SDK Manual

SM1001-28, 2015-06-18

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.

Please note that there may be possible mistake or omission of information in this document.

Applicable Device
This document is applicable for Windows 7.0, iOS 7.0 and Android 2.3.3 above

Revision History

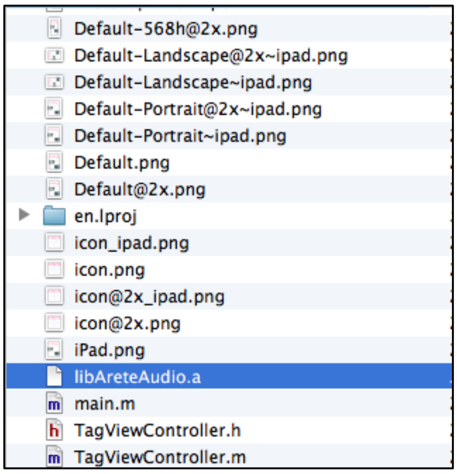
Version	Date	Changes
Rev. 10	2013-07-23	Initially released
Rev. 19	2013-12-26	Changed company address Added revision history
Rev. 20	2014-01-29	Added tag RSSI APIs
Rev. 21	2014-03-25	Renamed from RSSI to interferer RSSI Removed unopened APIs
Rev. 22	2014-05-20	Fully revised due to API2
Rev. 23	2014-08-18	Added session control
Rev. 24	2014-10-10	Modified failureReceived
Rev. 25	2014-11-28	Added startReadTagsWithTid
Rev. 26	2015-01-27	Added parameter ranges - Start Read Tags - Start Read Tags with RSSI - Start Read Tags with TID - Set Session - Set FH and LBT Parameters - Set Output Power Level
Rev. 27	2015-03-12	Added readFromTagMemoryLong Fixed the equation calculating battery gauge
Rev. 28	2015-06-18	Added Windows API for ARETE POP2

Contents

1 Adding Library

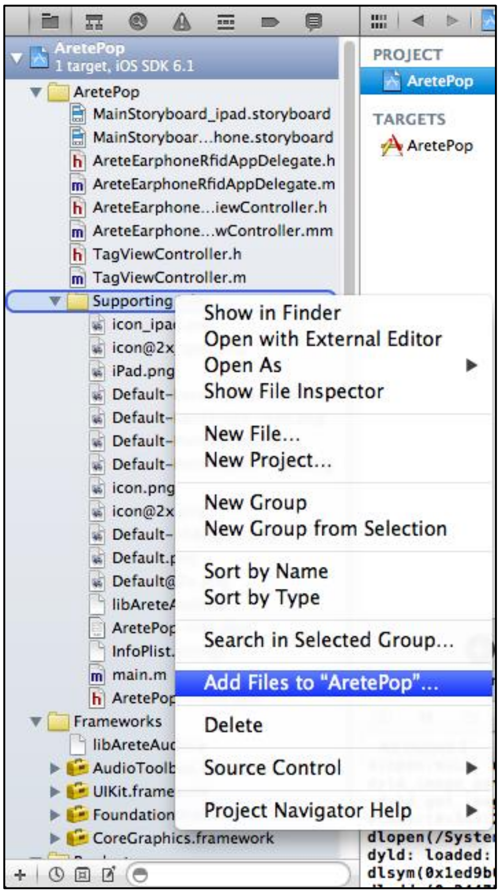
1.1 iOS

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

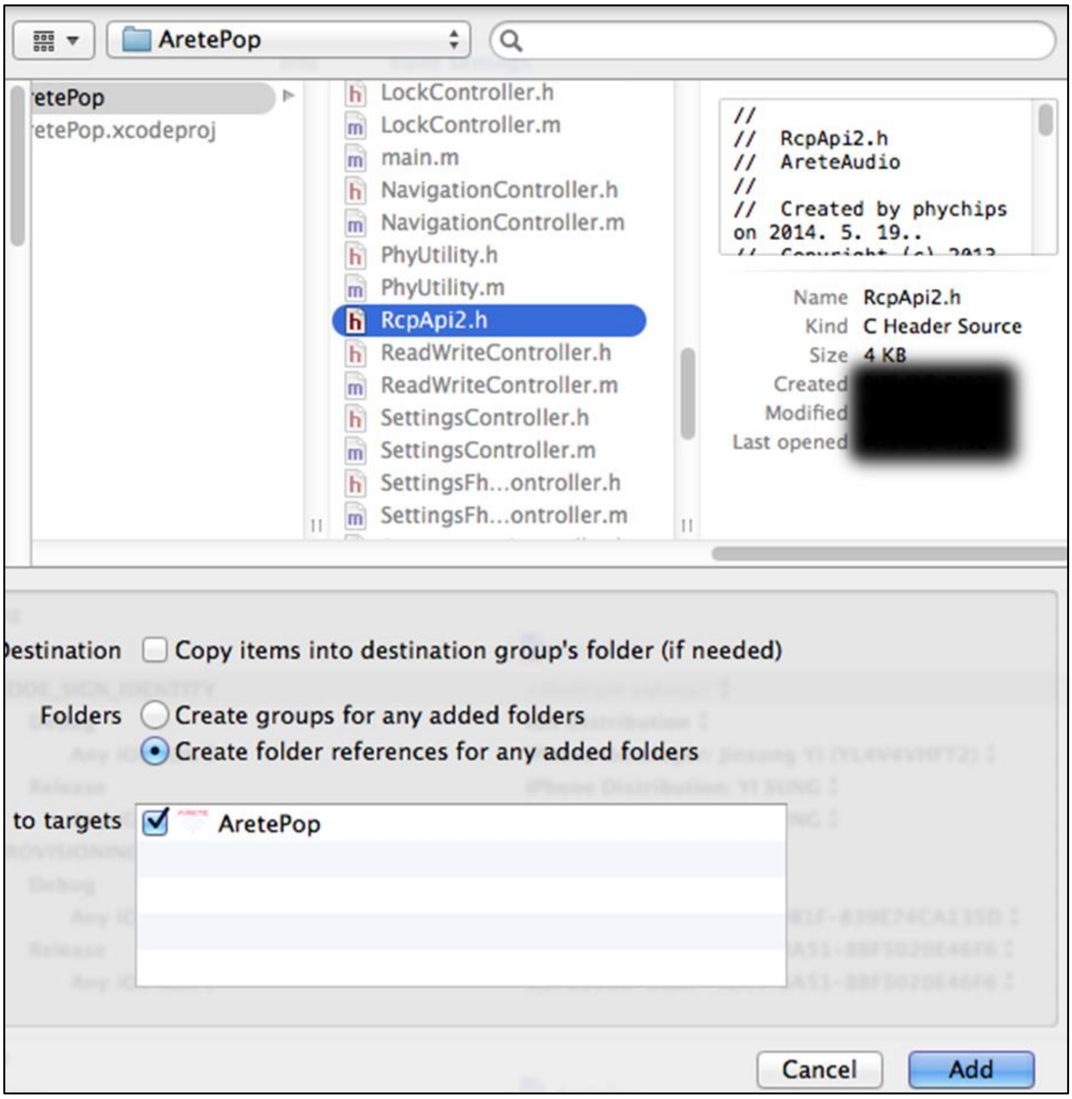


Default-568h@2x.png
Default-Landscape@2x~ipad.png
Default-Landscape~ipad.png
Default-Portrait@2x~ipad.png
Default-Portrait~ipad.png
Default.png
Default@2x.png
en.lproj
icon_ipad.png
icon.png
icon@2x_ipad.png
icon@2x.png
iPad.png
libAreteAudio.a
main.m
TagViewController.h
TagViewController.m

Select add files to your project.



Select library files and click add.



```
//
//  RcpApi2.h
//  AreteAudio
//
//  Created by phychips
on 2014. 5. 19..
```

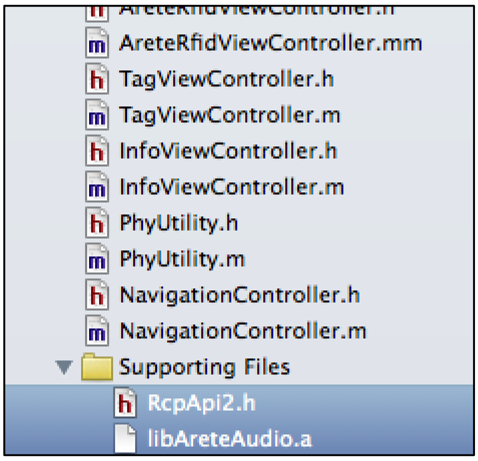
Name RcpApi2.h
Kind C Header Source
Size 4 KB
Created [redacted]
Modified
Last opened

estination □ Copy items into destination group's folder (if needed)

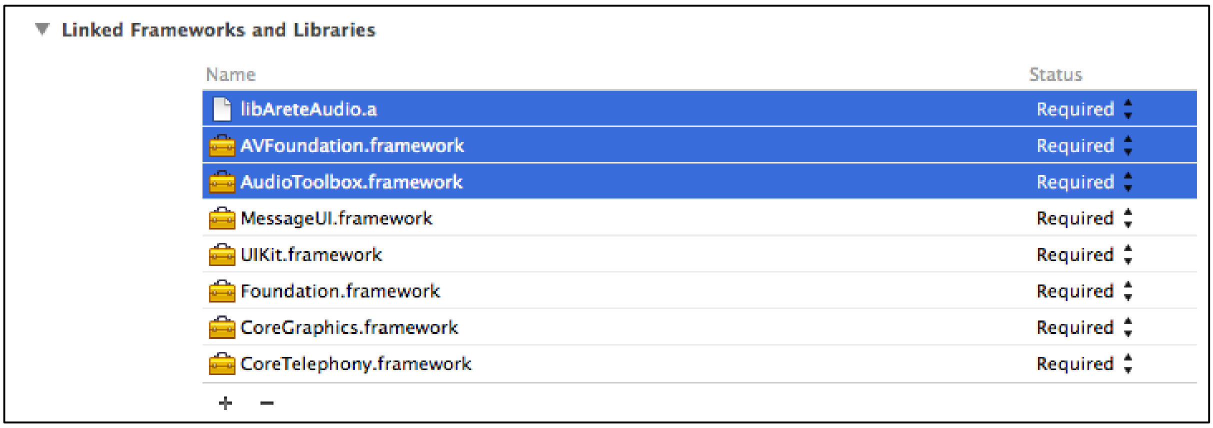
Folders ○ Create groups for any added folders
● Create folder references for any added folders

to targets ☑ AretePop

Files had been added.



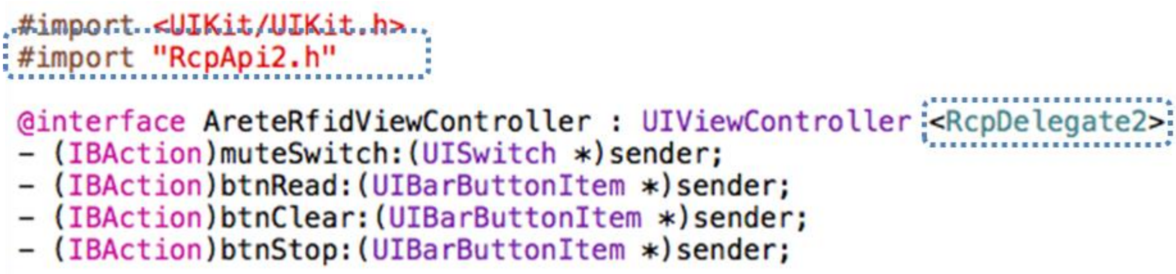
Add library file to Linked Frameworks and Libraries.



Linked Frameworks and Libraries

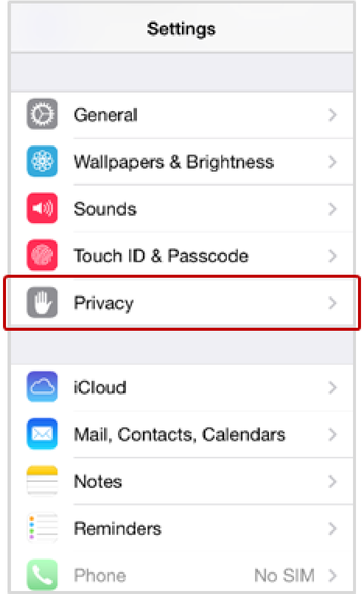
Name	Status
libAreteAudio.a	Required
AVFoundation.framework	Required
AudioToolbox.framework	Required
MessageUI.framework	Required
UIKit.framework	Required
Foundation.framework	Required
CoreGraphics.framework	Required
CoreTelephony.framework	Required

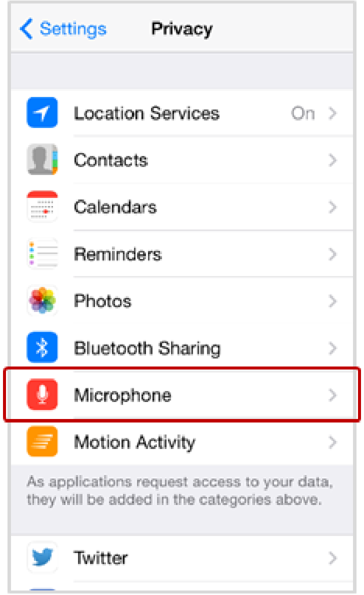
Import header file and set delegate to your ViewController.



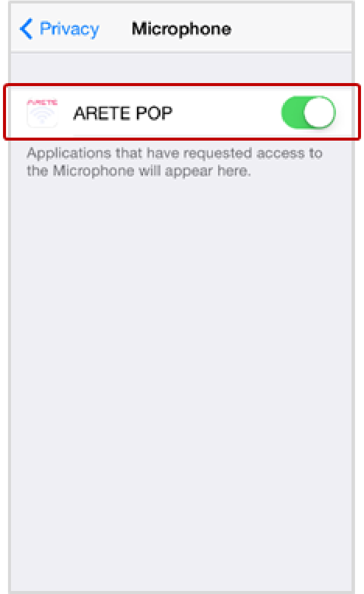
```
#import <UIKit/UIKit.h>
#import "RcpApi2.h"

@interface AreteRfidViewController : UIViewController <RcpDelegate2>
- (IBAction)muteSwitch:(UISwitch *)sender;
- (IBAction)btnRead:(UIBarButtonItem *)sender;
- (IBAction)btnClear:(UIBarButtonItem *)sender;
- (IBAction)btnStop:(UIBarButtonItem *)sender;
```

Allow microphone access permission.





As applications request access to your data, they will be added in the categories above.

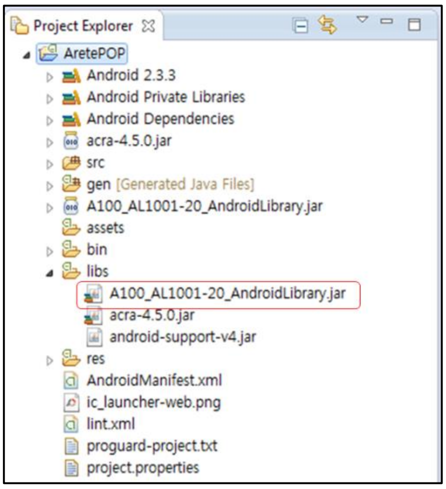


Applications that have requested access to the Microphone will appear here.

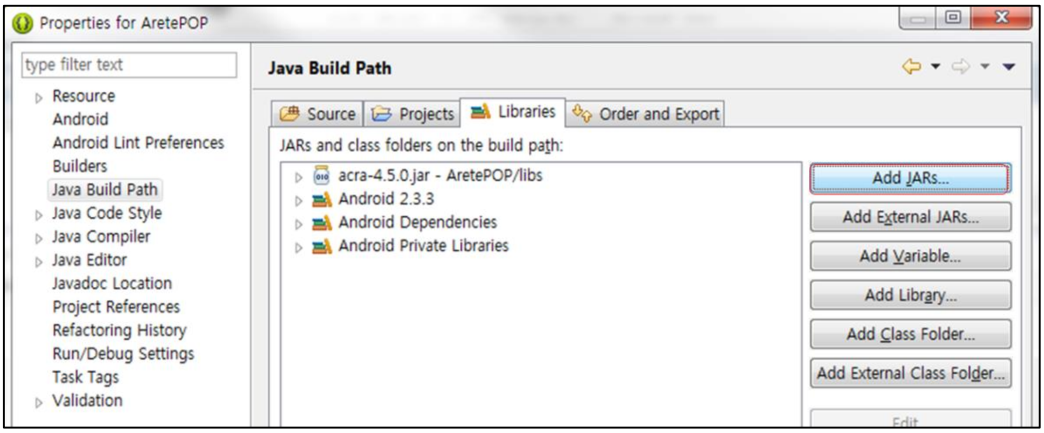
1.2 Android

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

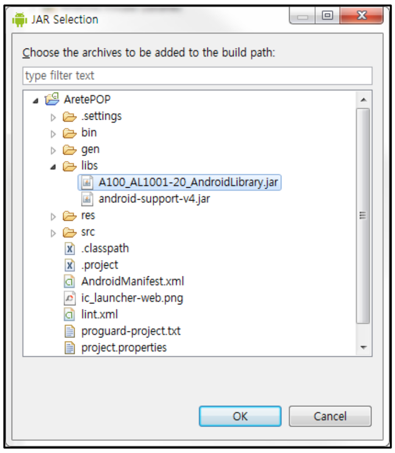
Add library file to your project.



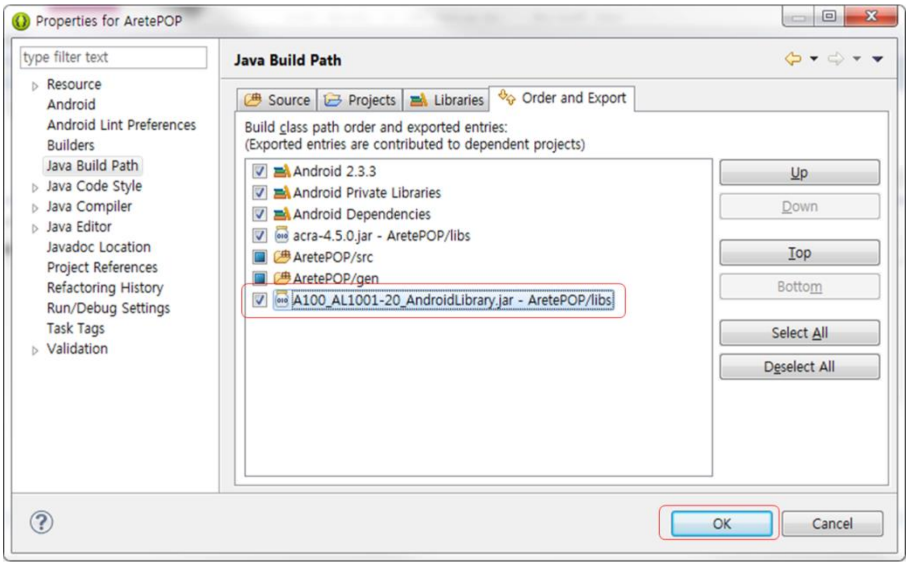
Select library files and click add.



Files had been added.



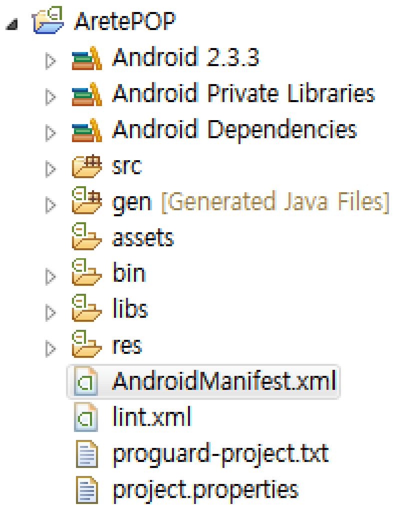
In “Order and Export” tab, check the library and click OK



Build class path order and exported entries:
(Exported entries are contributed to dependent projects)

- ☑ Android 2.3.3
- ☑ Android Private Libraries
- ☑ Android Dependencies
- ☑ acra-4.5.0.jar - AretePOP/libs
- ■ AretePOP/src
- ■ AretePOP/gen
- ☑ A100_AL1001-20_AndroidLibrary.jar - AretePOP/libs

Edit AndroidManifest.xml

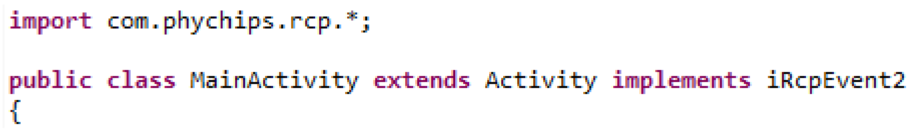


```
AretePOP
  Android 2.3.3
  Android Private Libraries
  Android Dependencies
  src
  gen [Generated Java Files]
  assets
  bin
  libs
  res
  AndroidManifest.xml
  lint.xml
  proguard-project.txt
  project.properties
```

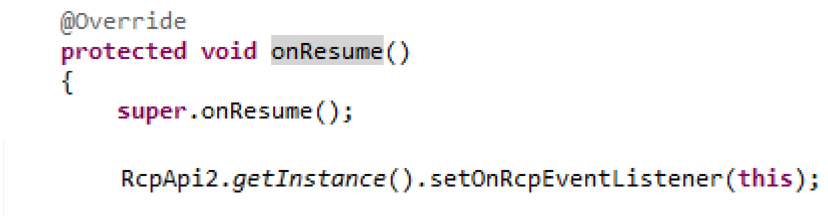
Add user permission

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.ACTION_HEADSET_PLUG" />
```

Import class and set interface to your Activity.



```
import com.phychips.rcp.*;

public class MainActivity extends Activity implements iRcpEvent2
{
```

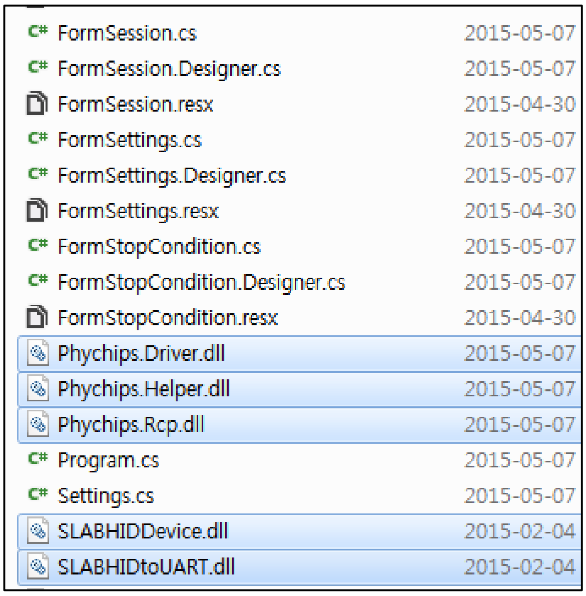


```
    @Override
    protected void onResume()
    {
        super.onResume();

        RcpApi2.getInstance().setOnRcpEventListener(this);
```

1.3 Windows

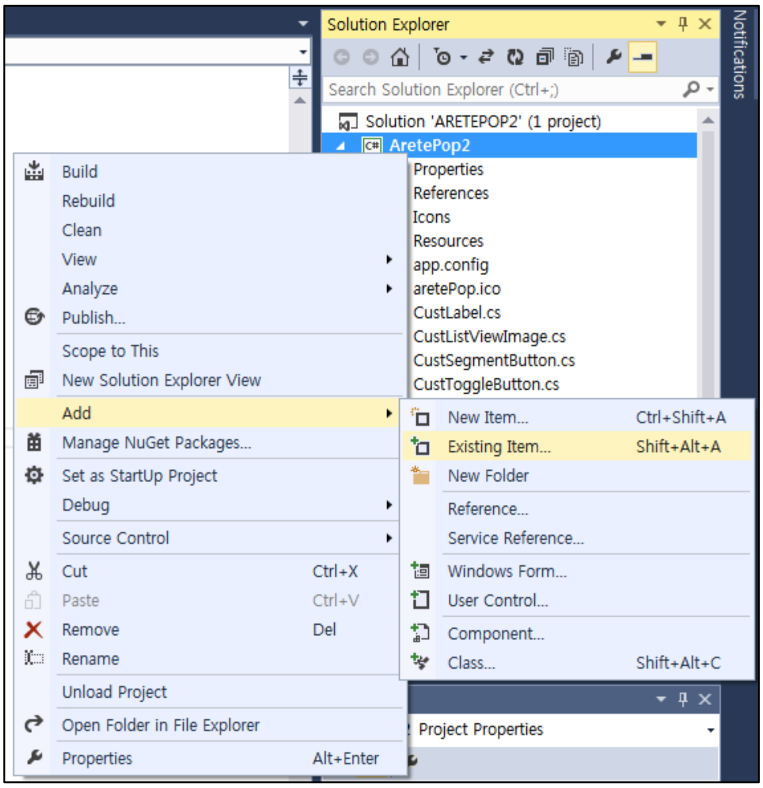
PLEASE NOTE THAT WINDOWS API IS APPLICABLE TO ONLY ARETE POP2.
Windows API does NOT use Audio interface but it use USB interface.

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

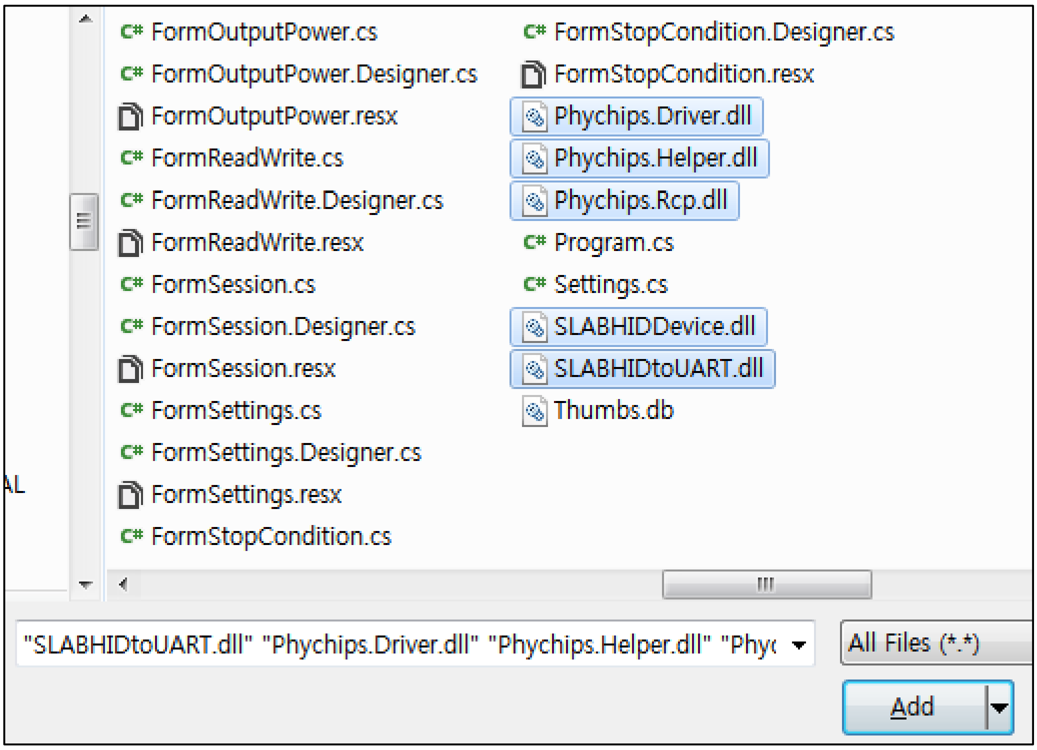


FormSession.cs	2015-05-07
FormSession.Designer.cs	2015-05-07
FormSession.resx	2015-04-30
FormSettings.cs	2015-05-07
FormSettings.Designer.cs	2015-05-07
FormSettings.resx	2015-04-30
FormStopCondition.cs	2015-05-07
FormStopCondition.Designer.cs	2015-05-07
FormStopCondition.resx	2015-04-30
Phychips.Driver.dll	2015-05-07
Phychips.Helper.dll	2015-05-07
Phychips.Rcp.dll	2015-05-07
Program.cs	2015-05-07
Settings.cs	2015-05-07
SLABHIDDevice.dll	2015-02-04
SLABHIDtoUART.dll	2015-02-04

Add library file to your project.

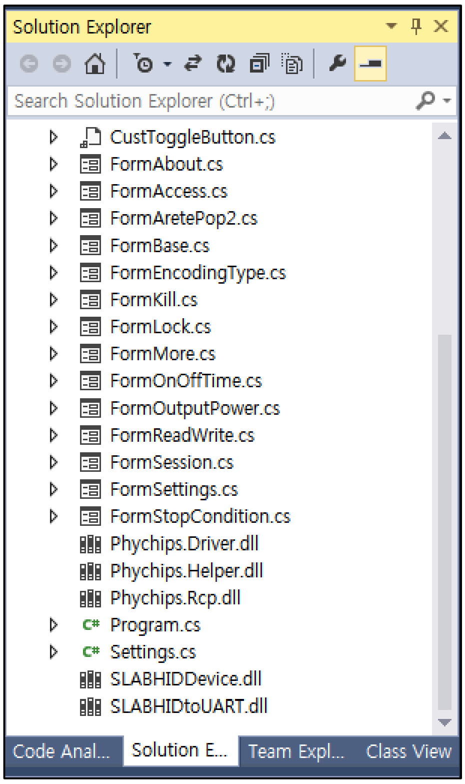


Select library files and click add.

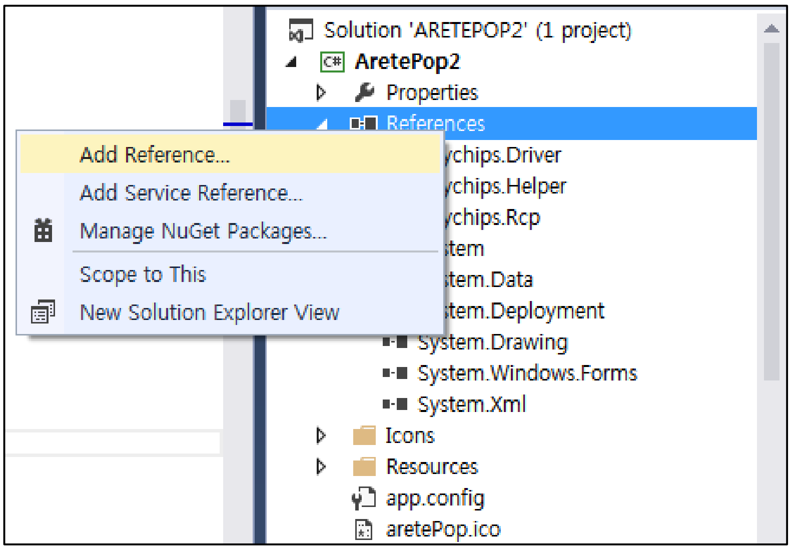


"SLABHIDtoUART.dll" "Phychips.Driver.dll" "Phychips.Helper.dll" "Phyc

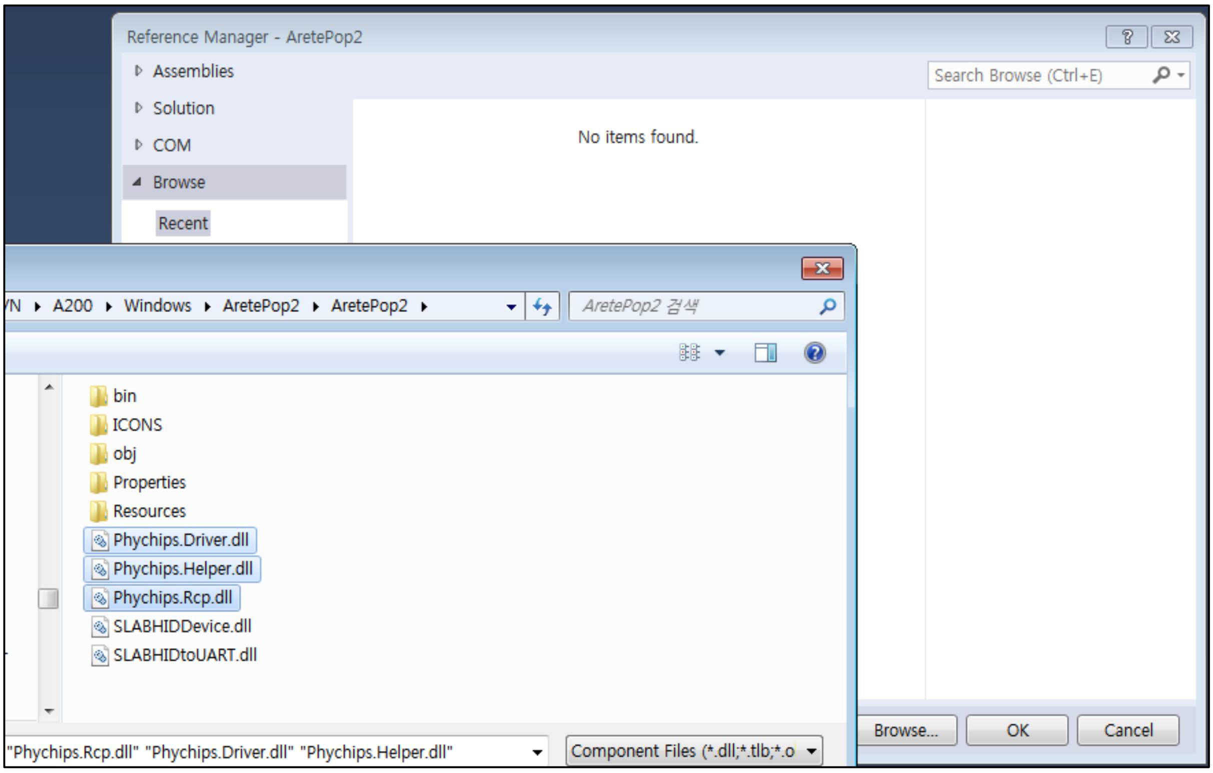
Files had been added.



Add Reference.

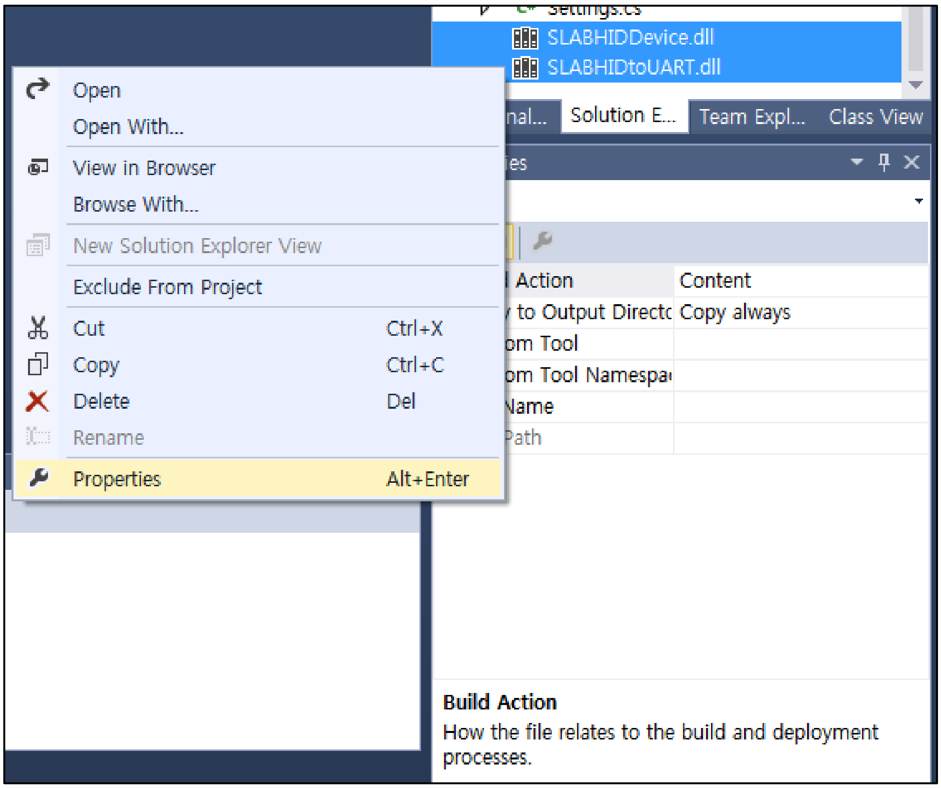


Browse and add c# libraries; Phychips.Driver.dll, Phychips.Helper.dll, and Phychps.Rcp.dll.



"Phychips.Rcp.dll" "Phychips.Driver.dll" "Phychips.Helper.dll"

Select c++ libraries; SLABHIDDevice.dll and SLABHIDtoUART.dll.
Edit Properties
Set ‘copy to output directory’ as ‘Copy always’



Action	Content
to Output Directc	Copy always
om Tool	
om Tool Namespa	
Name	
Path	

Build Action
How the file relates to the build and deployment processes.

Import class and set interface to your Form.

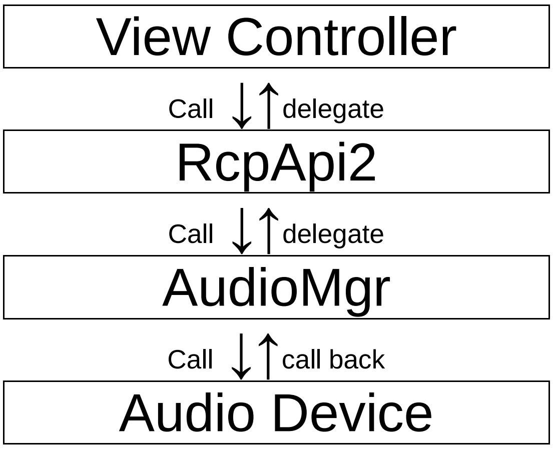
```
namespace Phychips.Arete
{
    partial class FormAretePop2

    private void InitializeComponent()
    {
      ……
      this.VisibleChanged += new System.EventHandler(this.Form_VisibleChanged);
      ……
    }
}
```

```
using Phychips.Rcp;

namespace Phychips.Arete
{
    public partial class FormAretePop2 : Form, IRcpEvent2
    {
        private void Form_VisibleChanged (object sender, EventArgs e)
        {
            RcpApi2.Instance.setOnRcpEventListener(this);
```

2 API overview

2.1 iOS

API classes are layered as shown below. AudioMgr is converting Byte Array to Audio Signal, or vice versa. RcpApi2 generates RCP Packet Byte Array



2.1.1 RcpApi2

● Propery

```
@property (nonatomic, assign) BOOL isConnected
@property (nonatomic, weak) id<RcpDelegate> delegate
```

● RcpDelegate Delegate

```
@optional
- (void)plugged:(BOOL)plug
- (void)resetReceived
- (void)successReceived:(NSData *)data commandCode:(uint8_t)commandCode
- (void)failureReceived:(uint8_t)errCode
- (void)tagReceived:(NSData *)pcEpc
- (void)tagWithRssiReceived:(NSData *)pcEpc rssi:(uint8_t)rssi
- (void)tagWithTidReceived:(NSData *)pcEpc tid:(NSData *)tid
- (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)sessionReceived:(uint8_t)session
- (void)fhLbtReceived:(NSData *)fhLb
- (void)txPowerLevelReceived:(uint8_t)power
- (void)tagMemoryReceived:(NSData *)data
- (void)batteryStateReceived:(NSData*)data
- (void)genericReceived:(NSData*)data
```

- Class Method

```
+ (RcpApi2*)sharedInstance
```

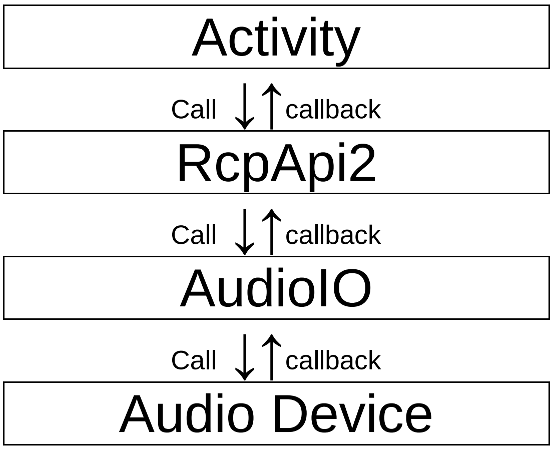
- Instance Method

```
- (BOOL)open
- (BOOL)isOpened
- (void)close
- (BOOL)startReadTags:(uint8_t)mtnu
             mtime:(uint8_t)mtime
             repeatCycle:(uint16_t)repeatCycle
- (BOOL)startReadTagsWithRssi:(uint8_t)mtnu
             mtime:(uint8_t)mtime
             repeatCycle:(uint16_t)repeatCycle
-(BOOL)startReadTagsWithTid:(uint8_t)mtnu
             mtime:(uint8_t)mtime
             repeatCycle:(uint16_t)repeatCycle;
- (BOOL)stopReadTags
- (BOOL)getRegion
- (BOOL)getReaderInfo:(uint8_t)infoType
- (BOOL)getSelectParam
- (BOOL)setSelectParam:(uint8_t)target
             action:(uint8_t)action
             memoryBank:(uint8_t)memoryBank
             pointer:(uint32_t)pointer
             length:(uint8_t)length
             mask:(NSData *)mask
- (BOOL)getQueryParam
- (BOOL)getChannel
- (BOOL)getSession;
- (BOOL)setSession:(uint8_t)session;
- (BOOL)getFhLbtParam
- (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
- (BOOL)getOutputPowerLevel
- (BOOL)setOutputPowerLevel:(uint16_t)power
- (BOOL)readFromTagMemory:(uint32_t)accessPassword
             epc:(NSData*)epc
             memoryBank:(uint8_t)memoryBank
             startAddress:(uint16_t)startAddress
             dataLength:(uint16_t)dataLength
- (BOOL)readFromTagMemoryLong:(uint32_t)accessPassword
             epc:(NSData*)epc
             memoryBank:(uint8_t)memoryBank
             startAddress:(uint16_t)startAddress
             dataLength:(uint16_t)dataLength;
- (BOOL)writeToTagMemory:(uint32_t)accessPassword
             epc:(NSData*)epc
             memoryBank:(uint8_t)memoryBank
             startAddress:(uint16_t)startAddress
             dataToWrite:(NSData*)dataToWrite
- (BOOL)killTag:(uint32_t)killPassword
             epc:(NSData*)epc
```

```
- (BOOL)lockTagMemory:(uint32_t)accessPassword
            epc:(NSData*)epc
            lockData:(uint32_t)lockData
- (BOOL)setBeep:(uint8_t)on
- (BOOL)genericTrasport:(uint32_t)accessPassword
            epc:(NSData *)epc
            ts:(uint8_t)ts
            rm:(uint8_t)rm
            sz:(uint8_t)sz
            gc:(NSData *)gc
```

2.2 Android

API classes are layered as shown below. AudioIO is converting Byte Array to Audio Signal, or vice versa. RcpApi2 generates RCP Packet Byte Array



2.2.1 Rcp Api2

- iRcpEvent2 Interface

void onResetReceived()
void onSuccessReceived(int[] data, int commandCode)
void onFailureReceived(int[] data)
void onTagReceived(int[] data)
void onTagWithRssiReceived(int[] data, int rssi)
void onTagWithTidReceived(int[] pcEpc, int[] tid)
void onReaderInfoReceived(int[] data)
void onRegionReceived(int region)
void onSelectParamReceived(int[] selParam)
void onQueryParamReceived(int[] data)
void onChannelReceived(int channel, int channelOffset)
void onFhLbtReceived(int[] data)
void onTxPowerLevelReceived(int power)
void onTagMemoryReceived(int[] data)
void onBatteryStateReceived(int[] data)
void onSessionReceived(int session)
void onGenericTransportReceived(final int[] dest)

- Class Method

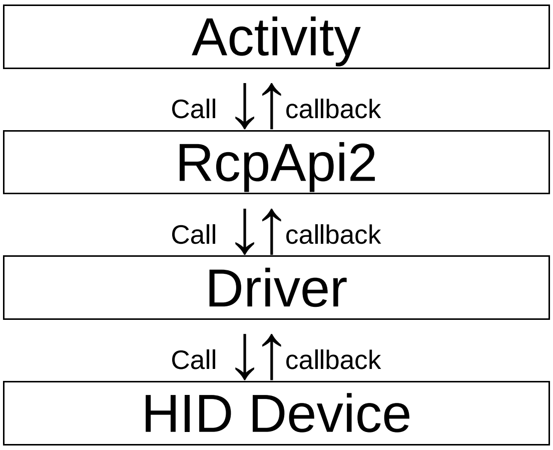
```
RcpApi2 getInstance()
```

- Instance Method

```
boolean open()
boolean isOpen()
boolean close()
boolean startReadTags(int max_tags,
          int max_time,
          int repeat_cycle)
boolean startReadTagsWithRssi(int max_tags,
          int max_time,
          int repeat_cycle)
boolean startReadTagsWithTid(max_tag,
          max_time,
          repeat_cycle)
boolean stopReadTags()
boolean getRegion()
boolean getReaderInfo(int type)
boolean getSelectParam()
boolean setSelectParam(int target,
          int action,
          int memoryBank,
          long pointer,
          int length,
          byte[] mask)
boolean getQueryParam()
boolean getChannel()
boolean getSession()
boolean setSession(int session)
boolean getFhLbtParam()
boolean setFhLbtParam(int readTime,
          int idleTime,
          int carrierSenseTime,
          int rfLevel,
          int frequencyHopping,
          int listenBeforeTalk,
          int continuousWave)
boolean getOutputPowerLevel()
boolean setOutputPowerLevel(int power_level)
boolean readFromTagMemory( long accessPassword,
          byte[] epc,
          int memoryBank,
          int startAddress,
          int dataLength)
boolean readFromTagMemoryLong( long accessPassword,
          byte[] epc,
          int memoryBank,
          int startAddress,
          int dataLength)
boolean writeToTagMemory( long accessPassword,
          byte[] epc,
          int memoryBank,
          int startAddress,
          byte[] data)
boolean killTag(long killPassword,
          byte[] epc,
          int recom)
boolean lockTagMemory(long accessPassword,
          byte[] epc,
          int lockData)
boolean setBeep(boolean state)
boolean genericTransport(long accessPassword,
          byte[] epc,
          int ts,
          int rm,
          int gcBitLen,
          byte[] gc)
```

2.3 Windows

API classes are layered as shown below. Driver is interfacing between HID USB device and software.



2.3.1 Rcp Api2

- Property

RcpApi2 Instance

- iRcpEvent2 Interface

void onPlugged(bool plug, string port)
void onResetReceived()
void onSuccessReceived(byte[] data, int commandCode)
void onFailureReceived(byte[] errCode)
void onTagReceived(byte[] pcEpc)
void onTagWithTidReceived(byte[] pcEpc, byte[] tid)
void onTagWithRssiReceived(byte[] pcEpc, int rssi)
void onReaderInfoReceived(byte[] data)
void onRegionReceived(int region)
void onSelectParamReceived(byte[] selParam)
void onQueryParamReceived(byte[] data)
void onChannelReceived(int channel, int channelOffset)
void onFhLbtReceived(byte[] data)
void onTxPowerLevelReceived(int data)
void onTagMemoryReceived(byte[] data)
void onTagMemoryLongReceived(byte[] dest)
void onBatteryStateReceived(byte[] dest)
void onSessionReceived(int session)
void onGenericTransportReceived(byte[] dest)

- Instance Method

```
bool open()
bool isOpen()
bool close()
bool startReadTags(int max_tags,
          int max_time,
          int repeat_cycle)
bool startReadTagsWithRssi(int max_tags,
          int max_time,
          int repeat_cycle)
```

```
bool startReadTagsWithTid(max_tag,
            max_time,
            repeat_cycle)
bool stopReadTags()
bool getRegion()
bool getReaderInfo(int type)
bool getSelectParam()
bool setSelectParam(int target,
            int action,
            int memoryBank,
            long pointer,
            int length,
            byte[] mask)
bool getQueryParam()
bool getChannel()
bool getSession()
bool setSession(int session)
bool getFhLbtParam()
bool setFhLbtParam(int readTime,
            int idleTime,
            int carrierSenseTime,
            int rfLevel,
            int frequencyHopping,
            int listenBeforeTalk,
            int continuousWave)
bool getOutputPowerLevel()
bool setOutputPowerLevel(int power_level)
bool readFromTagMemory( long accessPassword,
            byte[] epc,
            int memoryBank,
            int startAddress,
            int dataLength)
bool readFromTagMemoryLong( long accessPassword,
            byte[] epc,
            int memoryBank,
            int startAddress,
            int dataLength)
bool writeToTagMemory( long accessPassword,
            byte[] epc,
            int memoryBank,
            int startAddress,
            byte[] data)
bool killTag(long killPassword,
            byte[] epc,
            int recom)
bool lockTagMemory(long accessPassword,
            byte[] epc,
            int lockData)
bool setBeep(bool state)
bool genericTransport(long accessPassword,
            byte[] epc,
            int ts,
            int rm,
            int gcBitLen,
            byte[] gc)
```

3 Methods and Callback

3.1 Singleton and set callback

Gets global Instance

iOS	**+ (RcpApi2*)sharedInstance**
Android	**RcpApi2 getInstance()**
Windows	**RcpApi2 Instance**
Description	Returns a singleton of API.
Parameters	None
Return Value	A singleton of RcpApi2.

To set delegate at your ViewController in iOS,

```
YourViewController.h

@interface YourViewController : UIViewController <RcpDelegate>

@end


YourViewController.m

@implementation YourViewController
- (void)viewWillAppear:(BOOL)animated
{
        [super viewWillAppear:animated];
        [RcpApi2 sharedInstance].delegate = self;
}
@end
```

To set interface at your Activity in Android,

```
public class YourClass extends Activity implements iRcpEvent2
{
        @Override
        protected void onResume()
        {
                super.onResume();
                RcpApi2.getInstance().setOnRcpEventListener(this);
        }
}
```

To set interface at your Form in Windows,

```
public partial class YourClass: Form
{

    private void InitializeComponent()
    {
        ……

        this.VisibleChanged += new System.EventHandler(this.Form_VisibleChanged);

        …..
    }
}

public partial class YourClass: Form, IRcpEvent2
{

    private void Form_VisibleChanged(object sender, EventArgs e)
    {
        RcpApi2.Instance.setOnRcpEventListener(this);

        …..
    }

}
```

3.2 Command, response and notification

- Command and response

Command methods are used to control reader. After user calls a method, a response method (callback) is sent to user. All methods have corresponding response method. Call back method is using delegate in iOS and interface in Android and Windows.

- Notification

Unlike response methods, the notification methods (callback) are independently sent to user. Call back method is using delegate in iOS and interface in Android and Windows. In ‘Reading tags’ mode, the notification method gives tag information and this method is sent to user during reading round.

Code	Methods		
	Call	Callback	
	Command	Response	Notification
0x03	getReaderInfo	(on)readerInfoReceived	
0x06	getRegion	(on)regionReceived	
0x08			(on)resetReceived
0x0B	getSelectParam	(on)selectParamReceived	
0x0C	setSelectParam	(on)successReceived	
0x0D	getQueryParam	(on)queryParamReceived	
0x11	getChannel	(on)channelReceived	
0x13	getFhLbtParam	(on)fhLbtReceived	
0x14	setFhLbtParam	(on)successReceived	
0x15	getOutputPowerLevel	(on)txPowerLevelReceived	
0x16	setOutputPowerLevel	(on)successReceived	
0x29	readFromTagMemory	(on)tagMemoryReceived	
0x2A	readFromTagMemoryLong		(on) tagMemoryLongReceived
0x2E	getSession	(on)sessionReceived	
0x2F	setSession	(on)successReceived	
0x36	startReadTags	(on)successReceived	(on)tagReceived
0x37	stopReadTags	(on)successReceived	
0x38	startReadTagsWithRssi	(on)successReceived	(on)tagWithRssiReceived
0x3A	startReadTagsWithTid	(on)successReceived	(on)tagWithTidReceived
0x46	writeToTagMemory	(on)successReceived	
0x4D	genericTransport	(on) onGenericTransportReceived	
0x65	killTag	(on)successReceived	
0x82	lockTagMemory	(on)successReceived	
0xAB	setBeep	(on)successReceived	
0xDD			(on)batteryStateReceived
0xFF		(on)failureReceived	

3.2.1 Plugging

- Notification

iOS	@optional **- (void)plugged:(BOOL)plug**
Android	None
Windows	**void onPlugged(bool plug, string port)**
Description	You can override this method to perform additional tasks associated with headset plug status or HID connection status
Parameters	plug - YES/true: Plugged - NO/false: Unplugged port (Windows only) - HID port name
Return Value	None

3.2.2 Failure

- Response

iOS	@optional **- (void) failureReceived:(NSData *)errCode**
Android	**void onFailureReceived(final int[] errCode)**
Windows	**void onFailureReceived(byte[] errCode)**
Description	You can override this method to perform additional tasks associated with RCP error.
Parameters	errCode[0]: RCP error code. - 0x09: Failure to read the tag memory - 0x0B: ‘startReadTags’ in Operation - 0x0D: Not in mode ‘Read Type C Tag ID Multiple’ - 0x0E: Invalid parameter - 0x10: Failure to write data - 0x12: Failure to kill a tag - 0x13: Failure to lock a tag - 0x15: Failure to read a tag - 0x18: Not supported command - 0xFF: CRC Error errCode[1]: Command code errCode[2]: More information
Return Value	None

3.2.3 Open and close audio device

- Command

iOS	**- (BOOL)open**
Android	**boolean open()**
Windows	**bool open()**
Description	Open audio device.
Parameters	None
Return Value	YES/true: Success NO/false: Failure

iOS	**- (BOOL)isOpened**
Android	**boolean isOpen()**
Windows	**bool isOpened()**
Description	Returns a Boolean value that indicating whether audio device is initialized.
Parameters	None
Return Value	YES/true: Open NO/false: Close

iOS	**- (void)close**
Android	**boolean close()**
Windows	**void close()**
Description	Close audio device.
Parameters	None
Return Value	None

- Notification

iOS	@optional **- (void)resetReceived**
Android	**void onResetReceived()**
Windows	**void onResetReceived()**
Description	You can override this method to perform additional tasks associated with checking reader connection.
Parameters	None
Return Value	None

3.2.4 Start Read Tags

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

- Command

iOS	**- (BOOL)startReadTags:(uint8_t) maxTags** **mtime:(uint8_t) maxTime** **repeatCycle:(uint16_t)repeatCycle**
Android	**boolean startReadTags(int maxTags,** **int maxTime,** **int repeatCycle)**
Windows	**bool startReadTags(int maxTags,** **int maxTime,** **int repeatCycle)**
Description	Start a tag read operation.
Parameters	maxTags: maximum number of tag to read. 0 to 250 tags. 0 is unlimited. maxTime: maximum elapsed time to tagging (sec). 0 to 250. 0 is unlimited. repeatCycle: Repeat cycle, how many times reader perform inventory round. 0 to 65530. 0 is unlimited.
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode - requesting command code
Return Value	None

- Notification

iOS	@optional **- (void) tagReceived:(NSData *)pcEpc**
Android	**void onTagReceived(int[]pcEpc)**
Windows	**void onTagReceived(byte[] pcEpc)**
Description	You can override this method to perform additional tasks associated with processing PC and EPC.
Parameters	Byte Array: PC + EPC
Return Value	None

Example) PC = 0x3000, EPC = 0xE2003411B802011383258566

PC(MSB)	PC(LSB)	EPC (MSB)			
0x30	0x00	0xE2	0x00	0x34	0x11
0xB8	0x02	0x01	0x13	0x83	0x25
	EPC (LSB)				
0x85	0x66				

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Read complete (0x1F) commandCode - requesting command code
Return Value	None

3.2.5 Start Read Tags with RSSI

Start a tag read operation, tag IDs are sent back to user though notification method.

- Command

iOS	**- (BOOL)startReadTagsWithRssi:(uint8_t) maxTags** **mtime:(uint8_t) maxTime** **repeatCycle:(uint16_t)repeatCycle**
Android	**boolean startReadTagsWithRssi(int maxTags,** **int maxTime,** **int repeatCycle)**
Windows	**bool startReadTagsWithRssi(int maxTags,** **int maxTime,** **int repeatCycle)**
Description	Start a tag read operation with RSSI.
Parameters	maxTags: maximum number of tag to read. 0 to 250 tags. 0 is unlimited. maxTime: maximum elapsed time to tagging (sec). 0 to 250. 0 is unlimited. repeatCycle: Repeat cycle, how many times reader perform inventory round. 0 to 65530. 0 is unlimited.
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode - requesting command code
Return Value	None

- Notification

iOS	@optional **- (void) tagWithRssiReceived:(NSData *)pcEpc rssi:(int8_t)rssi**
Android	**void onTagWithRssiReceived(int[] data, int rssi)**
Windows	**void onTagWithRssiReceived(byte[] pcEpc, int rssi)**
Description	You can override this method to perform additional tasks associated with processing PC and EPC with RSSI.
Parameters	pcEpc: PC + EPC rssi: RSSI value in dBm
Return Value	None

Example) PC = 0x3000, EPC = 0xE2003411B802011383258566

PC(MSB)	PC(LSB)	EPC (MSB)			
0x30	0x00	0xE2	0x00	0x34	0x11
0xB8	0x02	0x01	0x13	0x83	0x25
	EPC (LSB)				
0x85	0x66				

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Read complete (0x1F) commandCode: requesting command code
Return Value	None

3.2.6 Start Read Tags with TID

Start an automatic tag read operation, tag ID and TID are sent back to user though notification method.

- Command

iOS	**- (BOOL) startReadTagsWithTid:(uint8_t) maxTags** **mtime:(uint8_t) maxTime** **repeatCycle:(uint16_t)repeatCycle**
Android	**boolean startReadTagsWithTid (int maxTags,** **int maxTime,** **int repeatCycle)**
Windows	**bool startReadTagsWithTid(int maxTags,** **int maxTime,** **int repeatCycle)**
Description	Start a tag read operation.
Parameters	maxTags: maximum number of tag to read. 0 to 250 tags. 0 is unlimited. maxTime: maximum elapsed time to tagging (sec). 0 to 250. 0 is unlimited. repeatCycle: repeat cycle, how many times reader perform inventory round. 0 to 65530. 0 is unlimited
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode: requesting command code
Return Value	None

- Notification

iOS	@optional **- (void) tagWithTidReceived:(NSData *)pcEpc tid:(NSData *)tid**
Android	**void onTagWithTidReceived (final int[] pcEpc, final int[] tid)**
Windows	**void onTagWithTidReceived(byte[] pcEpc, byte[] tid)**
Description	You can override this method to perform additional tasks associated with processing PC and EPC.
Parameters	Byte Array: PC + EPC
Return Value	None

Example)

Arg 1: PC = 0x3000, EPC = 0xE2003411B802011383258566

PC(MSB)	PC(LSB)	EPC (MSB)			
0x30	0x00	0xE2	0x00	0x34	0x11
0xB8	0x02	0x01	0x13	0x83	0x25
	EPC (LSB)				
0x85	0x66				

Arg 2: TID = 0xE2003411B802011383258566

TID (MSB)					
0xE2	0x00	0x34	0x11	0xB8	0x02
					EPC (LSB)
0x01	0x13	0x83	0x25	0x85	0x66

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Read complete (0x1F) commandCode: requesting command code
Return Value	None

3.2.7 Stop Read Tags

Stop a tag read operation

- Command

iOS	**- (BOOL)stopReadTags**
Android	**boolean stopReadTags()**
Windows	**bool stopReadTags()**
Description	Returns a Boolean value that indicating whether command is forwarded reader to stop an automatic read2 operation.
Parameters	None
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode - requesting command code
Return Value	None

3.2.8 Get Region

Get the current region. ARETE POP and POP2 uses individual channel table that depends on region. List of region code follows below.

- Command

iOS	**- (BOOL)getRegion**
Android	**boolean getRegion()**
Windows	**bool getRegion()**
Description	Returns a Boolean value that indicating whether command is forwarded reader to get the current region.
Parameters	None
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)regionReceived:(uint8_t)region**
Android	**void onRegionReceived(int region)**
Windows	**void onRegionReceived(int region)**
Description	You can override this method to perform additional tasks associated with receiving region.
Parameters	- Korea (0x11) - US (0x22) - EU (0x31) - Japan (0x41) - China (0x52) - Hong Kong, Singapore, and Australia (0x71)
Return Value	None

Example)
region = 0x11

3.2.9 Get Reader Information

Get basic information from the reader.

- Command

iOS	**- (BOOL)getReaderInfo:(uint8_t)infoType**
Android	**boolean getReaderInfo(int type)**
Windows	**bool getReaderInfo(int type)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to get the reader info.
Parameters	- MODEL(0x00) - SN(0x01) - STATUS(0xB0) - INFO(0xB1)
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)readerInfoReceived:(NSData *)data**
Android	**void onReaderInfoReceived(int[] data)**
Windows	**void onReaderInfoReceived(byte[] data)**
Description	You can override this method to perform additional tasks associated with receiving reader information parameters.
Parameters	Byte Array: RCP select packet response payload.
Return Value	None

3.2.10 Get Type C A/I Select Parameters

Get 18000-63 air interface protocol command ‘Select’ parameters.

- Command

iOS	**- (BOOL)getSelectParam**
Android	**boolean getSelectParam()**
Windows	**bool getSelectParam()**
Description	Returns a Boolean value that indicating whether command is forwarded reader to get 18000-63 air interface protocol command ‘Select’ parameters.
Parameters	None
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)selectParamReceived:(NSData *)selParam**
Android	**void onSelectParamReceived(int[] selParam)**
Windows	**void onSelectParamReceived(byte[] selParam)**
Description	You can override this method to perform additional tasks associated with receiving select parameters.
Parameters	Byte Array: - target (3-bit): S0 (000), S1 (001), S2 (010), S3 (011), SL (100) - action (3-bit): Refer to ISO18000-63. - memory Bank (2-bit): 00 RFU, 01 EPC, 10 TID, 11 User - pointer (32-bit): Starting mask address - length (8-bit): mask length bits - reserve (8-bit): Reserved 0000000 value should be placed here. - mask (0~255 bits): Mask value
Return Value	None

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

T	A	M	Ptr (MSB)			Ptr (LSB)	Length
000	000	11	0x00	0x00	0x00	0xFF	0x20
Reserve			Mask (MSB)			Mask (LSB)	
0000000			0xFF	0xFF	0x00	0x00	

3.2.11 Set Type C A/I Select Parameters

Set 18000-63 air interface protocol command ‘Select’ parameters.

- Command

iOS	**- (BOOL)setSelectParam:(uint8_t)target** **action:(uint8_t)action** **memoryBank:(uint8_t)memoryBank** **pointer:(uint32_t)pointer** **length:(uint8_t)length** **mask:(NSData *)mask**
Android	**boolean setSelectParam(int target,** **int action,** **int memoryBank,** **long pointer,** **int length,** **byte[] mask)**
Windows	**bool setSelectParam(int target,** **int action,** **int memoryBank,** **long pointer,** **int length,** **byte[] mask)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to set 18000-63 air interface protocol command ‘Select’ parameters.
Parameters	- target: S0 (0), S1 (1), S2 (2), S3 (3), SL (4) - action: Refer to ISO18000-63. - memoryBank: RFU (0), EPC (1), TID (2), User (3) - pointer: Starting mask address - length: mask length bits - mask: Mask value
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode: requesting command code
Return Value	None

Example)
commandCode = 0x00

3.2.12 Get Type C A/I Query Parameters

Get 18000-63 air interface protocol command 'Query' parameters.

- Command

iOS	**- (BOOL)getQueryParam**
Android	**boolean getQueryParam()**
Windows	**bool getQueryParam()**
Description	Returns a Boolean value that indicating whether command is forwarded reader to get 18000-63 air interface protocol command 'Query' parameters.
Parameters	None
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)queryParamReceived:(NSData *)qryParam**
Android	**void onQueryParamReceived(int[] data)**
Windows	**void onQueryParamReceived(byte[] data)**
Description	You can override this method to perform additional tasks associated with receiving query parameters.
Parameters	Byte Array: - 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.
Return Value	None

Example) DR=8, M=1, TRext=Use pilot tone, Sel=All, Session=S0, Target=A, Q=4

DR	M	TR	Sel	S	T	Q	RSV
0	00	1	00	00	0	0100	000

3.2.13 Get current RF Channel

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

- Command

iOS	**- (BOOL)getChannel**
Android	**boolean getChannel()**
Windows	**bool getChannel()**
Description	Returns a Boolean value that indicating whether command is forwarded reader to get RF channel. This method is valid only for non-FH mode.
Parameters	None
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)channelReceived:(uint8_t)channel** **channelOffset:(uint8_t)channelOffset**
Android	**void onChannelReceived(int channel,** **int channelOffset)**
Windows	**onChannelReceived(int channel,** **int channelOffset)**
Description	You can override this method to perform additional tasks associated with receiving channel.
Parameters	- channel(8-bit): Channel number. The range of channel number depends on regional settings - channelOffset(8-bit): Channel number offset for miller subcarrier.
Return Value	None

3.2.14 Get Session

Get session used in query parameter.

- Command

iOS	**- (BOOL) getSession**
Android	**boolean getSession ()**
Windows	**bool getSession()**
Description	Returns a Boolean value that indicating whether command is forwarded reader to get session.
Parameters	None
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)sessionReceived:(uint8_t)session**
Android	**void onSessionReceived(final int session)**
Windows	**void onSessionReceived(int session)**
Description	You can override this method to perform additional tasks associated with receiving channel.
Parameters	- session: session in query parameter.
Return Value	None

Example) session = 0 (S0)

3.2.15 Set Session

Set session used in query parameter.

- Command

iOS	**- (BOOL)setSession:(uint8_t)session**
Android	**boolean setSession(int session)**
Windows	**bool setSession(int session)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to set session.
Parameters	- session: S0(0), S1(1), S2(2), S3(3); session in query parameter.
Return Value	YES/true: Success NO/false: Failure

Example) session = 0 (S0)

- Response

iOS	@optional **- (void)successReceived:(NSData *)data commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data, int commandCode)**
Windows	**void onSuccessReceived(byte[] data, int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode: requesting command code
Return Value	None

Example)
commandCode = 0x00

3.2.16 Get FH and LBT Parameters

Get FH and LBT control parameters

- Command

iOS	**- (BOOL)getFhLbtParam**
Android	**boolean getFhLbtParam()**
Windows	**bool getFhLbtParam()**
Description	Returns a Boolean value that indicating whether command is forwarded reader to get FH and LBT control parameters.
Parameters	None
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)fhLbtReceived:(NSData *)fhLb**
Android	**void onFhLbtReceived(int[] data)**
Windows	**void onFhLbtReceived(byte[] data)**
Description	You can override this method to perform additional tasks associated with receiving FH and LBT control parameters.
Parameters	Byte Array: - 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)
Return Value	None

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

RT MSB	RT (LSB)	IT (MSB)	IT (LSB)	CST (MSB)	CST (LSB)
0x01	0x90	0x00	0x64	0x00	0x0A
RFL (MSB)	RFL (LSB)	FH	LBT	CW	
0xFD	0x8A	0x00	0x01	0x00	

3.2.17 Set FH and LBT Parameters

Set FH and LBT Parameters

- Command

iOS	**- (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;**
Android	**boolean setFhLbtParam(int readTime,** **int idleTime,** **int carrierSenseTime,** **int rfLevel,** **int frequencyHopping,** **int listenBeforeTalk,** **int continuousWave)**
Windows	**bool setFhLbtParam(int readTime,** **int idleTime,** **int carrierSenseTime,** **int lbtLevel,** **int frequencyHopping,** **int listenBeforeTalk,** **int continuousWave)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to set FH and LBT control parameters.
Parameters	- RT (16-bit): Read Time. 10 to 40000, 1 = 1ms. - IT (16-bit): Idle Time. 10 to 40000, 1 = 1ms. - CST (16-bit): Carrier Sense Time. 10 to 1000, 1 = 1ms. - RFL (16-bit): Target RF power level in -dBm x 10. 740 is valid for Japanese regulatory. - FH (8-bit): enable (0x01) / disable (0x00) - LBT (8-bit): enable (0x01) / disable (0x00) - CW (8-bit): enable (0x01) / disable (0x00)
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode: requesting command code
Return Value	None

Example)
commandCode = 0x00

3.2.18 Get Output Power Level

Get current output power level

- Command

iOS	**- (BOOL)getOutputPowerLevel**
Android	**boolean getOutputPowerLevel()**
Windows	**bool getOutputPowerLevel()**
Description	Returns a Boolean value that indicating whether command is forwarded reader to get current output power level.
Parameters	None
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)txPowerLevelReceived:(uint8_t)power**
Android	**void onTxPowerLevelReceived(int power)**
Windows	**void onTxPowerLevelReceived(int power)**
Description	You can override this method to perform additional tasks associated with receiving current output power level
Parameters	- power : output power in dBm x 10
Return Value	None

Example) power = 200 (20.0 dBm)

3.2.19 Set Output Power Level

Set current output power level.

- Command

iOS	**- (BOOL)setOutputPowerLevel:(uint16_t)power**
Android	**boolean setOutputPowerLevel(int power)**
Windows	**bool setOutputPowerLevel(int power)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to set current output power level.
Parameters	- power: output power in dBm x 10. 200 to 250. (Japanese band : 200 to 230)
Return Value	YES/true: Success NO/false: Failure

Example) power = 200 (20.0 dBm)

- Response

iOS	@optional **- (void)successReceived:(NSData *)data commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data, int commandCode)**
Windows	**void onSuccessReceived(byte[] data, int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode: requesting command code
Return Value	None

Example)
commandCode = 0x00

3.2.20 Read Tag Data

Read tag data from specified memory bank.

- Command

iOS	**- (BOOL)readFromTagMemory:(uint32_t)accessPassword** **epc:(NSData*)epc** **memoryBank:(uint8_t)memoryBank** **startAddress:(uint16_t)startAddress** **dataLength:(uint16_t)dataLength;**
Android	**boolean readFromTagMemory(long accessPassword,** **byte[] epc,** **int memoryBank,** **int startAddress,** **int dataLength)**
Windows	**bool readFromTagMemory(long accessPassword,** **byte[] epc,** **int memoryBank,** **int startAddress,** **int dataLength)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to read tag memory from specified memory bank.
Parameters	- 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)
Return Value	YES/true: data: result code - Success (0x00) commandCode: requesting command code NO/false: Failure

Example)
accessPassword = 0x00000000
epc = 0xE2003411B802011526370494
memoryBank = 0x00 (RFU)
Start Address = 0x0000
Length = 0x04 (Word Count)

- Response

iOS	@optional **- (void)tagMemoryReceived:(NSData *)data**
Android	**void onTagMemoryReceived(int[] data)**
Windows	**void onTagMemoryReceived(byte[] data)**
Description	You can override this method to perform additional tasks associated with receiving tag memory data from specified memory bank.
Parameters	Byte Array: Tag Memory Data.
Return Value	None

Example)
RFU memory bank = 0x0000000000000000

RFU 00h		RFU 10h		RFU 20h	
0x00	0x00	0x00	0x00	0x00	0x00
RFU 30h					
0x00	0x00				

3.2.21 Read Tag Data Long

Read tag data chunk from specified memory bank.

- Command

iOS	**- (BOOL)readFromTagMemoryLong:(uint32_t)accessPassword** **epc:(NSData*)epc** **memoryBank:(uint8_t)memoryBank** **startAddress:(uint16_t)startAddress** **dataLength:(uint16_t)dataLength;**
Android	**boolean readFromTagMemoryLong(long accessPassword,** **byte[] epc,** **int memoryBank,** **int startAddress,** **int dataLength)**
Windows	**bool readFromTagMemoryLong(long accessPassword,** **byte[] epc,** **int memoryBank,** **int startAddress,** **int dataLength)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to read tag memory from specified memory bank.
Parameters	- 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)
Return Value	YES/true: Success - data: result code NO/false: Failure

Example) Reading 4kbits
accessPassword = 0x00000000
epc = 0xE2003411B802011526370494
memoryBank = 0x00 (RFU)
Start Address = 0x0000
Length = 0x0200 (Word Count)

- Notification

iOS	@optional **- (void)tagMemoryLongReceived:(NSData *)data**
Android	**void onTagMemoryLongReceived(int[] data)**
Windows	**void onTagMemoryLongReceived(byte[] data)**
Description	You can override this method to perform additional tasks associated with receiving tag memory data from specified memory bank.
Parameters	Byte Array: Tag Memory Data.
Return Value	None

Example)
1st notification: 0x0000 ~ 0x0800 (0~2K bits data)

Start Address		Word Count	USER 10h		USER 20h	
0x00	0x00	0x80	0x00	0x00	0x00	0x00
USER 30h		. . .	USER 7E0h		USER 7F0h	
0x00	0x00	0x00	0x00	0x00	0x00	0x00

2nd notification: 0x0800 ~ 0x1000 (Word Count, 2K ~ 4K bits data)

Start Address		Word Count	USER 10h		USER 20h	
0x08	0x00	0x80	0x00	0x00	0x00	0x00
USER 30h		. . .	USER 7E0h		USER 7F0h	
0x00	0x00	0x00	0x00	0x00	0x00	0x00

3.2.22 Write Tag Data

Write tag data.

- Command

iOS	**- (BOOL)writeToTagMemory:(uint32_t)accessPassword** **epc:(NSData*)epc** **memoryBank:(uint8_t)memoryBank** **startAddress:(uint16_t)startAddress** **dataToWrite:(NSData*)dataToWrite;**
Android	**boolean writeToTagMemory:(uint32_t)accessPassword** **epc:(NSData*)epc** **memoryBank:(uint8_t)memoryBank** **startAddress:(uint16_t)startAddress** **dataToWrite:(NSData*)dataToWrite**
Windows	**bool writeToTagMemory(long accessPassword,** **byte[] epc,** **int memoryBank,** **int startAddress,** **byte[] data)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to write tag data.
Parameters	- 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 RFU, 0x01 EPC, 0x02 TID, 0x03 User - startAddress: Starting Address word pointer - dataToWrite: Data to write
Return Value	YES/true: Success NO/false: Failure

Example)
accessPassword = 0x00000000
epc = 0xE2003411B802011526370494
memoryBank = 0x00 (RFU)
Start Address = 0x0000
Data to write = 0x1234567800000000 (4 word)

- Response

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode: requesting command code
Return Value	None

Example)
commandCode = 0x00

3.2.23 Generic Transport

Get data from EM Microelectronic tags.

- Command

iOS	**- (BOOL)genericTrasport:(uint32_t)accessPassword** **epc:(NSData *)epc** **ts:(uint8_t)ts** **rm:(uint8_t)rm** **sz:(uint8_t)sz** **gc:(NSData *)gc**
Android	**boolean genericTransport(long accessPassword,** **byte[] epc,** **int ts,** **int rm,** **int gcBitLen,** **byte[] gc)**
Windows	**bool genericTransport(long accessPassword,** **byte[] epc,** **int ts,** **int rm,** **int gcBitLen,** **byte[] gc)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to get data from EM Microelectronic tags.
Parameters	- accessPassword: Access Password if target memory bank was password protected. Otherwise, set AP filed to 0x00000000. - epc: Target tag's EPC - ts: Transmission parameter. Reserved as 0xC6. - rm: Response parameter. Reserved as 0x00. - gcBitLen: Generic command length in bits (excluding the ePC handle and CRC16) - gc: Generic command payload (length equal to SZ divided by 8 rounded up)
Return Value	YES/true: Success NO/false: Failure

Example)
accessPassword = 0x00000000
epc = 0x0080B0403C000000120A8A67
ts = 0x06
rm = 0x0000
gcBitLen = 18
gc = {0xE0, 0x01, 0x40} (Command: GetSensorData, don't send UID, get new sample)

- Response

iOS	**- (void)genericReceived:(NSData*)data;**
Android	**void onGenericTransportReceived(final int[] dest)**
Windows	**void onGenericTransportReceived(byte[] dest)**
Description	You can override this method to perform additional tasks associated with receiving data from EM Microelectronic tag
Parameters	Byte Array: tag data including SZ and GR. - 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)
Return Value	None

Example)
Tag Data = 0x00610032000000000000425F0D79D7
SZ = 0x0061(97 bits)
GR = 0x0032000000000000425F0D79D7
{Header(1 bit) + **Sensor data(32 bits)** + UTC(32 bits) + RN(16 bits) + CRC16(16 bits)}

- Header = 0
- Sensor data = 0x00640000 (25℃)
- UTC = 0x00000000
- RN = 0x84BE
- CRC = 0x1AF3
- Dummy = 0xAE

3.2.24 Kill Tag

Kill a Tag.

- Command

iOS	**- (BOOL)killTag:(uint32_t)killPassword** **epc:(NSData*)epc**
Android	**boolean killTag(long killPassword,** **byte[] epc)**
Windows	**bool killTag(long killPassword,** **byte[] epc)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to kill a Tag.
Parameters	- killPassword: Kill password. If killPassword field set to 0x00000000, 'Kill' command do not work. The target tag ignores it. - epc: Target tag's EPC
Return Value	YES/true: Success NO/false: Failure

Example)
killPassword = 0x87654321
epc = 0xE2003411B802011526370494

- Response

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode: requesting command code
Return Value	None

3.2.25 Lock Tag

Lock an indicated memory bank in the tag.

- Command

iOS	**- (BOOL)lockTagMemory:(uint32_t)accessPassword** **epc:(NSData*)epc** **lockData:(uint32_t)lockData**
Android	**boolean lockTagMemory(long accessPassword,** **byte[] epc,** **int lockData)**
Windows	**bool lockTagMemory(long accessPassword,** **byte[] epc,** **int lockData)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to lock an indicated memory bank in the tag.
Parameters	- 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.
Return Value	YES/true: Success NO/false: Failure

Example)
accessPassword = 0x00000000
epc = 0xE2003411B802011526370494
lockData = 0x080200
{Binary: 0000 (dummy) + 1000000000 (mask) + 1000000000 (lock data)}

- Response

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode: requesting command code
Return Value	None

3.2.26 Set Beep On

Turn the beep on/off.

- Command

iOS	**- (BOOL)setBeep:(uint8_t)on**
Android	**boolean setBeep(boolean state)**
Windows	**bool setBeep(bool state)**
Description	Returns a Boolean value that indicating whether command is forwarded reader to turn beep on/off.
Parameters	- On (0xFF) - Off (0x00)
Return Value	YES/true: Success NO/false: Failure

- Response

iOS	@optional **- (void)successReceived:(NSData *)data** **commandCode:(uint8_t)commandCode**
Android	**void onSuccessReceived(int[] data,** **int commandCode)**
Windows	**void onSuccessReceived(byte[] data,** **int commandCode)**
Description	You can override this method to perform additional tasks associated with non-parameter command acknowledge.
Parameters	data: result code - Success (0x00) commandCode: requesting command code
Return Value	None

3.2.27 Battery State

Notify battery voltage.

- Notification

iOS	**- (void)batteryStateReceived:(NSData*)data**
Android	**void onBatteryStateReceived(int[] data)**
Windows	**void onBatteryStateReceived(byte[] dest)**
Description	You can override this method to perform additional tasks associated with receiving ADC values.
Parameters	Byte Array: RCP packet response payload. - VAL (8-bit): Current value - MIN (8-bit): Minimum value of ADC - MAX (8-bit): Maximum value of ADC
Return Value	None

Example) VAL = 0x3C, MIN = 0x00, MAX = 0xFF

VAL	MIN	MAX
0x3C	0x00	0xFF

Calculating battery gauge = (VAL-MIN) / (MAX – MIN) * 100 (%)

4 Customer Service

ARETE mobile Customer Service
Phone: + 82 42 864 2402
Fax: + 82 42 864 2403
Email: sales@phychips.com

Address
PHYCHIPS Inc.
#104 Migun Technoworld 2, 187 Techno 2-ro, Yuseong-gu, Daejeon, Korea 305-500

Working Day and time
Monday to Friday
09:00~18:00(Korean Time, GMT Time + 9 hours)

Local Customer Service
Please contact where you buy.