3.12.2 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

### 3.13 Get Tx Power Level

Get current Tx power level.

#### 3.13.1 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

#### 3.13.2 Response

Message Type: Response (0x01)

Code: Get Tx Power Level (0x15)

Arguments

- PWR (16-bit): PR9200 Power

Example) PWR = 200 (20.0 dBm)

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	PWR (MSB)	PWR (LSB)	End Mark
0xBB	0x01	0x15	0x00	0x02	0x00	0xC8	0x7E
CRC-16							
0xNNNN							

### 3.14 Set Tx Power Level

Set current Tx power level.

#### 3.14.1 Command

Message Type: Command (0x00)

Code: Set Tx Power Level (0x16)

Arguments

- PWR (16-bit): PR9200 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							

#### 3.14.2 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

### 3.15 RF CW signal control

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

#### 3.15.1 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

#### 3.15.2 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

### 3.16 Read Type C Tag Data

Read Type C tag data from specified memory bank.

#### 3.16.1 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 filed 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)

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		

#### 3.16.2 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	

### 3.17 Get Frequency Hopping Table

Get current frequency hopping table.

#### 3.17.1 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

#### 3.17.2 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		

### 3.18 Set Frequency Hopping Table

Set current frequency hopping table.

#### 3.18.1 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		

#### 3.18.2 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



### 3.20 Start Auto Read2

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

#### 3.20.1 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				

#### 3.20.2 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

#### 3.20.3 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

### 3.21 Stop Auto Read2

Stop an automatic read2 operation.

#### 3.21.1 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

#### 3.21.2 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

### 3.22 Write Type C Tag Data

Write type C tag data.

#### 3.22.1 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 filed 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		

#### 3.22.2 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

### 3.23 Kill Type C Tag

Kill a Tag.

#### 3.23.1 Command

Message Type: Command (0x00)

Code: Kill Type C Tag (0x65)

Arguments

- KP (32-bit): Kill Password. If KP field 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

Example)

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

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	KP (MSB)		
0xBB	0x00	0x65	0x00	0x12	0x87	0x65	0x43
KP (LSB)	UL (MSB)	UL (LSB)	EPC (MSB)				
0x21	0x00	0x0C	0xE2	0x00	0x34	0x11	0xB8
						EPC (LSB)	End Mark
0x02	0x01	0x15	0x26	0x37	0x04	0x94	0x7E
CRC-16							
0xNNNN							

#### 3.23.2 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

### 3.24 Lock Type C Tag

Lock an indicated memory bank in the tag.

#### 3.24.1 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 field 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				

#### 3.24.2 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

### 3.25 Set Beep On

Turn the beep on/off.

#### 3.25.1 Command

Message Type: Command (0x00)

Code: Set Beep On (0xAB)

Arguments

- On (0xFF)

- Off (0x00)

Example) Turn beep on.

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

#### 3.25.2 Response

Message Type: Response (0x01)

Code: Set Beep On (0xAB)

Arguments

- Success (0x00)

Example) Success

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

### 3.26 Get Temperature

Get current temperature

#### 3.26.1 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

#### 3.26.2 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



### 3.27 Get RSSI

Get RSSI level

#### 3.27.1 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

#### 3.27.2 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							

### 3.28 Update Registry

Sets Registry Update function

#### 3.28.1 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

#### 3.28.2 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

### 3.29 Erase Registry

Sets Registry Erase function

#### 3.29.1 Command

Message Type: Command (0x00)

Code Erase Registry (0xD3)

Arguments

- Arg (8-bit): Erase (0xFF)

Example) Erase Registry

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x00	0xD3	0x00	0x01	0xFF	0x7E	0xNNNN

#### 3.29.2 Response

Message Type: Response (0x01)

Code Erase Registry (0xD3)

Arguments

- Success (0x00)

Example) Success

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	Arg	End Mark	CRC-16
0xBB	0x01	0xD3	0x00	0x01	0x00	0x7E	0xNNNN

### 3.30 Get Registry Item

Gets Registry items

#### 3.30.1 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							

#### 3.30.2 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							

### 3.31 Get ADC value

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

#### 3.31.1 Command

Message Type: Command (0x00)

Code: Get ADC value (0xDD)

Arguments

- None

Example) Get current RF channel

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

#### 3.31.2 Response

Message Type: Response (0x01)

Code: Get ADC value (0xDD)

Arguments

- VAL (8-bit): Current value
- MIN (8-bit): Minimum value of ADC
- MAX (8-bit): Maximum value of ADC

Example) Channel Number = 10

Preamble	Msg Type	Code	PL (MSB)	PL (LSB)	VAL	MIN	MAX
0xBB	0x01	0xDD	0x00	0x02	0x3C	0x00	0xFF
End Mark	CRC-16						
0x7E	0xNNNN						

### 3.32 Command failure

Response to invalid command

Message Type: Response (0x01)

Code: Command failure (0xFF)

Error codes (8-bit)

- Failure to read the tag memory (0x09)
- Failure to write data (0x10)
- 'Read Type C Tag ID Multiple' in Operation (0x0B)
- Not in mode 'Read Type C Tag ID Multiple' (0x0D)
- Invalid parameter (0x0E)
- Failure to kill a tag (0x12)
- Failure to lock a tag (0x13)
- Failure to read a tag (0x15)
- Not supported command (0x18)
- CRC Error (0xFF)

Example) Invalid parameter

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

## 4 Customer Service

### **ARETE mobile Customer Service**

Phone: + 82 42 864 2402

Fax: + 82 42 864 2403

Email: [sales@phychips.com](mailto:sales@phychips.com)

### **Address**

PHYCHIPS Inc.

#205 Migun Technoworld 1, 533 Yongsan-dong, Yuseong-gu, Daejeon, Korea, 305-500

### **Working Day and time**

Monday to Friday

09:00~18:00(Korean Time, GMT Time + 9 hours)

### **RMA Rule**

[http://www.aret-mobile.com/pdf/RMA%20Policy\\_ARETE%20mobile.pdf](http://www.aret-mobile.com/pdf/RMA%20Policy_ARETE%20mobile.pdf)

### **Local Customer Service**

Please contact where you buy.