



AsReader DOCK SDK 4

SDK Reference Guide V1.3

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

Modification

No.	Version	Modified Content	Date
1	1.2	Initial version	2018/07/23
2	1.3	getReaderInfo: Changed the parameter description	2019/01/04

Contents

1.	SDK Usage	6
1.1	Add SDK	6
1.2	Add AsReader protocol	7
1.3	Use SDK in Class.....	7
1.4	Precaution.....	7
2.	AsReaderDevice Class	8
2.1	getSDKVersion.....	8
2.2	setTriggerModeDefault.....	8
2.3	getReaderInfo	8
2.4	setBeep	8
2.5	setReaderPower	9
2.6	setReaderPower	9
2.7	setTagCount.....	10
3.	AsReaderBarcodeDevice Class	11
3.1	startScan	11
3.2	stopScan	11
3.3	doFactoryReset.....	11
3.4	setSymbologyPrefix	11
4.	AsReaderInfo Class	12
4.1	Properties.....	12
5.	AsReaderRFIDProtocol Class.....	15
5.1	AsReaderRFIDDeviceDelegate	15
5.1.1	pcEpcReceived	15
5.1.2	pcEpcRssiReceived	15
5.1.3	rssiReceived	15
5.1.4	didSetOutputPowerLevel.....	15
5.1.5	didSetChannelParamReceived	15
5.1.6	didSetAntiCollision	16
5.1.7	didSetSession	16
5.1.8	channelReceived	16
5.1.9	anticolParamReceived	16
5.1.10	txPowerLevelReceived	16
5.1.11	onOffTimeChanged.....	16
5.1.12	fhLbtReceived	17
5.1.13	hoppingTableReceived.....	17
5.1.14	didSetFhLbt.....	17
5.1.15	didSetOptiFreqHPTable	17
5.1.16	didSetFHmodeChanged	17
5.1.17	rfidModuleVersionReceived	17
5.1.18	rfidOnOffTimeReceived	18
5.1.19	writtenReceived	18
5.1.20	sessionReceived.....	18
5.1.21	tagMemoryReceived.....	18
5.1.22	killedReceived.....	18
5.1.23	lockedReceived	18

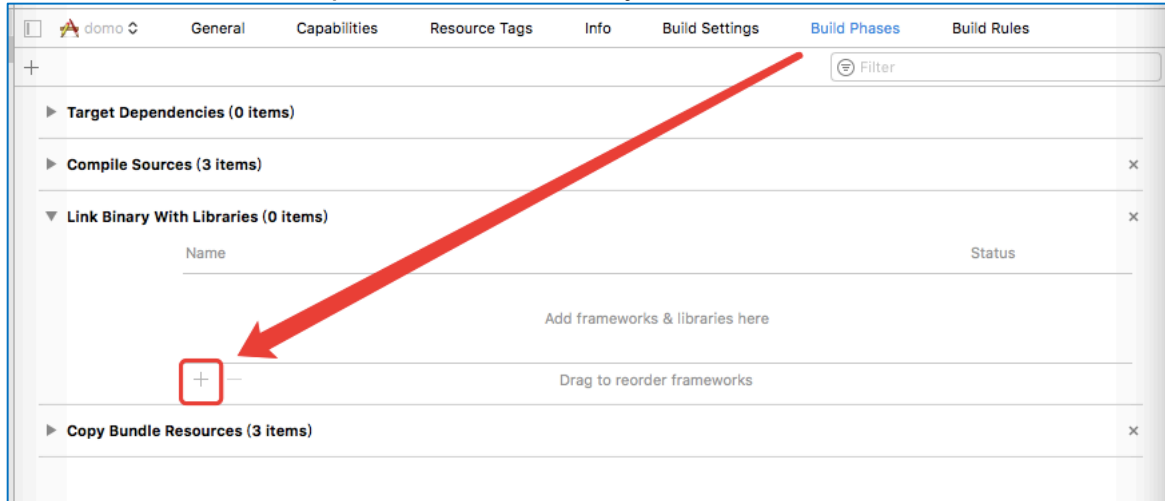
5.1.24	responseReboot	18
5.1.25	updatedRegistry.....	19
6.	AsReaderNFCProtocol Class.....	20
6.1	AsReaderNFCDeviceDelegate	20
6.1.1	nfcDataReceived	20
7.	AsReaderRFIDDevice Class	21
7.1	stopScan	21
7.2	startReadTagAndTIDwithtagNum	21
7.3	getChannel.....	21
7.4	setChannel.....	21
7.5	getFhLbtParameter	22
7.6	getOutputPowerLevel	22
7.7	setOutputPowerLevel.....	22
7.8	writeTagMemoryWithAccessPassword	23
7.9	killTagWithPassword	23
7.10	lockTagMemoryWithAccessPassword	23
7.11	getSession	24
7.12	setSession	24
7.13	getAnticollision	24
7.14	setAnticollision	24
7.15	updateRegistry	25
7.16	getRFIDModuleVersion	25
7.17	setHoppingOnOff	25
7.18	writeTagMemoryWithEPC	25
7.19	readTagWithAccessPassword	26
7.20	setOptimumFrequencyHoppingTable.....	26
7.21	getFrequencyHoppingMode.....	26
7.22	getStopCondition	26
7.23	setSmartHoppingOnOff.....	27
7.24	getRegion.....	27
8.	AsReaderDeviceProtocol Class	28
8.1	AsReaderDeviceProtocol	28
8.1.1	responsePowerOnOff	28
8.1.2	responsePowerOnOff	28
8.1.3	plugged.....	28
8.1.4	readerConnected.....	28
8.1.5	pushedTriggerButton	28
8.1.6	receivedScanData	29
8.1.7	allDataReceived	29
8.1.8	batteryReceived	29
8.1.9	onAsReaderTriggerKeyEventStatus.....	29
8.1.10	errReceived.....	29
9.	AsReaderNFCDevice Class	30
9.1	sendData.....	30
9.2	startScan.....	30
9.3	stopScan	30
10.	AsReaderBarcodeProtocol Class	31
10.1	barcodeDataReceived	31

10.2	receiveFactoryReset.....	31
------	--------------------------	----

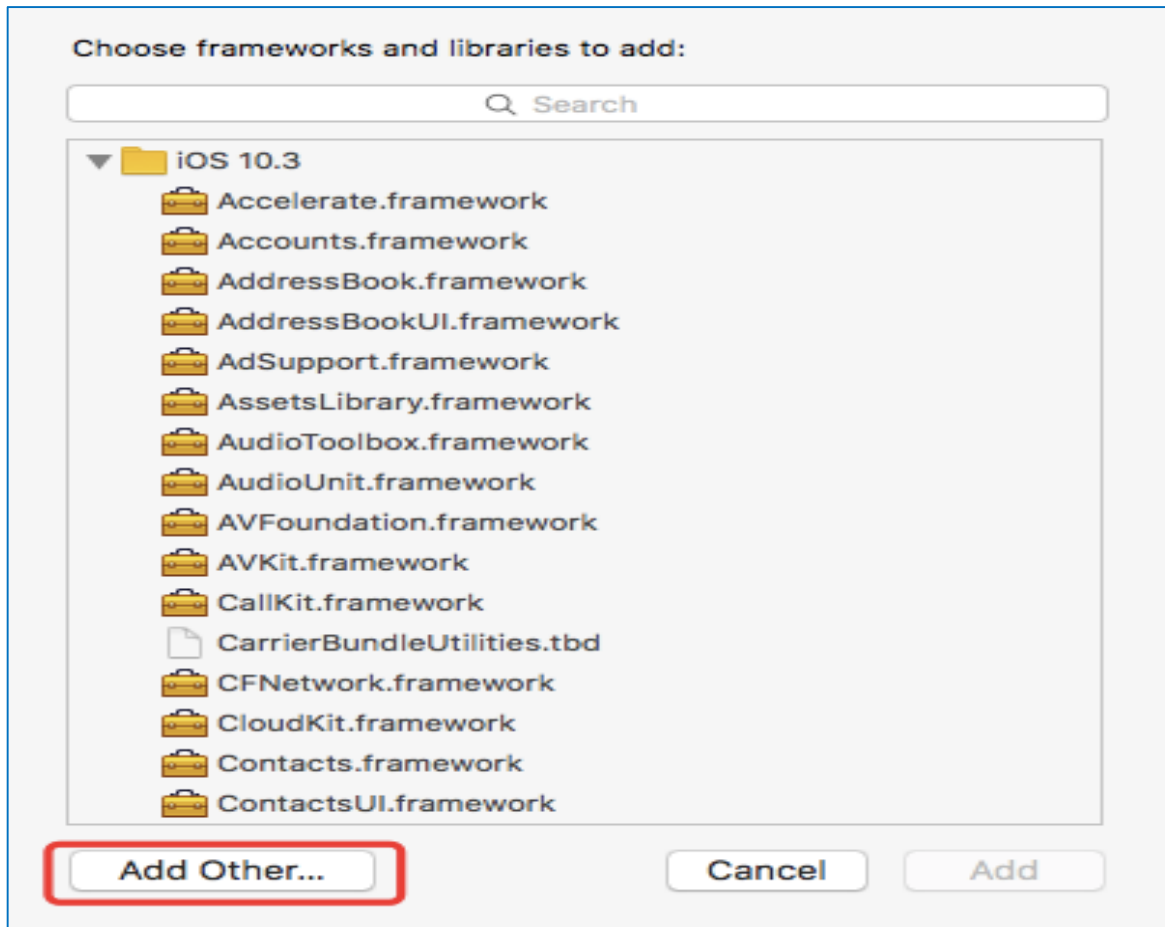
1.SDK Usage

1.1 Add SDK

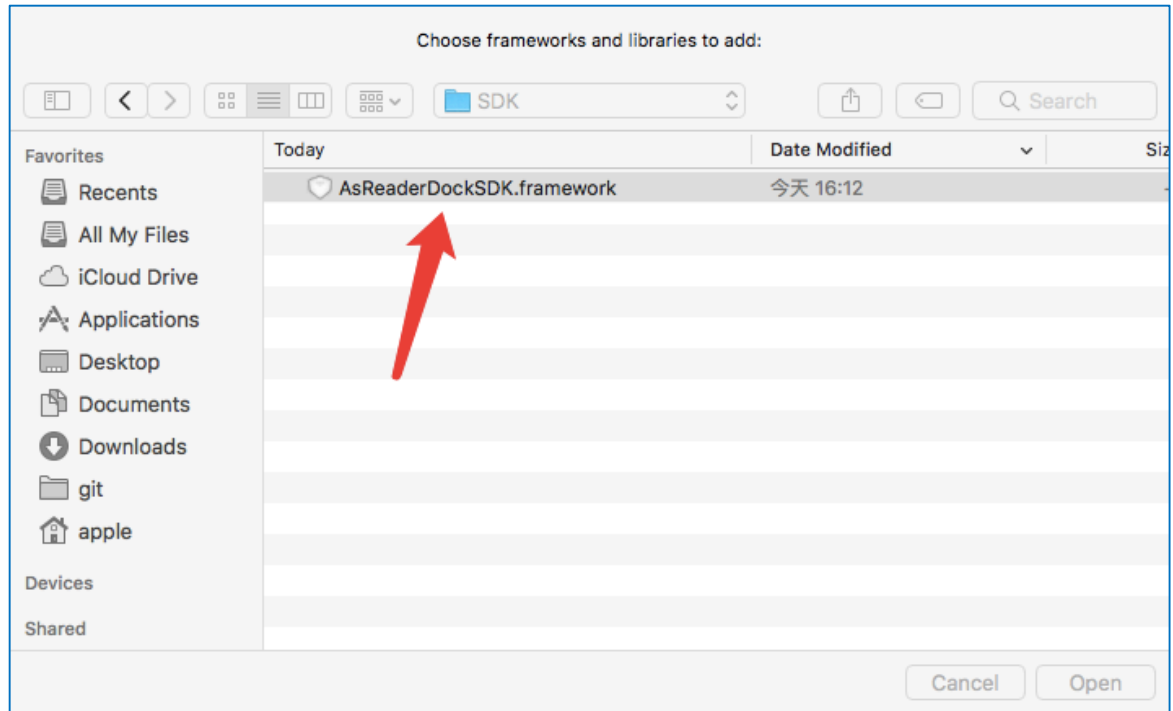
1.1.1 TARGET -> Build phases -> Link Binary With Libraries



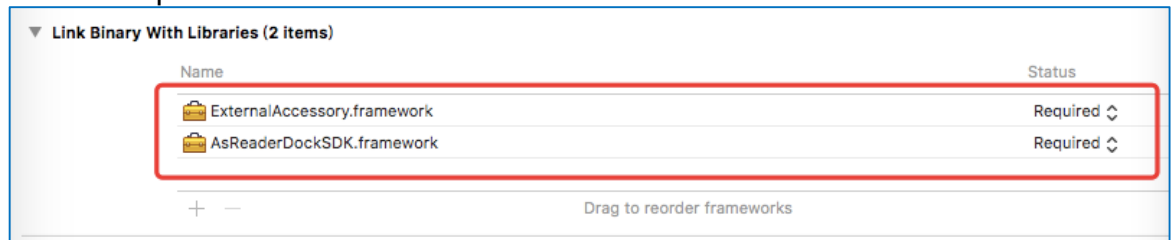
1.1.2 Select “Add Other...”



1.1.3 Add AsReaderDockSDK.framework



1.1.4 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



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

2.1 getSDKVersion

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

Description:Get SDK version.

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

2.2 setTriggerModeDefault

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

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

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

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

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

Parameter:infoType:model(0x00)/ RFID Version(0x01)/manufacturer(0x02) /frequency(0x03)/tag type(0x04)

Return value:

YES: success

NO: failure

2.4 setBeep

```
- (BOOL)setBeep:(int)beepOn  
  setVibration:(int)vibrationOn  
  setIllumination:(int)illuminationOn  
  setLED:(int)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:1/Off:0
vibrationOn:On:1/Off:0
illuminationon:On:1/Off:0
led: ON:1/led:Off:0

Return value:

YES: success
NO: failure

2.5 setReaderPower

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

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

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

3.1 startScan

- (BOOL)startScan;

Description:The reader starts scanning barcodes.

Return value:

YES: success

NO: failure

3.2 stopScan

- (BOOL)stopScan;

Description:The reader stops scanning barcodes.

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

-(BOOL)setSymbologyPrefix;

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

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(nonatomic, readonly) NSString *region; //region
```

```
@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; //symboloyprefix
```

```
@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
```

AsReader

@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: This function is called when tag data is received.

Parameter: pcEpc: pcEPC data

5.1.2 pcEpcRssiReceived

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

Description: This function is called when tag data with RSSI is received.

Parameter:

pcEpc: pcEPC data

rssi: RSSI data

5.1.3 rssiReceived

```
- (void)rssiReceived:(int)rssi;
```

Description: This function is called when tag rssi is received.

Parameter: rssi: rssi data

5.1.4 didSetOutputPowerLevel

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

Description: This function is called when a response code to "Set Output Power Level" is received.

Parameter: status: success: 0x00/failure: others

5.1.5 didSetChannelParamReceived

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

Description: This function is called when a response code to "Set current RF Channel" is received.

Parameter: statusCode: success: 0x00/failure: others

5.1.6 didSetAntiCollision

- (void)didSetAntiCollision:(int)status;

Description:This function is called when a response code to "Set Anti-Collision Mode" is received.

Parameter: status:success:0x00/failure: others

5.1.7 didSetSession

- (void)didSetSession:(int)status;

Description:This function is called when a response code to "Set Session" is received.

Parameter: status:success:0x00/failure:others

5.1.8 channelReceived

- (void)channelReceived:(uint8_t)channel
channelOffset:(uint_8)channelOffset;

Description:This function is called when a response code to "Get current RF Channel" is received.

Parameter:

channel: channel of rfid module

channeloffset: channel offset of rfid module

5.1.9 anticolParamReceived

- (void)anticolParamReceived:(uint8_t)mode Counter:(uint_8)counter;

Description: This function is called when a response code to "Get Anti-Collision Mode" is received.

Parameter:

mode:fixed Q: 0x00/dynamic Q:0x01

counter :counter value

5.1.10 txPowerLevelReceived

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

Description:This function is called when a response code to "Get Tx Power Level" is received.

Parameter:power : Payload (Power 2bytes, MinPower 2bytes, Max Power 2bytes)

5.1.11 onOffTimeChanged

- (void)onOffTimeChanged;

Description:This function is called when a response code to "Set FH and LBT Parameters" is received.

5.1.12 fhLbtReceived

```
- (void)fhLbtReceived:(NSData *)fhLb;
```

Description:Response of "getFhLbtParam".

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 when a response code to "Get Frequency Hopping Table" is received.

Parameter:table: table size (8bit)

5.1.14 didSetFhLbt

```
- (void)didSetFhLbt:(uint8_t)status;
```

Description:This function is called when a response code to "Set FH and LBT Parameters" is received.

Parameter:status:success:0x00/failure: others

5.1.15 didSetOptiFreqHPTable

```
- (void)didSetOptiFreqHPTable:(uint_8)status;
```

Description: This function is called when a response code to "Set Optimum Frequency Hopping Table" is received.

Parameter:status:start:0x00/finish: 0x01

5.1.16 didSetFHmodeChanged

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

Description:This function is called when a response code to "Set Frequency Hopping Mode" is received.

5.1.17 rfidModuleVersionReceived

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

Description:This function is called when a response code to "Get Reader Information" is received.

5.1.18 rfidOnOffTimeReceived

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

Description: This function is called when a response code to "Get FH and LBT Parameters" is received.

5.1.19 writtenReceived

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

Description: Response of "writeTagMemoryWithEPC".

Parameter: statusCode: success:(0)/failure: others

5.1.20 sessionReceived

```
- (void)sessionReceived:(uint8_t)session;
```

Description: This function is called when a response code to "Get Session" is received.

Parameter: session: S0(0x00), S1(0x01), S2(0x02), S3(0x03), Dev. mode(0xF0)

5.1.21 tagMemoryReceived

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

Description: Get tag memory.

Parameter: data: memory information of tag

5.1.22 killedReceived

```
- (void)killedReceived:(uint8_t)statusCode;
```

Description: This function is called when a response code to "Kill Type C Tag" is received.

Parameter: statusCode: success: 0x00/failure: others

5.1.23 lockedReceived

```
- (void)lockedReceived:(uint8_t)statusCode;
```

Description: This function is called when a response code to "Lock Type C Tag" is received.

Parameter: statusCode: success: 0x00/failure: others

5.1.24 responseReboot

```
- (void)responseReboot:(uint8_t)status;
```

Description:This function is called when rebooting (firmware update).

Parameter:status:success: 0x00/failure: others

5.1.25 updatedRegistry

- (void)updatedRegistry:(uint8_t)statusCode;

Description:This function is called when a response code to "Update Registry" is received.

Parameter:statusCode:success: 0x00/failure: others

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:(uint8_t)channel
channelOffset:(uint8_t)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

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

Description:Send the "Get Tx Power Level" command to the reader to set the 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*)dataToWrite;
```

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:(uint8_t)session;

Description:Send the "Set Session" command to the reader to set the current session.

Parameter:session: S0:0x00/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:(uint8_t)mode
Counter:(uint8_t)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: 0x00/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.

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.

@protocol AsReaderDeviceProtocol <NSObject>

8.1 AsReaderDeviceProtocol

8.1.1 responsePowerOnOff

- (void)responsePowerOnOff:(BOOL)isOn
HWMModeChange:(BOOL)isHWMModeChange;

Description:This function is called when the reader sends a response code to "setReaderPower".

Parameter:

isOn:On(YES),Off(NO)

isHWMModeChange: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:(uint8_t)status;

Description:Notification from the module about "Power Reset". This function is called when the reader's connection status changes.

Parameter:status:connected:0xFF/disconnected:0x00

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 errReceived

```
-(void)errReceived:(NSData *)errCode;
```

Description:Response to an invalid command.

Parameter:errCode: 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.

10.1 barcodeDataReceived

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

Description:This function is called when barcode data is received.

Parameter: data: barcode data

10.2 receiveFactoryReset

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

Description:This function is called when a Barcode type reader sends a response code to "Barcode Factory Reset".

Parameter:status: reset start:0x00/reset complete:0xFF