AsReader DOCK SDK 4

SDK Reference Guide

For ASX-300R, ASX-301R, ASX-510R, ASX-520R, ASR-010D, ASR-020D, ASR-030D, ASR-031D, ASR-0230D, ASR-0231D, ASR-0240D, ASR-022D

Asterisk, Inc.

Modification

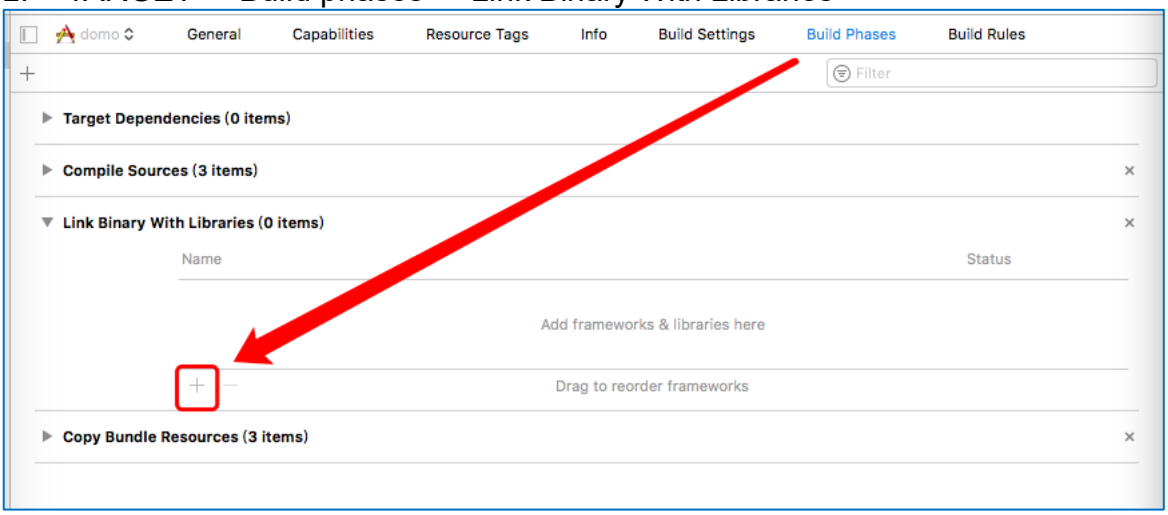
No.	Version	Modified Content	Date
1	1.2	Initial version	2018/7/23
2	1.3	getReaderInfo: Changed the parameter description	2019/1/4
3	1.4	Add some note information about 022D	2020/1/13
4	1.6	**1.** Modify functions that do not match the SDK 2. Add following functions: 5.1.27 selectParamReceived 7.28 setSelectParameter 7.29 getSelectParameter 7.30 setQueryParam	2020/7/15

Contents

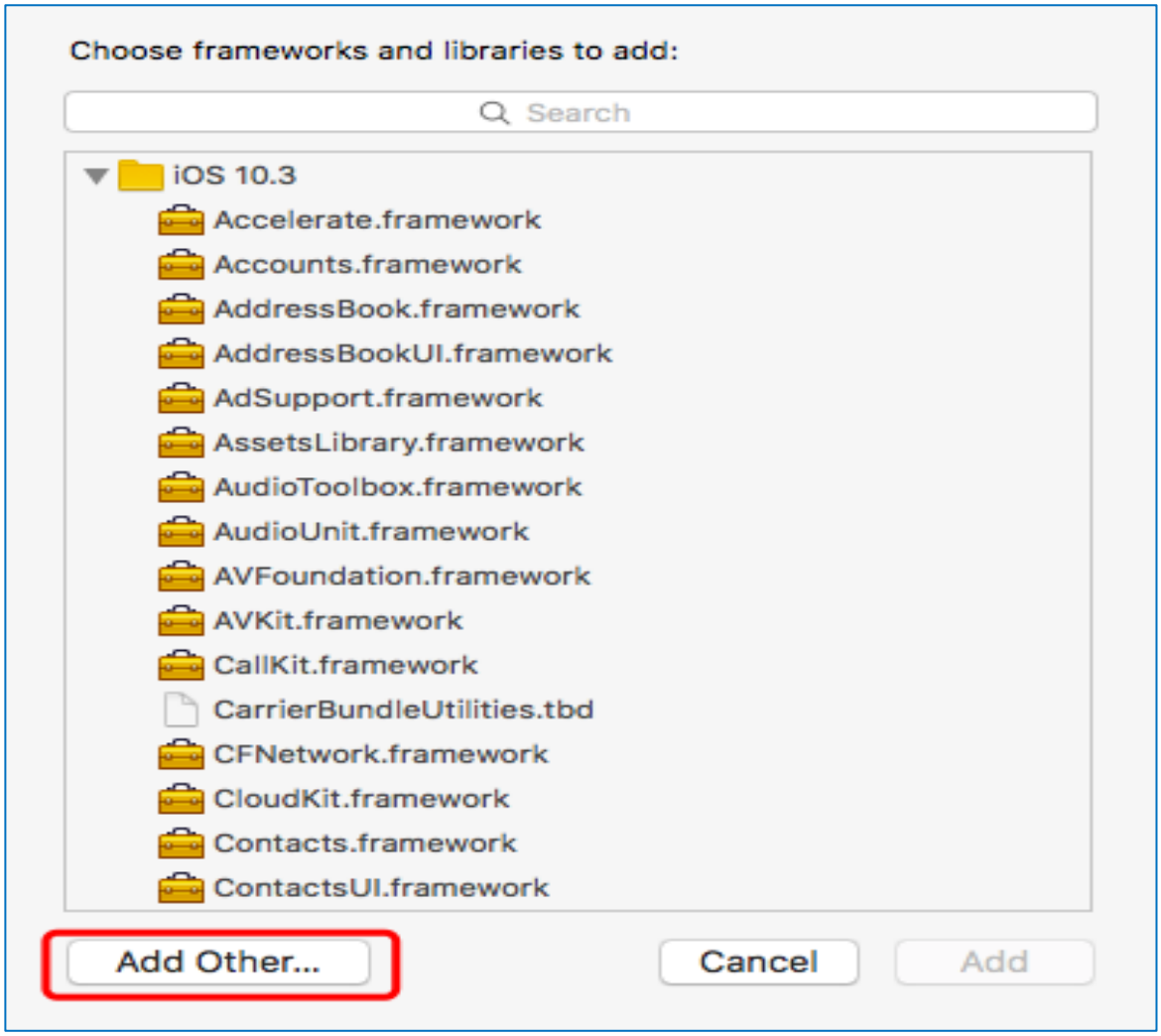
1.SDK Usage

1.1 Add SDK

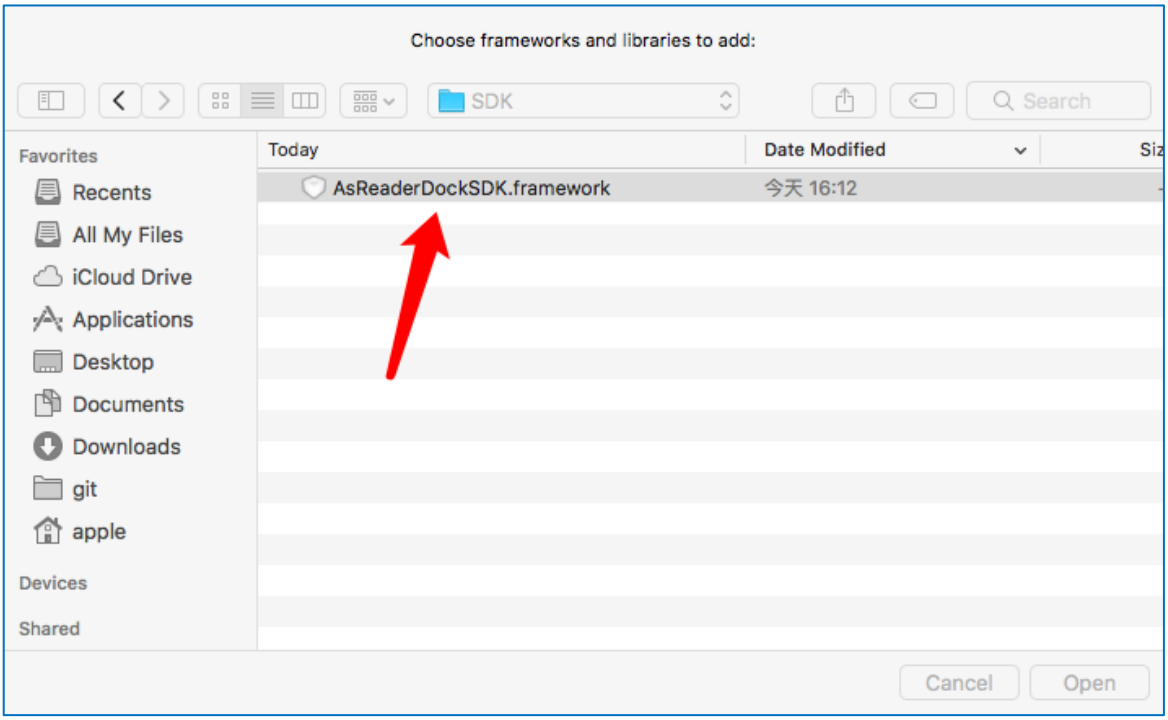
1. TARGET -> Build phases -> Link Binary With Libraries



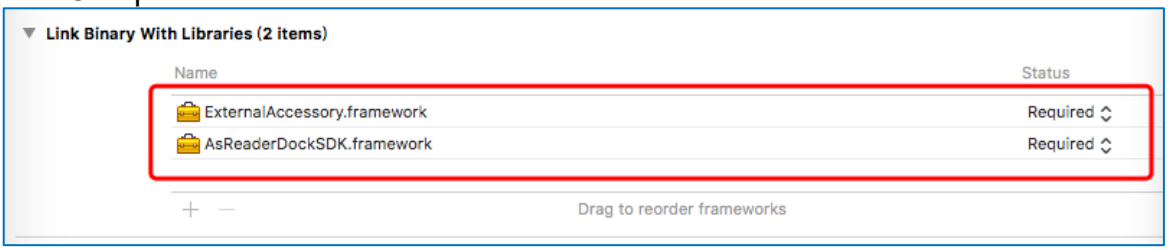
2. Select “Add Other…”



3. Add AsReaderDockSDK.framework



4. Complete as shown



1.2 Add AsReader protocol

In **Supported external accessory protocols** of plist, add the corresponding protocol to the following devices.

ASR-0230D,0240D:jp.co.asx.asreader.0240D

ASR-022D:jp.co.asx.asreader.6dongle.barcode

1.3 Use SDK in Class

Import the SDK Class header file into the Objective C project, the following is one example:
#import “AsReaderDevice.h”

1.4 Precaution

If you need to support C++ while using the SDK in Objective C, change the imported SDK header file suffix from *.m to *.mm, or import the libc++ library and compile.

2. AsReaderDevice Class

Supported AsReader:
ASX-300R,ASX-301R,ASX-510R,ASX-520R,ASR-010D,ASR-020D,ASR-030D,ASR-031D,ASR-0230D,ASR-0231D,ASR-0240D,ASR-022D

2.1 getSDKVersion

Description:Get SDK version.

```
+ (NSString*) getSDKVersion;
```

Return value:getSDKVersion(NSString),for example:1.0.0

2.2 setTriggerModeDefault

Note: This method only supports ASR-022D, ASR-0230D, ASR-0231D and ASR-0240D.

```
+ (void) setTriggerModeDefault:(BOOL)isDefault;
```

Description: Set AsReader trigger default mode.
Parameter:isDefault
YES: execute trigger default mode (scan)
NO: user custom mode (through the delegate method

2.3 getReaderInfo

Description: Send the "Get Reader Information" command to the reader to get basic information about the reader.

```
- (BOOL)getReaderInfo:(int)infoType;
```

Parameter:infoType:model(0)/ RFID Version(0x01)/ manufacturer(0x02) / frequency(0x03)/ tag type(0x04)
Return value:
YES: success
NO: failure

2.4 setBeep

```
- (BOOL)setBeep:(BOOL)beepOn
    setVibration:(BOOL)vibrationOn
setIllumination:(BOOL)illuminationOn
          setLED:(BOOL)led;
```

Description:Send the "setting" command to the reader to set the settings when reading a tag. Beep, vibration, illumination, and LED settings can be set.
Parameter:
beepOn:On (YES)/ Off (NO)
vibrationOn:On (YES)/ Off (NO)
illuminationon:On (YES)/ Off (NO)
led: ON (YES)/ led:Off (NO)
Return value:

YES: success
NO: failure

2.5 setReaderPower

Description:Set reader power on/ off with options. When the reader is set to power on, the beep, vibration, illumination, and LED settings can be set at the same time.

```
- (int)setReaderPower:(BOOL)isOn
                 beep:(BOOL)isBeep
            vibration:(BOOL)isVib
                  led:(BOOL)isLed
         illumination:(BOOL)isIllu
                 mode:(int)nDeviceType;
```

Parameter:

inOn:ON (YES)/ Off (NO)
isBeep:ON (YES)/ Off (NO)
isVib:ON (YES)/ Off (NO)
isLed:ON (YES)/ Off (NO)
isIllu:ON (YES)/ Off (NO)
nDeviceType: device type(int)

Return value: Returns 99 if nDeviceType is unknown, 1 if a command is added to the queue

2.6 setReaderPower

```
- (int)setReaderPower:(BOOL)isOn
                 beep:(BOOL)isBeep
            vibration:(BOOL)isVib
                  led:(BOOL)isLed
         illumination:(BOOL)isIllu
      connectedBeep:(BOOL)isConnectedBeep
                 mode:(int)nDeviceType;
```

Description:Set reader power on/ off with options. When the reader is set to power on, the beep, vibration, illumination, and LED settings can be set at the same time.

Parameter:

inOn:ON (YES)/ Off (NO)
isBeep:ON (YES)/ Off (NO)
isVib:ON (YES)/ Off (NO)
isLed:ON (YES)/ Off (NO)
isIllu:ON (YES)/ Off (NO)
isConnectedBeep: ON (YES)/ Off (NO)
nDeviceType: device type(int)

Return value: Returns 99 if nDeviceType is unknown, 1 if a command is added to the queue

2.7 setTagCount

Description:Send the "Set Stop Condition" command to the reader to set the

- **(void) setTagCount:(int)mtnu setSacnTime:(int)mtime setCycle:(int)repeatCycle;**

stop point of start-auto-read. This should only be used on RFID type.

Parameter:

mtnu:Maximum number of tags to read
mtime:Maximum elapsed time for tagging(sec)
repeatCycle:How many times the reader performs an inventory round

3. AsReaderBarcodeDevice Class

Supported AsReader:
ASX-510R,ASX-520R,ASR-010D,ASR-020D,ASR-0230D,ASR-0231D,ASR-0240D, ASR-022D.

3.1 startScan

Description:The reader starts scanning barcodes.

```
- (BOOL)startScan;
```

Return value:

YES: success
NO: failure

3.2 stopScan

Description:The reader stops scanning barcodes.

```
- (BOOL)stopScan;
```

Return value:

YES: success
NO: failure

3.3 doFactoryReset

```
- (BOOL)doFactoryReset;
```

Description:Factory reset (barcode mode).
Return value:

YES: success
NO: failure

3.4 setSymbologyPrefix

Note: this method only supports the Barcode mode of ASR-022D, ASR-023D, ASR-0231D, and ASR-0240D.

```
-(BOOL)setSymbologyPrefix;
```

Description: Display or hide the prefix of a barcode. (The read result for a barcode “123”, when display prefix is enabled, will be “A123”).
Return value:

YES: success
NO: the device does not support this feature or current mode is not barcode scanning mode

4. AsReaderInfo Class

4.1 Properties

```
@property(nonatomic,readonly) NSString *deviceName; / / device name
```

```
@property(nonatomic,readonly) NSString *deviceID; / / device ID
```

```
@property(nonatomic,readonly) NSString *deviceHardware; / / device H/ W
```

```
@property(nonatomic,readonly) NSString *deviceManufacturer; / / device manufacture
```

```
@property(nonatomic,readonly) NSString *deviceModelNumber; / / device model number
```

```
@property(nonatomic,readonly) NSString *deviceSerialNumber; / / device serial number
```

```
@property(nonatomic,readonly) NSString *deviceProtocol; / / device protocol
```

```
@property(readonly,assign) int currentSelectDevice; / / select current device
```

```
@property(readonly,assign) int readerType; / / reader type
```

```
@property(readonly,assign) BOOL isSmartHopping; / / smart hopping
```

```
@property(readonly, assign) BOOL isShowPrintNSLog; / / print log
```

```
@property(nonatomic,readonly) NSString *rfidModuleVersion; / / RFID module ver
```

```
@property(readonly,assign) BOOL isPowerOn; / / power on the device
@property(readonly,assign) BOOL canUseRFID; / / RFID can be used
@property(readonly,assign) BOOL canUseBarcode; / / Barcode can be used
@property(readonly,assign) BOOL canUseNFC; / / NFC can be used
@property(readonly,assign) BOOL isBeep; / / beep
@property(readonly,assign) BOOL isVibration; / / vibration
@property(readonly,assign) BOOL isLED; / / LED
@property(readonly,assign) BOOL isIllumination; / / illumination
@property(readonly,assign) BOOL isSymboloyPrefix; / / symbologyprefix
@property(readonly,assign) BOOL isTriggerModeDefault; / / trigger mode
@property(readonly,assign) float rfidpower; / / RFID power
@property(readonly,assign) float rfidPowerMax; / / RFID max power
@property(readonly,assign) float rfidPowerMin; / / RFID min power
@property(readonly,assign) int    rfidOnTime; / / RFID on time
@property(readonly,assign) int    rfidOffTime; / / RFID off time
@property(readonly,assign) int    nRFIDchannel; / / RFID channel
@property(readonly,assign) int    count; / / tag count
```

```
@property(readonly,assign) int      scanTime; / / scan time
```

```
@property(readonly,assign) int      cycle; / / scan cycle
```

```
@property(readonly,assign) int      carrierSenseTime; / / carrier sense time
```

```
@property(readonly,assign) int      targetRFPowerLevel; / / RF power level
```

```
@property(readonly,assign) int      rfidListenBeforeTalk; / / RFID LBT
```

```
@property(readonly,assign) int      rfidFrequencyHopping; / / RFID FH
```

```
@property(readonly,assign) int      rfidContinuousWave; / / RFID CW
```

5. AsReaderRFIDProtocol Class

Supported AsReader:
ASX-300R,ASX-301R,ASR-030D,ASR-031D,ASR-0230D,ASR-0231D.

```
@protocol AsReaderRFIDDeviceDelegate <NSObject>
```

5.1 AsReaderRFIDDeviceDelegate

5.1.1 pcEpcReceived

```
- (void)pcEpcReceived:(NSData *)pcEpc;
```

Description: To receive RFID tag data.
This function is called to return the execution result of the function startScan once the function startScan is called back.
Parameter: pcEpc:pcEPCdata

5.1.2 pcEpcRssiReceived

```
- (void)pcEpcRssiReceived:(NSData *)pcEpc rssi:(int)rssi;
```

Description: To receive the RFID tag data with RSSI.
This function is called to return the execution result of the function startReadTagsAndRssiWithTagNum once the function startReadTagsAndRssiWithTagNum is called back.
Parameter:
pcEpc:pcEPC data
rssi: RSSI data

5.1.3 didSetOutputPowerLevel

```
- (void)didSetOutputPowerLevel:(int)status;
```

Description: This function is called to return the execution result of the function setOutputPowerLevel once the function setOutputPowerLevel is called back.
Parameter:status:success: 0/ failure: others

5.1.4 didSetChannelParamReceived

```
- (void)didSetChannelParamReceived:(int)statusCode;
```

Description: This function is called to return the execution result of the function setChannel once the function setChannel is called back.
Parameter:statusCode:success: 0/ failure: others

5.1.5 didSetAntiCollision

```
- (void)didSetAntiCollision:(int)status;
```

Description: This function is called to return the execution result of the

function setAnticolision once the function setAnticollision is called back.
Parameter: status:success:0/ failure: others

5.1.6 didSetSession

- (void)didSetSession:(int)status;

Description: This function is called to return the execution result of the function setSession once the function setSession is called back.
Parameter: status:success:0/ failure:others

5.1.7 channelReceived

- (void)channelReceived:(int)channel channelOffset:(int)channelOffset;

Description: This function is called to return the execution result of the function getChannel once the function getChannel is called back.
Parameter:

channel: channel of rfid module
channeloffset: channel offset of rfid module

5.1.8 anticolParamReceived

- (void)anticolParamReceived:(int)mode Counter:(int)counter;

Description: This function is called to return the execution result of the function getAnticolision once the function getAnticollision is called back.
Parameter:

mode:fixed Q: 0/ dynamic Q:0x01
counter :counter value

5.1.9 txPowerLevelReceived

- (void)txPowerLevelReceived:(NSData*)power;

Description:
This function is called to return the execution result of the function getOutputPowerLevel once the function getOutputPowerLevel is called back.
Assign the Tx power of RFID to the CommonReaderInfo class.

fRFIDpower: Tx power of current RFID
fRFIDpowerMax: Maximum Tx power that can be set
fRFIDpowerMin: Minimum Tx power that can be set

5.1.10 regionReceived

- (void)regionReceived:(int)region;

Description: This function is called to return the execution result of the function getRegion once the function getRegion is called back.
Parameter:

Region: region

5.1.11 onOffTimeChanged

```
- (void)onOffTimeChanged;
```

Description: This function is called to return the execution result of the function setReadTime once the function setReadTime is called back.

5.1.12 fhLbtReceived

```
- (void)f'hLbtReceived:(NSData *)fhLb;
```

Description: This function is called to return the execution result of the function getFhLbtParameter once the function getFhLbtParameter is called back.
Parameter: fhLb: Read time (16 bits), idle time (16 bits), carrier monitoring time (16 bits), target RF power level (16 bits), FH (8 bits),LBT (8 bits), CW (8 bits)

5.1.13 hoppingTableReceived

```
- (void)hoppingTableReceived:(NSData *)table;
```

Description: This function is called to return the execution result of the function getFrequencyHoppingTable once the function getFrequencyHoppingTable is called back.
Parameter:table: table size (8bit)

5.1.14 didSetFhLbt

```
- (void)didSetFhLbt:(int)status;
```

Description: This function is called to return the execution result of the function getFrequencyHoppingTable once the function getFrequencyHoppingTable is called back.
Parameter:status:success:0/ failure: others

5.1.15 didSetOptiFreqHPTable

```
- (void)didSetOptiFreqHPTable:(int)status;
```

Description: This function is called to return the execution result of the function setFreqHoppingTable once the function setFreqHoppingTable is called back.
Parameter:status:start:0/ finish: 0x01

5.1.16 didSetFHmodeChanged

```
- (void)didSetFHmodeChanged;
```

Description: This function is called to return the execution result of the

function setFrequencyHoppingMode once the function setFrequencyHoppingMode is called back.

5.1.17 rfidModuleVersionReceived

```
- (void)rfidModuleVersionReceived;
```

Description: This function is called to return the execution result of the function getRFIDModuleVersion once the function getRFIDModuleVersion is called back.

5.1.18 rfidOnOffTimeReceived

```
- (void)rfidOnOffTimeReceived:(NSData*)data;
```

Description: This function is called to return the execution result of the function getRFIDOnOffTime once the function getRFIDOnOffTime is called back.

5.1.19 writtenReceived

```
- (void)writtenReceived:(int)statusCode;
```

Description: This function is called to return the execution result of the function writeTagMemoryWithEPC once the function writeTagMemoryWithEPC is called back.
Parameter: statusCode: success:(0)/ failure: others

5.1.20 sessionReceived

```
- (void)sessionReceived:(int)session;
```

Description: This function is called to return the execution result of the function getSession once the function getSession is called back.
Parameter: session: S0(0), S1(1), S2(2), S3(3), Dev. mode(240)

5.1.21 tagMemoryReceived

```
- (void)tagMemoryReceived:(NSData *)data;
```

Description: This function is called to return the execution result of the function readTagWithAccessPassword once the function readTagWithAccessPassword is called back.
Parameter: data: memory information of tag

5.1.22 killedReceived

```
- (void)killedReceived:(int)statusCode;
```

Description: This function is called to return the execution result of the function killTagWithPassword once the function killTagWithPassword is called

back.
Parameter: statusCode: success: 0/ failure: others

5.1.23 lockedReceived

- (void)lockedReceived:(int)statusCode;

Description: This function is called to return the execution result of the function lockTagMemoryWithAccessPassword once the function lockTagMemoryWithAccessPassword is called back.
Parameter:statusCode: success: 0/ failure: others

5.1.24 responseReboot

- (void)responseReboot:(int)status;

Description: This function is called to return the result of the device restart after the device enters the restart. (firmware update)
Parameter:status:success: 0/ failure: others

5.1.25 updatedRegistry

- (void)updatedRegistry:(int)statusCode;

Description: This function is called to return the execution result of the function updateRegistry once the function updateRegistry is called back.
Parameter:statusCode:success: 0/ failure: others

5.1.26 pcEpcSensorDataReceived

- (void)pcEpcSensorDataReceived:(NSData *)pcEpc sensorData:(NSData *)sensorData;

Description: This function is called to return the execution result of the function startReadTagsRFM once the function startReadTagsRFM is called back.
Parameter:

pcEpc: Temperature tag/ Humidity tag data
sensorData: Temperature/ Humidity data

```
- (void)pcEpcSensorDataReceived:(NSData *)pcEpc sensorData:(NSData *)sensorData
{
    int codeType;/ / Tag type, 2 (Humidity tag)/ 3 (Temperature tag)
    int onChipRssiCodeValue;/ / Tag chip RSSI data
    int sensorCodeValue;/ / Temperature / Humidity data (Hex)
    double calcTemp;/ / Temperature(Celsius scale)
    NSMutableString *tmptagid;/ / Tag pcepc data (Hex)
    NSData *tagid = pcEpc;
    NSData *taghex = sensorData;
    / / pcepc NSData转NSString
    tmptagid = [[NSMutableString alloc] init];
    unsigned char* ptrtagid= (unsigned char*) [tagid bytes];
    for(int i = 0; i < tagid.length; i++)
        [tmptagid appendFormat:@"%02X", *ptrtagid++ & 0xFF ];

    / / Temperature/ Humidity data parsing
    Byte *b = (Byte*) [taghex bytes];
    codeType = b[0];
    onChipRssiCodeValue = (b[1] << 8) | b[2];
    sensorCodeValue = (b[3] << 8) | b[4];
    double code1 = 0;
    double temp1 = 0;
    double code2 = 0;
    double temp2 = 0;
    double tempCode = sensorCodeValue;
    if (codeType == 3) {
        int temp = b[7] << 4;
        code1 = temp + ((b[8] >> 4) & 0x0F);
        temp = (b[8] & 0x0F) << 7;
        temp1 = temp + ((b[9] >> 1) & 0x7F);
        temp = (b[9] & 0x01) << 8;
        temp = (temp + b[10]) << 3;
        code2 = temp + ((b[11] >> 5) & 0x07);
        temp = (b[11] & 0x1F) << 6;
        temp2 = temp + ((b[12] >> 2) & 0x3F);
        calcTemp = ((temp2 - temp1) / (code2 - code1) * (tempCode - code1) + temp1 -
800) / 10;
    }
}
```

5.1.27 pcEpcSensorDataReceived

- (void)selectParamReceived: (NSData *)selParam;

Description: This function is called to return the execution result of the function getSelectParam once the function getSelectParam is called back.
Parameter:

selParam:

Target (3 bit)
Action (3 bit)
Bank (2 bit)
Offset (32 bit)
Length (8 bit)
Truncation (1 bit)
Reserve (7 bit)
Mask (0~255 bit)

6. AsReaderNFCProtocol Class

Supported AsReader: ASR-0240D.

```
@protocol AsReaderNFCDeviceDelegate <NSObject>
```

6.1 AsReaderNFCDeviceDelegate

6.1.1 nfcDataReceived

```
- (void)nfcDataReceived:(NSData *)data;
```

Description:This function is called when nfc tag data is received.
Parameter:data:NFC tag data

7. AsReaderRFIDDevice Class

Supported Asreader:
ASX-300R,ASX-301R,ASR-030D,ASR-031D,ASR-0230D,ASR-0231D.

7.1 stopScan

```
- (BOOL)stopScan;
```

Description:Stop reading tags.
Return value:

YES: success
NO: failure

7.2 startReadTagAndTIDwithtagNum

```
- (BOOL)startReadTagAndTidWithTagNum:(int)maxTags
                             maxTime:(int)maxTime
                         repeatCycle:(int)repeatCycle;
```

Description:Start an automatic tag read operation, tag IDs with TID are sent back to user though notification packets.
Parameter:

maxTags:Maximum number of tags to read
maxTime:Maximum elapsed time to read tags (sec)
repeatCycle:How many times the reader performs an inventory round

Return value:

YES: success
NO: failure

7.3 getChannel

```
- (BOOL)getChannel;
```

Description:Send the "Get current RF Channel" command to the reader to get the RF channel. This command is valid only for non-FH mode.
Return value:

YES: success
NO: failure

7.4 setChannel

```
- (BOOL)setChannel:(int)channel
      channelOffset:(int)channelOffset;
```

Description:Send the "Set current RF Channel" command to the reader to set the RF channel. This command is valid only for non-FHSS mode.
Parameter:

channel : Channel number. The range of channel number depends on regional settings
channelOffset : Channel number offset for miller subcarrier.

Return value:

YES: success
NO: failure

7.5 getFhLbtParameter

```
- (BOOL)getFhLbtParameter;
```

Description:To get the parameters of FH and LBT.

Return value:

YES: success
NO: failure

7.6 getOutputPowerLevel

```
- (BOOL)getOutputPowerLevel;
```

Description:Send the "Get Tx Power Level" command to the reader to get the current, minimum, and maximum Tx power level.

Return value:

YES: success
NO: failure

7.7 setOutputPowerLevel

Description:Send the "Get Tx Power Level" command to the reader to set the

```
- (BOOL)setOutputPowerLevel:(int)powerLevel;
```

current, minimum, and maximum Tx power level.

Parameter:powerLevel:Tx power

Return value:

YES: success
NO: failure

7.8 writeTagMemoryWithAccessPassword

```
-(BOOL)writeTagMemoryWithAccessPassword:(int)accessPassword
                                    epc:(NSData *)epc
                           memoryBank:(int)memoryBank
                           startAddress:(int)startAddress
                           dataToWrite:(NSData*)dataToW
                           rite;
```

Description:To write the data of the tag

Parameter:

accessPassword:access password 00000000
epc:the EPC of tag
memoryBank: RFU(0) / EPC(1) / TID(2) / User(3)
startAddress: starting address
dataToWrite:data to write

Return value:

YES: success
NO: failure

7.9 killTagWithPassword

```
- (BOOL)killTagWithPassword:(int)password
                        epc:(NSData *)epc;
```

Description:To kill the tag.
Note:Must set kill password before killing tag.

Parameter:

password: password. (The tag will not be killed if the password is set to “00000000”.)
epc:Target tag’s EPC

Return value:

YES: success
NO: failure

7.10 lockTagMemoryWithAccessPassword

```
- (BOOL)lockTagMemoryWithAccessPassword:(int)accessPassword
                                    epc:(NSData *)epc
                               lockData:(int)lockData;
```

Description:To log the tag.
Note:Be sure to set the access password before locking tag.

Parameter:

accessPassword(The tag will not be locked if the password is set to “00000000”.)
epc:the EPC of tag
lockData:Lock data

Return value:

YES: success
NO: failure

7.11 getSession

```
- (BOOL)getSession;
```

Description:Send the "Get Session" command to the reader to get the current session.

Return value:

YES: success
NO: failure

7.12 setSession

- (BOOL)setSession:(int)session;

Description:Send the "Set Session" command to the reader to set the current session.
Parameter:session: S0:0/ S1:0x01/ S2:0x02/ S3:0x03/ Dev.mode:0xF0
Return value:

YES: success
NO: failure

7.13 getAnticollision

- (BOOL)getAnticollision;

Description:Send the "get Anti-Collision Mode" command to the reader to get the Anti-collision algorithm.
Return value:

YES: success
NO: failure

7.14 setAnticollision

- (BOOL)setAnticollision:(int)mode
Counter:(int)counter;

Description:Send the "Set Anti-Collision Mode" command to the reader to set the Anti-collision algorithm.
Parameter:

mode:Anti-collision Mode (8-bit), fixed Q: 0/ Dynamic Q:0x01
counter:change target at N-th Tx On according to inventory round result(default:1)

Return value:

YES: success
NO: failure

7.15 updateRegistry

- (BOOL)updateRegistry;

Description:Update registry.
Return value:

YES: success
NO: failure

7.16 getRFIDModuleVersion

```
- (BOOL)getRFIDModuleVersion;
```

Description:Send the "Get Reader Information" command to the reader to get basic information from the reader.
Return value:
YES: success
NO: failure

7.17 setHoppingOnOff

```
- (BOOL)setHoppingOnOff:(BOOL)isOn;
```

Description:Send the "Set FH and LBT Parameters" command to the reader. Only set frequencyHopping and listenBeforeTalk in FH and LBT Parameters. continuousWave is continuousWave 0.
Parameter: isOn:
YES:Set frequencyHopping is 2 and listenBeforeTalk is 1.
NO:Set frequencyHopping is 1 and listenBeforeTalk is 2.
Return value:
YES: success
NO: failure

7.18 writeTagMemoryWithEPC

```
- (BOOL)writeTagMemoryWithEPC:(NSData *)epc
                dataToWriteAscii:(NSString *)dataToWrite;
```

Description: Send the "Write Type C Tag Data" command to the reader to write type C tag data.
Parameter:
epc: target tag's EPC
dataToWrite:data to write
Return value:
YES: success
NO: failure

7.19 readTagWithAccessPassword

```
- (BOOL)readTagWithAccessPassword:(int)accessPassword
                              epc:(NSData *)epc
                       memoryBank:(int)memoryBank
                     startAddress:(int)startAddress
                       dataLength:(int)dataLength;
```

Description:To read the Type C tag data of specified memory

Parameter:

accessPassword:Access password
epc:Tag
memoryBank:RFU (0) / EPC (1) / TID (2) / User (3)
startAddress:The start address
dataLength: Length of the data

Return value:

YES: success
NO: failure

7.20 setOptimumFrequencyHoppingTable

- (BOOL)setOptimumFrequencyHoppingTable;

Description:Set optimum frequency hopping table.

Return value:

YES: success
NO: failure

7.21 getFrequencyHoppingMode

- (BOOL)getFrequencyHoppingMode;

Description:Send the "Get Frequency Hopping Mode" command to the reader to get frequency hopping mode.

Return value:

YES: success
NO: failure

7.22 getStopCondition

- (BOOL)getStopCondition;

Description:Send the "Get Stop Condition" command to the reader to get the stop point of start-auto-read.

Return value:

YES: success
NO: failure

7.23 setSmartHoppingOnOff

- (BOOL)setSmartHoppingOnOff:(BOOL)isOn;

Description:Send the "Set Frequency Hopping Mode" command to the reader to set frequency hopping mode.

Parameter:isOn:FH mode;(YES:smart hopping mode/ NO: normal mode)

Return value:

YES: success
NO: failure

7.24 getRegion

```
- (BOOL)getRegion;
```

Description:To get the region information.

7.25 startReadTagsRFM

```
- (BOOL)startReadTagsRFM:(int)codeType
                 maxTags:(int)maxTags
                 maxTime:(int)maxTime
             repeatCycle:(int)repeatCycle;
```

Description: Enable AsReader to read RFID temperature and humidity tags
Parameter:

codeType: The type of tag to read. Temperature tag: 3/ Humidity tag: 2
mtnu: The maximum number of tags to read
mtime: The maximum elapsed time (Unit: second)
repeatCycle: The maximum number of inventory cycles.

Return value:

YES: success
NO: failure

7.26 setReadTime

```
- (BOOL)setReadTime:(int)ReadTime
            idleTime:(int)IdleTime;
```

Description: Set the read time and the idle time.
Parameter:

ReadTime: Read time (ms)
IdleTime: Idle time (ms)

Note: The function setFhLbtParameter is recommended when setting On/ OffTime and Hopping in turn.
Return value:

YES: success
NO: failure

7.27 setFhLbtParameter

```
- (BOOL)setFhLbtParameter:(int)ReadTime
                 idleTime:(int)IdleTime
          carrierSenseTime:(int)carrierSenseTime
       targetRFPowerLevel:(int)targetRFPowerLevel
          frequencyHopping:(int)frequencyHopping
          listenBeforeTalk:(int)listenBeforeTalk
            continuousWave:(int)continuousWave;
```

Description: Set the parameters FH and LBT.

Parameter:

ReadTime: Read time(ms)。
IdleTime: Idle time (ms)。
carrierSenseTime: Carrier listening time. Fixed value: 50
targetRFPowerLevel: Target RF power. Fixed value: -740
frequencyHopping: Enable: 1 or above/ Disable: 0
listenBeforeTalk: Enable: 1 or above/ Disable: 0
continuousWave: Fixed value: 0

Note:

To enable Hopping, the value of the parameter frequencyHopping needs to be set to 2 and the value of the listenBeforeTalk needs to be set to 1.
To disable Hopping, the value of the frequencyHopping parameter needs to be set to 1 and the value of the listenBeforeTalk needs to be set to 2.

Return value:

YES: success
NO: failure

7.28 setSelectParameter

```
- (BOOL)setSelectParameter:(int)target
                    action:(int)action
                  memoryBank:(int)memoryBank
                    pointer:(int)pointer
                     length:(int)length
                   truncate:(int)truncate
                       mask:(NSData *)mask;
```

Description: Set the function of data mask when scanning code.

Parameter:

target: session: S0 (000b), S1 (001b), S2 (010b), S3 (011b), SL (100b)
action: Standard ISO18000-6C
memoryBank: Bank. RFU(00b), EPC(01b), TID(10b), User(11b)
pointer: The offset address of the mask
length: The data length of the mask
truncate: Used to control whether tag data that meets mask criteria is truncated. 0 means no truncation.
mask: The mask data

Return value:

YES: success
NO: failure

7.29 getSelectParameter

- (BOOL)getSelectParameter;

Description: Gets the configuration parameters for AsReader's “Select” function.

Return value:

YES: success

NO: failure

Delegate:

No.	Function	Description	Parameter	Parameter Value
5.1.27	selectParamReceived	Return configuration parameters	selParam	**The data structur**e: Target (3bit), Action (3bit), Memory Bank (2bit), Pointer (32bit), length (8bit), Truncate (1bit), reserve (7bit), Mask (0~255 bit)

7.30 setQueryParam

- (BOOL)setQueryParam:(int)divideRatio
m:(int)m
trext:(int)trext
selection:(int)selection
session:(int)session
target:(int)target
qValue:(int)qValue;

Description: Set the parameters for the query.

Parameter:

dr: DR=8(0), DR=64/3 (1)

m: M=1 (0), M=2 (1), M=4 (2), M=8 (3)

trext: No pilot tone(0), Use pilot tone(1)

sel: All(0 or 1), ~SL(2), SL(3)

session: S0(0), S1(1), S2(2), S3(3)

target: A(0), B(1)

q: Range: 0-15. 2_q is the number of slots per inventory cycle.

Return value:

YES: success

NO: failure

8. AsReaderDeviceProtocol Class

Supported AsReader:
ASX-300R,ASX-301R,ASX-510R,ASX-520R ,ASR-010D,ASR-020D,ASR-030D,ASR-031D,ASR-0230D,ASR-0231D,ASR-0240D,ASR-022D.

8.1 AsReaderDeviceProtocol

```
@protocol AsReaderDeviceProtocol <NSObject>
```

8.1.1 responsePowerOnOff

```
- (void)responsePowerOnOff:(BOOL)isOn
HWModeChange:(BOOL)isHWModeChange;
```

Description:This function is called when the reader sends a response code to "setReaderPower".
Parameter:
isOn: Power on (YES), Power off (NO)
isHWModeChange:Indicates whether HW mode has changed

8.1.2 responsePowerOnOff

```
- (void)releasedTriggerButton;
```

Description:This function is called when the trigger button of the reader is released.

8.1.3 plugged

```
- (void)plugged:(BOOL)plug;
```

Description:This function is called when the plug state between the reader and iPhone changes.
Parameter: plug:plugged: YES/ unplugged: NO

8.1.4 readerConnected

```
- (void)readerConnected:(int)status;
```

Description:Notification from the module about "Power Reset". This function is called when the reader's connection status changes.
Parameter:status:connected:255/ disconnected:0

8.1.5 pushedTriggerButton

```
- (void)pushedTriggerButton;
```

Description:This function is called when the trigger button of the reader is pressed.

8.1.6 receivedScanData

```
-(void)receivedScanData:(NSData *)readData
                DeviceType:(int)nDeviceType;
```

Description:This function is called when tag data is received.
Parameter:
readData: tag data
nDeviceType: unknown: 99/ barcode:0 / RFID:1/ NFC:2

8.1.7 allDataReceived

```
- (void)allDataReceived:(NSData *)data;
```

Description:This function is called when tag data (all types) is received.
Parameter:data: tag data

8.1.8 batteryReceived

```
- (void)batteryReceived:(int)battery;
```

Description: This function is called when the battery level of reader is received.
Parameter:battery: battery level

8.1.9 onAsReaderTriggerKeyEventStatus

```
- (void)onAsReaderTriggerKeyEventStatus:(NSString*)status;
```

Description:Response the status when the trigger key is being pressing.
Parameter:status:status

8.1.10 errorReceived

```
- (void)errorReceived:(NSData *)errorCode;
```

Description:Response to an invalid command.
Parameter:errorCode: payload (error code, command code, sub error code)

9. AsReaderNFCDevice Class

Supported AsReader:ASR-0240D.

```
#define NFC_CMD_INVENTORYSET {0x02, 0x00, 0x6F, 0x02, 0x03, 0xE8, 0x03,0x61, 0x0D}
#define NFC_CMD_STARTSCAN {0x02, 0x00, 0x4E, 0x07, 0x00, 0x51, 0x0F, 0x80, 0xFF, 0xFF, 0x00, 0x03, 0x38, 0x0D}
#define NFC_CMD_STOPSCAN {0x02, 0x00, 0x4E, 0x07, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x03, 0xDA, 0x0D}
```

NFC_CMD_INVENTORYSET: command to take inventory
NFC_CMD_STARTSCAN: command to start scanning
NFC_CMD_STOPSCAN: command to stop scanning

9.1 sendData

```
- (BOOL)sendData:(NSData *)sendData;
```

Description:Send data to the reader.
Parameter:sendData: send data
Return value:
YES: success
NO: failure

9.2 startScan

```
- (BOOL)startScan;
```

Description:NFC type reader starts to scan tags.
Return value:
YES: success
NO: failure

9.3 stopScan

```
- (BOOL)stopScan;
```

Description:NFC type reader stops scanning tags.
Return value:
YES: success
NO: failure

10. AsReaderBarcodeProtocol Class

Supported AsReader:
ASX-510R,ASX-520R,ASR-010D,ASR-020D,ASR-0230D,ASR-0231D,ASR-0240D,ASR-022D.

10.1 barcodeDataReceived

```
- (void)barcodeDataReceived:(NSData *)data;
```

Description: To receive the barcode data.
This function is called to return the execution result of the function startScan once the function startScan is called back.
Parameter: data: barcode data

10.2 receiveFactoryReset

```
- (void)receiveFactoryReset:(int)status;
```

Description: This function is called to return the execution result of the function doFactoryReset once the function doFactoryReset is called back.
Parameter:status: reset start: 0/ reset complete: 255