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,ASR-M24D

Asterisk, Inc.

Revision History

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
5	1.7	Add information to Precaution	2022/06/07
6	1.8	Modify the protocol of 0230D/0240D	2023/1/31
7	1.9	1. Add function: setModulationBLF 2. Add Framework: AsReaderBLESDK.framework, modify the screenshot of SDK usage. 3. Delete deligate: allDataReceived. 4. Delete property: readerType. 5. Modify the return value descriptions of the function: setReaderPower and the function setReaderPower. 6. Modify property: currentReaderMode. 7. Add property: connectionType. 8. Add enumeration types: ConnectionType, SaveType. 9. Add delegates: didSetModulation, receivedSleepTime. 10. Add functions: startBleScan, disConnectBLE, connectBLE, getSleepTimeForBLEDevice, setSleepTimeForBLEDevice.	2023/4/10
8	1.10	1. Modify the description of the function releasedTriggerButton/pushedTriggerButton. 2. The following functions were added: SetSleepBeep, setCodeID setSleepTime, setOCRType setAndroidHIDEnable, setiOSHIDEnable, setPresentationMode, getCodeID, getSl EepTime, getOCR getHID, getSymbologies setSsiParamWithData, setSsiParamWithDictionary, getPresentationMode, getAutoLaunch, s EtAutoLaunch getSecurity, receivedCodeID receivedOCR, receivedHID, receivedPresentationMode, receivedSleepBeep, receivedBa RcodeSetSsiSuccess	2024/1/22

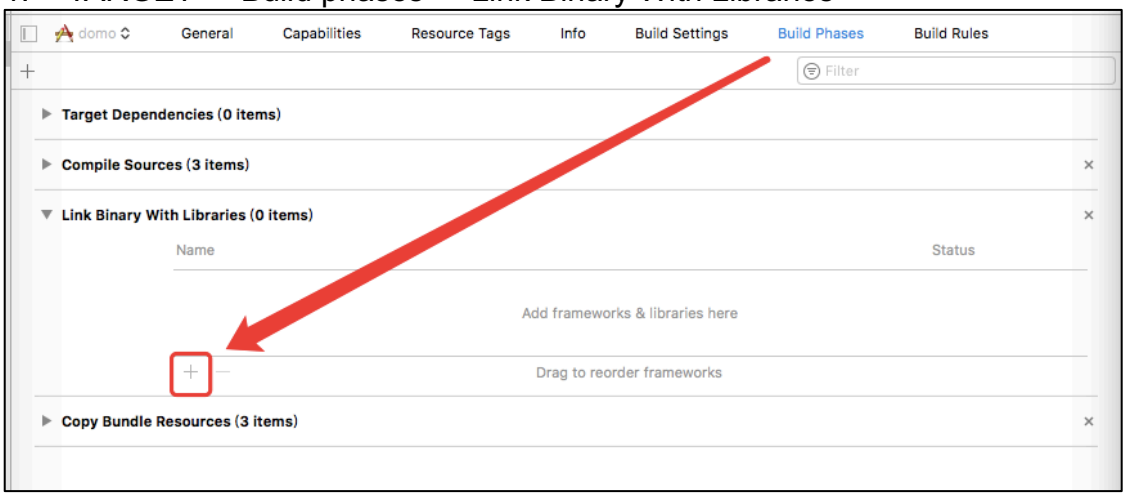
		receivedSymbologies, receivedGetAutoLaunch receivedBarcodeSecurity, receivedSleepTime	
9	1.11	1. Added chapter 12 ErrorCode List 2. Modified the description in section 8.1.9 errorReceived	2024/1/22

Contents

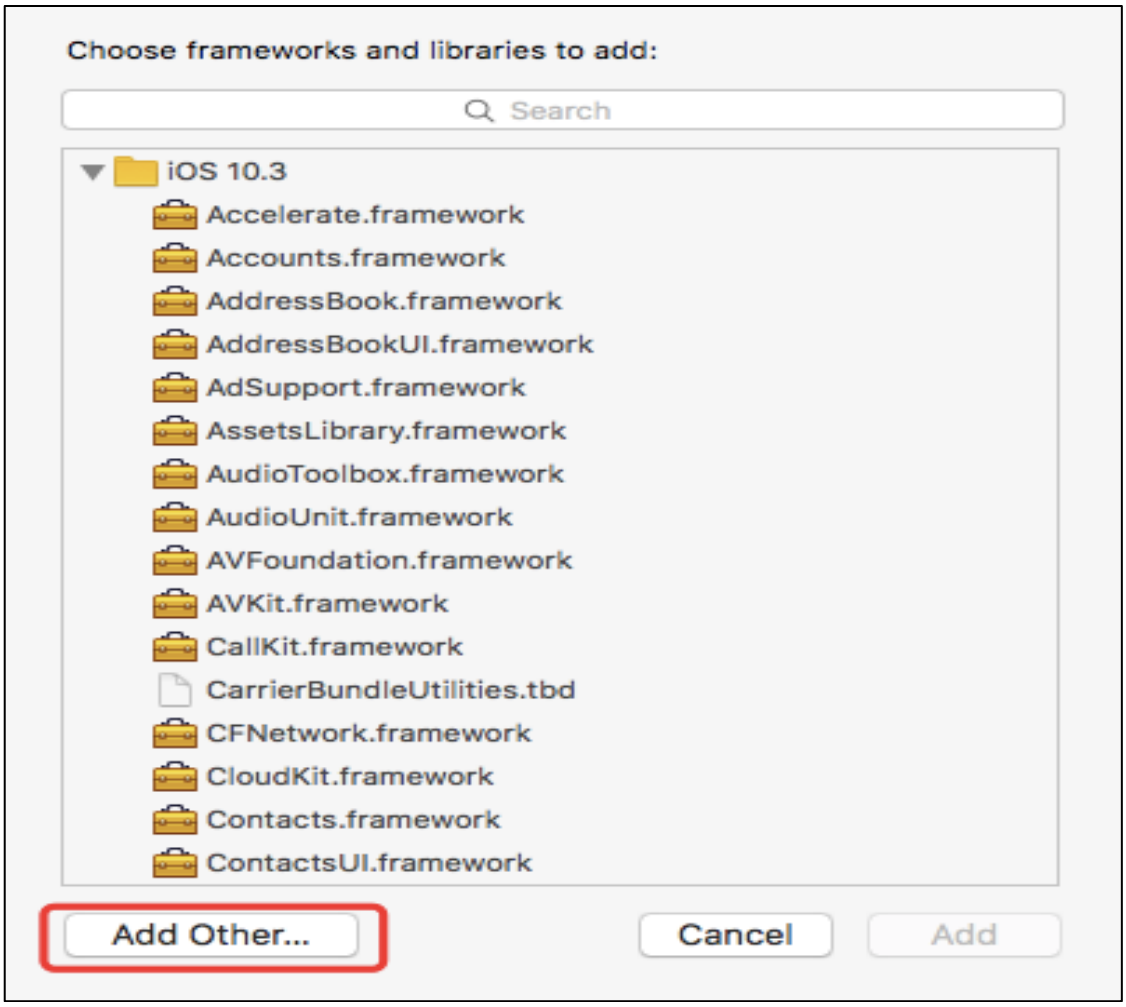
1.SDK Usage

1.1 Add SDK

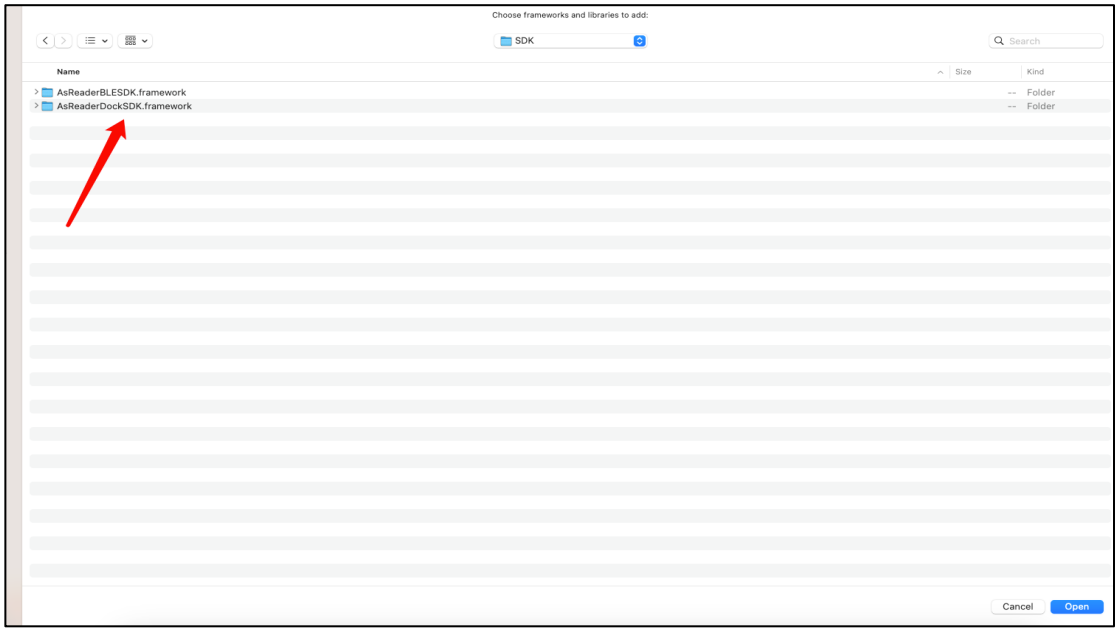
1. TARGET -> Build phases -> Link Binary With Libraries



2. Select “Add Other…”



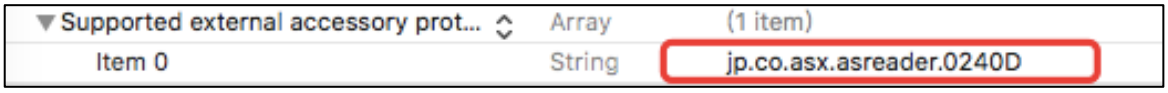
3. Add AsReaderDockSDK.framework



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.0230D, 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.
In the case of sequential command sending, please send the next command after receiving a reply from the previous command.
Otherwise, AsReader may not work properly.

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

```
+ (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-022D, 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: 0/ RFID Version: 1/ manufacturer: 2 / frequency: 3/ tag type: 4
Return value:
YES: success
NO: failed

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: isBeepOn: whether to beep after setting. 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: failed

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: int type. Returns 99 if nDeviceType is unknown.

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: int type. Returns 99 if nDeviceType is unknown.

2.7 setTagCount

```
- (void) setTagCount: (int)mtnu setScanTime: (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

2.8 startBleScan

```
- (BOOL)startBleScan;
```

Description: Start searching for Bluetooth devices.
Return Value:
YES: success
NO: failed

2.9 disConnectBLE

```
- (void)disConnectBLE;
```

Description: Disconnect the Bluetooth device.

2.10 connectBLE

```
- (void)connectBLE: (CBPeripheral *)peripheral;
```

Description: To connect to the Bluetooth device.
Parameter: peripheral: A peripheral object (AsReader device) connected to iOS via Bluetooth.

2.11 getSleepTimeForBLEDevice

```
- (int)getSleepTimeForBLEDevice;
```

Description: Gets the sleep time of the Bluetooth device.

Return Value: the sleep time of the Bluetooth device.

2.12 setSleepTimeForBLEDevice

- (int)setSleepTimeForBLEDevice: (int)min type: (SaveType)type;

Description: Sets the sleep time of the Bluetooth device.
Parameter: min: the sleep time of the Bluetooth device.
type: Enum type. SaveType. See 11.5 SaveType.
Return Value: 1: success.
0: failure.

3. AsReaderBarcodeDevice Class

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

3.1 startScan

- (BOOL)startScan;

Description: The reader starts scanning barcodes.
Return value:

YES: success
NO: failed

3.2 stopScan

- (BOOL)stopScan;

Description: The reader stops scanning barcodes.
Return value:

YES: success
NO: failed

3.3 doFactoryReset

- (BOOL)doFactoryReset;

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

YES: success
NO: failed

3.4 setSymbologyPrefix

-(BOOL)setSymbologyPrefix;

Note: this method only supports the Barcode mode of ASR-022D, 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

3.5 setSleepBeep(for M24D)

- (BOOL)setSleepBeep: (BOOL)isOn;

Description: Sets Whether to enable the sleep function.

Once this function is executed, the function receivedSleepBeep (see 10.8) will be called back to receive the setting result.
Parameter: isOn: Enabled (YES)/ Disabled (NO)
Return Value: YES: success
NO: failed

3.6 setCodeID(for M24D)

- (BOOL)setCodeID: (CODEId)type isBeepOn: (BOOL)isBeepOn;

Description: Sets the CodeID type.
Parameter: type: type of CodeID, isBeepOn: whether to beep after setting. On (YES)/ Off (NO)
Return Value: YES: success
NO: failed

3.7 setSleepTime(for M24D)

- (BOOL)setSleepTime: (int)time;

Description: Sets the sleep time.
Once this function is executed, the function receivedSleepBeep (see 10.8) will be called back to receive the sleep time.
Parameter: time: sleep duration.
Return Value: YES: success
NO: failed

3.8 setOCRType(for M24D)

- (BOOL)setOCRType: (OCRType)type isBeepOn: (BOOL)isBeepOn;

Description: Sets the type of OCR and whether to enable it.
Parameter: type: type of OCR, isBeepOn: whether to beep after setting. On (YES)/ Off (NO)
Return Value: YES: success
NO: failed

3.9 setAndroidHIDEnable(for M24D)

- (BOOL)setAndroidHIDEnable: (BOOL)isOn;

Description: Sets whether to enable HID mode when the device works with an Android device.
Parameter: isOn: Enabled (YES)/ Disabled (NO)
Return Value: YES: success
NO: failed

3.10 setiOSHIDEnable(for M24D)

- (BOOL)setiOSHIDEnable: (BOOL)isOn;

Description: Sets whether to enable HID mode when the device works with an iOS device.
Parameter: isOn: Enabled (YES)/ Disabled (NO)
Return Value: YES: success
NO: failed

3.11 setPresentationMode(for M24D)

- (BOOL)setPresentationMode: (BOOL)isOn isBeepOn: (BOOL)isBeepOn;

Description: Sets whether to enable presentation mode.
Parameter:
isOn: Indicates that the presentation mode is enabled or not. Enabled (YES)/ Disabled (NO)
isBeepOn: whether to beep after setting. On (YES)/ Off (NO)
Return Value: YES: success
NO: failed

3.12 getCodeID(for M24D)

- (BOOL)getCodeID;

Description: Gets CodeID.
Once this function is executed, the function receivedCodeID (see 10.4) will be called back to receive the CodeID.
Return Value: YES: success
NO: failed

3.13 getSleepTime(for M24D)

- (BOOL)getSleepTime;

Description: Gets the sleep time.
Once this function is executed, the functions receivedSleepTime (see 8.1.11) and receivedSleepTime (see 8.1.12) will be called back to receive the sleep time.
Return Value: YES: success
NO: failed

3.14 getOCR(for M24D)

- (BOOL)getOCR;

Description: Gets the OCR type.
Once this function is executed, the function receivedOCR (see 10.5) will be called back to receive the OCR type.
Return Value: YES: success
NO: failed

3.15 getHID(for M24D)

- (BOOL)getHID;

Description: Gets the status of the HID mode.
Once this function is executed, the function receivedHID (see 10.6) will be called back to receive the status of the HID mode.
Return Value: YES: success
NO: failed

3.16 getSymbologies(for M24D)

- (BOOL)getSymbologies;

Description: Gets whether the code type can be read.
Once this function is executed, the function receivedSymbologies (see 10.10) will be called back to receive whether the code type can be read.
Return Value: YES: success
NO: failed

3.17 setSsiParamWithData(for M24D)

-(BOOL)setSsiParamWithData: (NSData*)symbolData beepOn: (BOOL)beepOn;

Description: Sets the parameter Ssi.
Parameter: symbolData: data, beepOn: whether to beep after setting. On (YES)/ Off (NO)
Return Value: YES: success
NO: failed

3.18 setSsiParamWithDictionary(for M24D)

- (BOOL)setSsiParamWithDictionary: (NSDictionary*)symbologies beepOn: (BOOL)beepOn;

Description: Sets the parameter Ssi.
Once this function is executed, the function receivedBarcodeSetSsiSuccess (see 10.9) will be called back to receive the the execution result of setting the Ssi status.
Parameter: symbologies: data, beepOn: whether to beep after setting. On (YES)/ Off (NO)
Return Value: YES: success
NO: failed

3.19 getPresentationMode(for M24D)

- (BOOL)getPresentationMode;

Description: Gets whether the presentation mode is enabled.
Once this function is executed, the function receivedPresentationMode (see 10.7) will be called back to receive whether the presentation mode is enabled.
Return Value: YES: success
NO: failed

3.20 getAutoLaunch(for M24D)

- (BOOL)getAutoLaunch;

Description: Gets whether the Auto-Launch function is enabled.

Once this function is executed, the function receivedGetAutoLaunch (see 10.11) will be called back to receive whether the Auto-Launch function is enabled.
Return Value: YES: success
NO: failed

3.21 setAutoLaunch(for M24D)

- (BOOL)setAutoLaunch: (BOOL)enable bundleId: (NSString*) bundleId;

Description: Sets whether the app starts automatically.
Parameter: enable: whether the app starts automatically, bundleId: bundleId.
Return Value: YES: success
NO: failed

3.22 getSecurity(for M24D)

- (BOOL)getSecurity;

Description: Gets the barcode security level.
Once this function is executed, the function receivedBarcodeSecurity (see 10.12) will be called back to receive the barcode security level.
Return Value: YES: success
NO: failed

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 will be called to return the execution result of the function startScan once the function startScan will be 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 will be called to return the execution result of the function startReadTagsAndRssiWithTagNum once the function startReadTagsAndRssiWithTagNum will be called back.
Parameter:
pcEpc: pcEPC data
rssi: RSSI data

5.1.3 didSetOutputPowerLevel

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

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

5.1.4 didSetChannelParamReceived

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

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

5.1.5 didSetAntiCollision

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

Description: This function will be called to return the execution result of the function setAnticolision once the function setAnticollision will be called back.
Parameter: status: success: 0/ failure: others

5.1.6 didSetSession

- (void)didSetSession: (int)status;

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

5.1.7 channelReceived

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

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

mode: fixed Q: 0/ dynamic Q: 1
counter: counter value

5.1.9 txPowerLevelReceived

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

Description: This function will be called to return the execution result of the function
getOutputPowerLevel once the function getOutoutPowerLevel will be called back.
Assign the Tx power of RFID to the AsReaderInfo class.

rfidPower: Tx power of current RFID
rfidPowerMax: Maximum Tx power that can be set
rfidPowerMin: Minimum Tx power that can be set

5.1.10 regionReceived

- (void)regionReceived: (int)region;

Description: This function will be called to return the execution result of the function getRegion once the function getRegion will be called back.

Parameter:

Region: region

5.1.11 onOffTimeChanged

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

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

5.1.12 fhLbtReceived

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

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

5.1.14 didSetFhLbt

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

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

5.1.15 didSetOptiFreqHPTable

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

Description: This function will be called to return the execution result of the function setFreqHoppingTable once the function setFreqHoppingTable will be called back.
Parameter: status: start: 0/ finish: 1

5.1.16 didSetFHmodeChanged

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

Description: This function will be called to return the execution result of the function setFrequencyHoppingMode once the function setFrequencyHoppingMode will be called back.

5.1.17 rfidModuleVersionReceived

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

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

5.1.18 rfidOnOffTimeReceived

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

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

5.1.19 writtenReceived

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

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

5.1.20 sessionReceived

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

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

5.1.22 killedReceived

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

Description: This function will be called to return the execution result of the

function killTagWithPassword once the function killTagWithPassword will be called back.
Parameter: statusCode: success: 0/ failure: others

5.1.23 lockedReceived

- (void)lockedReceived: (int)statusCode;

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

5.1.24 responseReboot

- (void)responseReboot: (int)status;

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

5.1.26 pcEpcSensorDataReceived

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

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

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

Sample Code:

```
- (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 Convert NSData to 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 selectParamReceived

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

Description: This function will be called to return the execution result of the function getSelectParam once the function getSelectParam will be 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)

5.1.28 didSetModulation:

- (void)didSetModulation: (int)status;

Description: This function is used to call back the execution result of the function setModulationBLF.
Parameter: status: success: 0/ 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 will be 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: failed

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: failed

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: failed

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: failed

7.5 getFhLbtParameter

- (BOOL)getFhLbtParameter;

Description: To get the parameters of FH and LBT.
Return value:
YES: success
NO: failed

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: failed

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: failed

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: failed

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: failed

7.10 lockTagMemoryWithAccessPassword

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

Description: To lock 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: failed

7.11 getSession

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

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

YES: success
NO: failed

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: 0/ S2: 2/ S3: 3
Return value:

YES: success
NO: failed

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: failed

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: 1
counter: change target at N-th Tx On according to inventory round result(default: 1)

Return value:

YES: success
NO: failed

7.15 updateRegistry

- (BOOL)updateRegistry;

Description: Update registry.
Return value:
YES: success
NO: failed

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: failed

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: failed

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: failed

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: failed

7.20 setOptimumFrequencyHoppingTable

```
- (BOOL)setOptimumFrequencyHoppingTable;
```

Description: Set optimum frequency hopping table.
Return value:

YES: success
NO: failed

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: failed

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: failed

7.23 setSmartHoppingOnOff

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: failed

7.24 getRegion

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

Description: To get the region information.
Return value:

YES: success
NO: failed

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: failed

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: failed

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: failed

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: failed

7.29 getSelectParameter

- (BOOL)getSelectParameter;

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

Return value:

YES: success
NO: failed

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:

divideRatio DR=8(0), DR=64/3 (1)
m: M=1 (0), M=2 (1), M=4 (2), M=8 (3)
M represents the data encoding type.
trext: No pilot tone(0), Use pilot tone(1)
selection: All(0 or 1), ~SL(2), SL(3)
session: S0(0), S1(1), S2(2), S3(3)
target: A(0), B(1)
qValue: Range: 0-15. 2^q is the number of slots per inventory cycle.

Return value:

YES: success
NO: failed

7.31 setModulationBLF

- (BOOL)setModulationBLF: (int)blf rxMod: (int)rxMod dr: (int)dr;

Description: Set the parameters for the RFID module.

Parameter:

blf: BLE_160 (160), BEL_250 (250), BEL_320 (320), BEL_640 (640)
rxMod: FM0 (0), M2 (1), M4 (2), M8 (3)
dr: 8 (0), 64/3 (1)

Return value:

YES: success
NO: failed

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 will be 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 releasedTriggerButton

- (void)releasedTriggerButton;

Description: This function will be called when the trigger button of the reader is released, and when the function setTriggerModeDefault will be called and the parameter value is “No “.

8.1.3 plugged

- (void)plugged: (BOOL)plug;

Description: This function will be 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 will be called when the reader’s connection status changes.
Parameter: status: connected: 255/ disconnected: 0

8.1.5 pushedTriggerButton

- (void)pushedTriggerButton;

Description: This function will be called when the trigger button of the reader is pressed, and when the function setTriggerModeDefault will be called and the parameter value is “No “.

8.1.6 receivedScanData

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

Description: This function will be call back to receive the scanned data.
Parameter:
readData: the read tag data.
nDeviceType: Fixed value -99

8.1.7 batteryReceived

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

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

8.1.8 onAsReaderTriggerKeyEventStatus

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

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

8.1.9 errorReceived

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

Description: Response to an invalid command.
Parameter: errorCode: Command Code, Sub Error Code
(See Chapter 12 for a list of error codes.)

Error Code Structure
First 2 digits: Error Code *Not used
Next 2 digits: Command Code
Last 2 digits: Sub Error Code

Example: 0x123456
Error Code: 12
Command Code: 34
Sub Error Code: 56

Sample code：

```
- (NSString*)errorReceived:(NSData *)errorCode{
    uint8_t *pData = (uint8_t*)[errorCode bytes];
    NSString *strCommandCode = [NSString stringWithFormat:@"0x%x",
pData[1]]; //Command Code
    NSString *strSubErrCode = [NSString stringWithFormat:@"0x%x",
pData[2]];   //Sub Error Code
```

8.1.10 unknownCommandReceived

-(void)unknownCommandReceived: (int)commandCode;

Description: Response to undefined commands.
Parameter: commandCode: data.

8.1.11 receivedSleepTime

- (void)receivedSleepTime: (int)time isAck: (BOOL)isAck;

Description: This function is used to call back the execution result of the function getSleepTimeForBLEDevice or function setSleepTimeForBLEDevice.
Parameter:
time: the sleep time of the Bluetooth device.
isAck:
YES: The result will return YES if the function setSleepTimeForBLEDevice will be called.
NO: The result will return NO if the function getSleepTimeForBLEDevice will be called.

8.1.12 receivedSleepTime

- (void)receivedSleepTime: (int)time;

Description: This function calls back the execution result after the getSleepTimeForBLEDevice or setSleepTimeForBLEDevice function is called.
Parameter: time: the sleep time of the Bluetooth device.

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: failed

9.2 startScan

- (BOOL)startScan;

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

9.3 stopScan

- (BOOL)stopScan;

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

10. AsReaderBarcodeProtocol Class

Supported AsReaders: 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 will be called to return the execution result of the function startScan once the function startScan will be called back.
Parameter: data: barcode data

10.2 receiveFactoryReset

- (void)receiveFactoryReset: (int)status;

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

10.3 receivedBypassPayload

- (void)receivedBypassPayload: (NSData *)rawData ;

Description: Returns command data after performing some operation.
This function will call back the result of the execution results of the functions setCustomPrefix, setDisableCustomPrefix, setCustomSuffix: , setDisableCustomSuffix, setDisableSymbologyPrefix, setBarcodeEngineUserCommand after they are called.
Parameter: rawData: data

10.4 receivedCodeID (for M24D)

- (void)receivedCodeID: (int)codeID;

Description: Returns the type of the CodeID.
This function receives the CodeID from the current callback function when the getCodeID function (see: 3.12) is called.
Parameter: codeID: codeID data

10.5 receivedOCR (for M24D)

- (void) receivedOCR: (BOOL)isNone ocrAon: (BOOL)isOCRAon ocrBOn: (BOOL)isOCRBon;

Description: This function returns whether the OCR type is enabled.
This function receives the OCR status from the current callback function when the getOCR function (see: 3.14) is called.

Parameter: isNone, isOCRAon, isOCRBon.

10.6 receivedHID (for M24D)

- (void)receivedHID: (BOOL)hidOn receivediOSHID: (BOOL)iOShidOn;

Description: This function returns whether the HID is enabled.
This function receives the HID status from the current callback function when the getHID function (see: 3.22) is called.
Parameter: isOn: whether the HID is enabled. On (YES)/ Off (NO)
iOShidOn: iOS or Android. iOS (YES) /Android (NO)

10.7 receivedPresentationMode (for M24D)

- (void)receivedPresentationMode: (BOOL)isOn;

Description: This function returns whether the Presentation mode is enabled.
This function receives the Presentation mode status from the current callback function when the getPresentationMode function (see: 3.19) is called.
Parameter: isOn: whether the Presentation mode is enabled. On (YES)/ Off (NO)

10.8 receivedSleepBeep (for M24D)

- (void)receivedSleepBeep: (NSData *)data;

Description: Returns the sleep time.
This function receives the setting result from the current callback function when the function setSleepTime (see: 3.7) is called.

Parameter: data: data

10.9 receivedBarcodeSetSsiSuccess (for M24D)

- (void)receivedBarcodeSetSsiSuccess: (NSData *)data;

Description: Returns the result of setting the Ssi.
This function receives the result of setting the Ssi from the current callback function when the setSsiParamWithDictionary (see: 3.18) is called.

Parameter: data: data

10.10 receivedSymbologies(for M24D)

- (void)receivedSymbologies: (NSDictionary*)symbols;

Description: Returns the status of barcode type setting item.
This function receives the status of the barcode type setting items when the function getSymbologies (see 3.16) is called.
Parameter: symbols: data.

10.11 receivedGetAutoLaunch (for M24D)

- (void)receivedGetAutoLaunch: (BOOL)isOn bundle: (NSString*)data;

Description: It returns the enabled status of the device's auto launch function.
This callback function receives the enable status of the device's auto launch function when the getAutoLaunch function (see 3.20) is called.
Parameter: isOn: auto launch (YES) /don't auto launch (NO)
data: Bundle ID

10.12 receivedBarcodeSecurity (for M24D)

- (void)receivedBarcodeSecurity: (NSDictionary*)security;

Description: It returns the security level of the barcode.
This callback function receives the security level of the barcode when the getSecurity function (see 3.22) is called.

Parameter: security: the security level of the barcode.

11. AsReaderInfoDefine Class

11.1 ReaderMode

Type of the current AsReader device:

ReaderModeUnknown = -1,
ReaderModeBarcode = 0,
ReaderModeRFID,
ReaderModeNFC,
ReaderModeDual,
ReaderModeRFIDLF

11.2 SupportType

Type of scan data supported by the current device:

SupportTypeNone = -1,
SupportTypeBarcode = 0,
SupportTypeRFID,
SupportTypeNFC,
SupportTypeDual,
SupportTypeRFIDLF

11.3 ReceiveDataType

Type of the received data:

ReceiveDataTypeUnknown = -1,
ReceiveDataTypeBarcode = 0,
ReceiveDataTypeRFID,
ReceiveDataTypeNFC,
ReceiveDataTypeRFIDLF

11.4 ConnectionType

Current connection type:

ConnectionTypeUSB,
ConnectionTypeBLE

11.5 SaveType

Current saving type:

SaveType_Permanet,
SaveType_Temporary

12. ErrorCode List

12.1 Command Code

CommandCode	Message code
0x01	Set Reader Power Control
0x03	Get Reader Information
0x06	Get Region
0x07	Set Region
0x08	Set System Reset
0x0B	Get Type C A/I Select Parameters
0x0C	Set Type C A/I Select Parameters
0x0D	Get Type C A/I Query Related Parameters
0x0E	Set Type C A/I Query Related Parameters
0x11	Get current RF Channel
0x12	Set current RF Channel
0x13	Get FH and LBT Parameters
0x14	Set FH and LBT Parameters
0x15	Get Tx Power Level
0x16	Set Tx Power Level
0x17	RF CW signal control
0x22	Read Type C UII
0x23	Read Type C UII RSSI
0x25	Read Type C UII TID
0x29	Read Type C Tag Data
0x2A	Read Type C Tag Long Data
0x2E	Get Session
0x2F	Set Session
0x30	Get Frequency Hopping Table
0x31	Set Frequency Hopping Table
0x32	Get Modulation
0x33	Set Modulation
0x34	Get Anti-Collision Mode
0x35	Set Anti-Collision Mode
0x36	Start Auto Read2
0x37	Stop Auto Read2
0x38	Start Auto Read RSSI
0x39	Stop Auto Read RSSI
0x46	Write Type C Tag Data
0x47	BlockWrite Type C Tag Data
0x48	BlockErase Type C Tag Data
0x83	BlockPermalock Type C Tag
0x65	Kill/Recom Type C Tag
0x82	Lock Type C Tag
0xAC	Antenna Check
0xC5	Get RSSI
0xC6	Scan RSSI
0xCA	Get DTC Result
0xD2	Update Registry
0xD4	Get Registry Item
0xFF	Command Failure
0xE4	Set Optimum Frequency Hopping Table
0xE5	Get Frequency Hopping Mode

0xE6	Set Frequency Hopping Mode
0xE7	Get Tx Leakage RSSI Level for Smart hopping Mode
0xE8	Set Tx Leakage RSSI Level for Smart hopping Mode
0xEC	Start Read with Fast Leakage Cal.
0xED	Request Fast Leakage Cal.

12.2 Sub Error Code for RFID

Supported AsReaders: ASR-030D, ASR-031D, ASR-0230D, ASR-0231D, ASR-0240D, ASR-L70D.

Sub Error Code	Description
0x01	Not supported
0x02	Insufficient privileges
0x03	Memory overrun
0x04	Memory locked
0x05	Crypto suite error
0x06	Command not encapsulated
0x07	ResponseBuffer overflow
0x08	Security timeout
0x0B	Insufficient power
0x0F	Non-specific error
0x11	Sensor Scheduling configuration
0x12	Tag Busy
0x13	Measurement type not supported
0x80	No tag detected
0x81	Handle acquisition failure
0x82	Access password failure
0x90	CRC error
0x91	Rx Timeout
0xA0	Registry update failure
0xA1	Registry erase failure
0xA2	Registry write failure
0xA3	Registry not exist
0xB0	UART failure
0xB1	SPI failure
0xB2	I2C failure
0xB3	GPIO failure
0xE0	Not supported command
0xE1	Undefined command
0xE2	Invalid parameter
0xE3	Too high parameter
0xE4	Too low parameter
0xE5	Failure automatic read operation
0xE6	Not automatic read mode
0xE7	Failure to get last response
0xE8	Failure to control test
0xE9	Failure to reset Reader
0xEA	Rfidblock control failure
0xEB	Automatic read in operation
0xF0	Undefined other error
0xF1	Failure to verify write operation
0xFC	Abnormal antenna
0xFF	None error

12.3 Sub Error Code for Barcodes

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

Sub Error Code	Description
0xE0	ERR_NOT SUPRT_CMD
0x0E	ERR_INVALID_PARM
0x17	FAIL_NOT_SUPRT_CMD
0x18	FAIL_UNDEF_CMD
0x01	FAIL_RDR_PWR_CTRL
0xE9	FAIL_RDR_ALREADY_PWR_ON
0xFF	FAIL_CRC_ERROR

12.4 Sub Error Code for M24D

Supported AsReaders: ASX-M24D

Sub Error Code	Description
0x17	Not Support Command
0x18	Undefined Command
0x01	Reader Power Control Fail
0x0E	Invalid Parameter
0x0A	Auto Read Fail
0xE9	Reader Already Power On
0xFF	CRC check Error
0x20	Barcode SSI Timeout
0x21	Barcode SSI No Ack
0x80	NV_item Write Fail
0x81	NV_item Write Fail because Battery Low voltage
0x82	NV_item erase Fail
0xE5	Auto Read in Operation Fail