



AsReader DOCK SDK 4

SDK マニュアル V1.10

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

変更履歴

No.	バージョン	変更内容	日付
1	1.0	新規作成	2018/03/22
2	1.1	説明追加 AsReaderBarcodeProtocol Class	2018/05/28
3	1.2	getOutputPowerLevel/setOutputPowerLevel/ txPowerLevelReceived メソッドの説明の修正	2018/06/21
4	1.3	getReaderInfo: 間違った引数の説明を修正	2019/01/04
5	1.4	022D に関する情報を追加 温度、湿度メソッド startReadTagsRFM/pcEpcSensorDataReceived を追加	2020/04/15
6	1.5	フォーマット調整、記載ミスを修正	2020/06/03
7	1.6	下記のメソッドを追加: 5.1.27 selectParamReceived 7.28 setSelectParameter 7.29 getSelectParameter 7.30 setQueryParam	2020/07/10
8	1.7	注意事項の説明を追加	2022/06/07
9	1.8	0230D/0240D のプロトコルの変更	2023/1/31
10	1.9	Framework: AsReaderBLESDK.framework の追加 一部のメソッドを追加 一部の列挙を追加	2023/3/31
11	1.10	1. releasedTriggerButton/pushedTriggerButton メソッドの説明の修正 2. ASR-M24D の対応に伴うメソッドの追加: setSleepBeep, setCodeID, setSleepTime, setOCRType, setAndroidHIDEnable, setiOSHIDEnable, setPresentationMode, getCodeID, getSleepTime, getOCR, getHID, getSymbolologies, setSsiParamWithData, setSsiParamWithDictionary, getPresentationMode, getAutoLaunch, setAutoLaunch, getSecurity, receivedCodeID, receivedOCR, receivedHID, receivedPresentationMode, receivedSleepBeep,	2024/08/15

AsReader

		receivedBarcodeSetSsiSuccess, receivedSymbologies , receivedGetAutoLaunch, receivedBarcodeSecurity, receivedSleepTime	
--	--	--	--

目次

1 SDK の使用	8
1.1 SDK 追加	8
1.2 AsReader protocol 追加	9
1.3 SDK のインポート	10
1.4 注意事項	10
2 AsReaderDevice Class	11
2.1 getSDKVersion	11
2.2 setTriggerModeDefault	11
2.3 getReaderInfo	11
2.4 setBeep	12
2.5 setReaderPower	12
2.6 setReaderPower	13
2.7 setTagCount	13
2.8 startBleScan	13
2.9 disconnectBLE	14
2.10 connectBLE	14
2.11 getSleepTimeForBLEDevice	14
2.12 setSleepTimeForBLEDevice	14
3 AsReaderBarcodeDevice Class	15
3.1 startScan	15
3.2 stopScan	15
3.3 doFactoryReset	15
3.4 setSymbologyPrefix	15
3.5 setSleepBeep	16
3.6 setCodeID	16
3.7 setSleepTime	16
3.8 setOCRType	17
3.9 setAndroidHIDEnable	17
3.10 setiOSHIDEnable	17
3.11 setPresentationMode	17
3.12 getCodeID	18
3.13 getSleepTime	18
3.14 getOCR	18
3.15 getHID	18
3.16 getSymbologies	19
3.17 setSsiParamWithData	19
3.18 setSsiParamWithDictionary	19
3.19 getPresentationMode	20
3.20 getAutoLaunch	20
3.21 setAutoLaunch	20
3.22 getSecurity	21
4 AsReaderInfo Class	22

4.1 Properties	22
5 AsReaderRFIDProtocol Class	25
5.1 AsReaderRFIDDeviceDelegate	25
5.1.1 pcEpcReceived	25
5.1.2 pcEpcRssiReceived	25
5.1.3 didSetOutputPowerLevel.....	25
5.1.4 didSetChannelParamReceived	25
5.1.5 didSetAntiCollision	25
5.1.6 didSetSession	26
5.1.7 channelReceived	26
5.1.8 anticolParamReceived.....	26
5.1.9 txPowerLevelReceived	26
5.1.10 regionReceived	26
5.1.11 onOffTimeChanged	27
5.1.12 fhLbtReceived	27
5.1.13 hoppingTableReceived.....	27
5.1.14 didSetFhLbt.....	27
5.1.15 didSetOptiFreqHPTable	27
5.1.16 didSetFHmodeChanged.....	27
5.1.17 rfidModuleVersionReceived	28
5.1.18 rfidOnOffTimeReceived	28
5.1.19 writtenReceived	28
5.1.20 sessionReceived.....	28
5.1.21 tagMemoryReceived.....	28
5.1.22 killedReceived.....	28
5.1.23 lockedReceived	29
5.1.24 responseReboot	29
5.1.25 updatedRegistry.....	29
5.1.26 pcEpcSensorDataReceived	29
5.1.27 selectParamReceived.....	31
5.1.28 didSetModulation:	31
6 AsReaderNFCProtocol Class	32
6.1 AsReaderNFCDeviceDelegate	32
6.1.1 nfcDataReceived	32
7 AsReaderRFIDDevice Class	33
7.1 stopScan	33
7.2 startReadTagsAndRssiWithTagNum	33
7.3 getChannel	33
7.4 setChannel	33
7.5 getFhLbtParameter	34
7.6 getOutputPowerLevel	34
7.7 setOutputPowerLevel	34
7.8 writeTagMemoryWithAccessPassword	35
7.9 killTagWithPassword	35

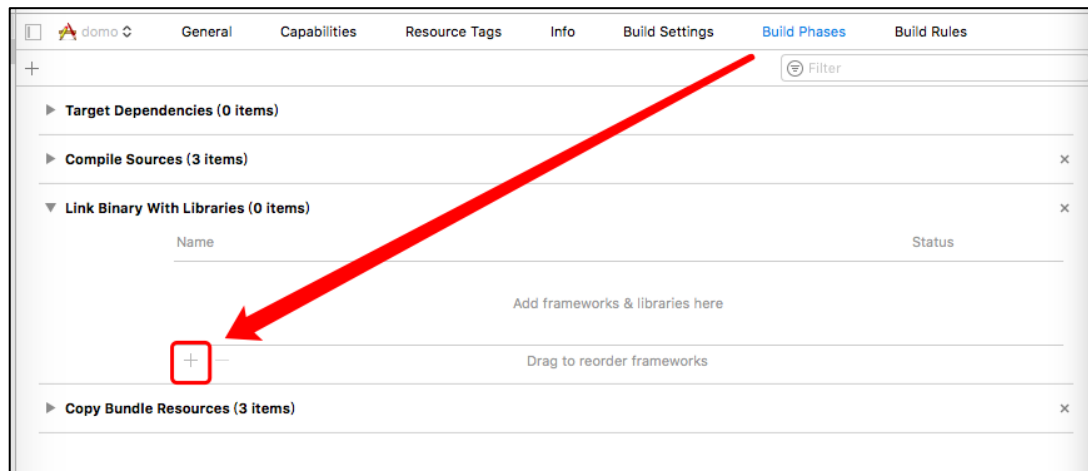
7.10 lockTagMemoryWithAccessPassword	35
7.11 getSession.....	36
7.12 setSession	36
7.13 getAnticollision.....	36
7.14 setAnticollision	36
7.15 updateRegistry.....	37
7.16 getRFIDModuleVersion	37
7.17 setHoppingOnOff.....	37
7.18 writeTagMemory	37
7.19 readTagWithAccessPassword	38
7.20 setOptimumFrequencyHoppingTable	38
7.21 getFrequencyHoppingMode.....	38
7.22 getStopCondition.....	38
7.23 setSmartHoppingOnOff	39
7.24 getRegion	39
7.25 startReadTagsRFM.....	39
7.26 setReadTime	40
7.27 setFhLbtParameter	40
7.28 setSelectParameter	41
7.29 getSelectParameter	41
7.30 setQueryParam	42
7.31 setModulationBLF	42
8 AsReaderDeviceProtocol Class.....	43
8.1 AsReaderDeviceProtocol.....	43
8.1.1 responsePowerOnOff	43
8.1.2 releasedTriggerButton	43
8.1.3 plugged	43
8.1.4 readerConnected	43
8.1.5 pushedTriggerButton	44
8.1.6 receivedScanData	44
8.1.7 allDataReceived.....	44
8.1.8 batteryReceived	44
8.1.9 onAsReaderTriggerKeyEventStatus	44
8.1.10 errorReceived	44
8.1.11 unknownCommandReceived	45
8.1.12 receivedSleepTime	45
8.1.13 receivedSleepTime	45
9 AsReaderNFCDevice Class	46
9.1 sendData	46
9.2 startScan.....	46
9.3 stopScan.....	46
10 AsReaderBarcodeProtocol Class	47
10.1 barcodeDataReceived	47
10.2 receiveFactoryReset	47

10.3 receivedBypassPayload	47
10.4 receivedCodeID.....	47
10.5 receivedOCR	48
10.6 receivedHID	48
10.7 receivedPresentationMode.....	48
10.8 receivedSleepBeep.....	48
10.9 receivedBarcodeSetSsiSuccess.....	49
10.10 receivedSymbologies.....	49
10.11 receivedGetAutoLaunch.....	49
10.12 receivedBarcodeSecurity	49
11 AsReaderInfoDefine Class.....	50
11.1 ReaderMode	50
11.2 SupportType.....	50
11.3 ReceiveDataType	50
11.4 ConnectionType.....	50
11.5 SaveType	50

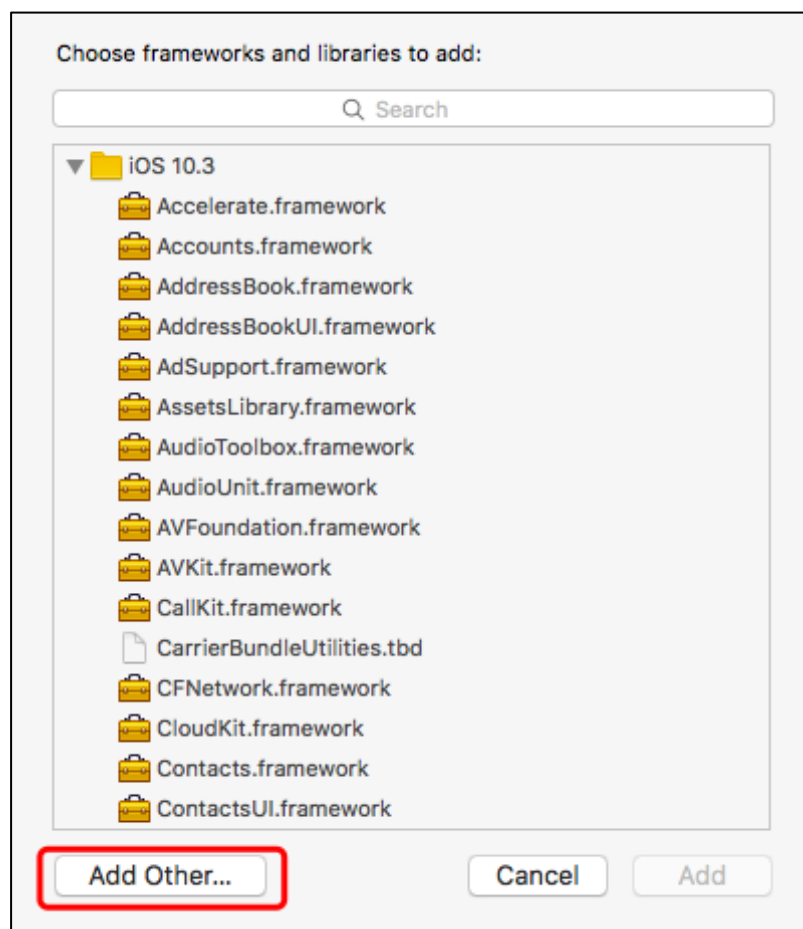
1 SDK の使用

1.1 SDK 追加

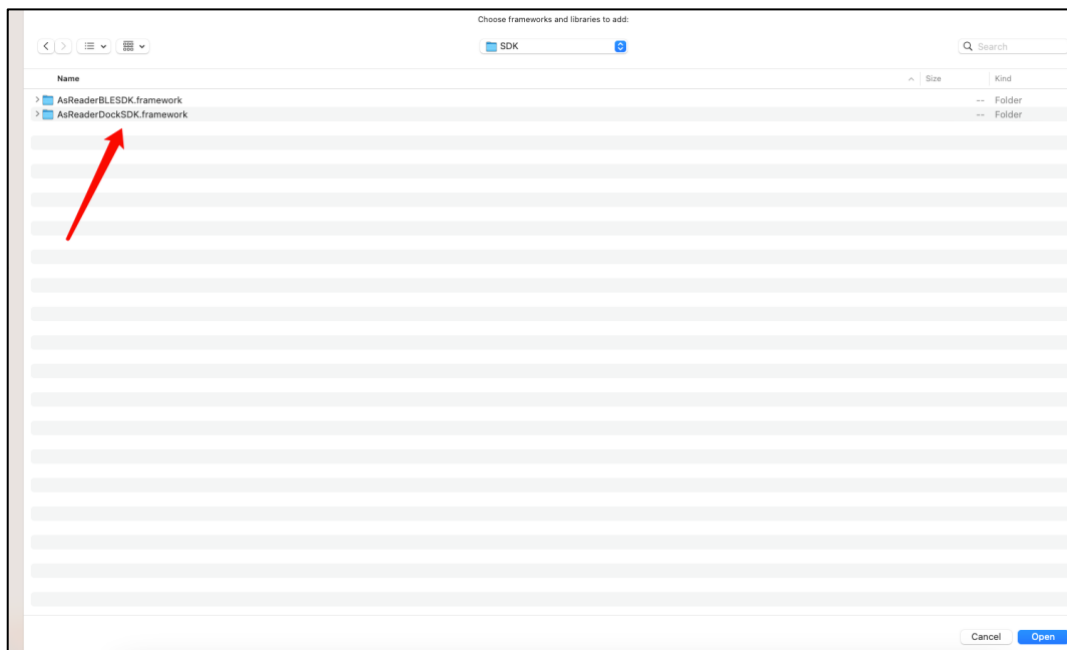
1.1.1 TARGET -> Build phases -> Link Binary With Libraries



1.1.2 「Add Other…」をクリックし、SDK を追加します。



1.1.3 AsReaderDockSDK.framework を追加します。



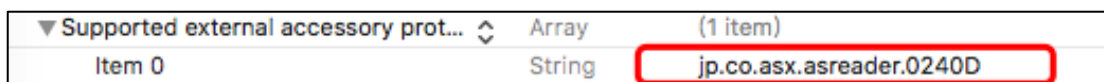
1.1.4 AsReaderDockSDK.framework が追加されていることを確認します。



1.2 AsReader protocol 追加

info.plist に Supported external accessory protocols を追加し、以下を追加します。

- ASX-510R, 520R: jp.co.asx.asreader.barcode
- ASR-010D, 020D, 022D, M24D: jp.co.asx.asreader.6dongle.barcode
- ASX-300R, ASX-301R: jp.co.asx.asreader.rfid
- ASR-030D, ASR-031D: jp.co.asx.asreader.6dongle.rfid
- ASR-0230D, ASR-0240D: jp.co.asx.asreader.0230D
jp.co.asx.asreader.0240D



1.3 SDK のインポート

使用したいプロジェクトの*.m と*.h に、SDK を import します。

```
#import "AsReaderDevice.h"
```

1.4 注意事項

本 SDK を使用する時は、C++言語を使用している為、プロジェクトに.mm ファイルを追加するか、Xcode の設定に libc++を追加する必要があります。

(.mm ファイルの中身は空でよい)

連続してコマンドを送信する場合は前回の送信したコマンドのレスポンスを受け取ってから次のコマンドを送信してください。

前回の送信したコマンドのレスポンスを受け取る前に次のコマンドを送信すると、正しく動作しない場合があります。

2 AsReaderDevice Class

サポートされている 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、ASR-M24D

2.1 getSDKVersion

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

説明: SDK バージョン情報を取得します。

返り値: バージョン (例: 1.0.0)。

2.2 setTriggerModeDefault

注意: このメソッドは ASR-0230D, ASR-0231D, ASR-0240D, ASR-022D のみサポートします。

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

説明: AsReader トリガーモードを設定します。

引数: YES: トリガー操作で読み取り実行する

NO: トリガー操作で読み取りを実行しない (イベント通知のみ)

2.3 getReaderInfo

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

説明: AsReader データ情報を取得します。

引数: infoType: データタイプ モジュール(0) / RFID Version(1)
/ メーカー(2) / 周波数(3) / タグタイプ(4)。

返り値: 成功: YES

失敗: NO

2.4 setBeep

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

説明: AsReader のビープ音、バイブ、イルミネーション、LED を設定します。

引数: beepOn: On (YES) / Off (NO)
vibrationOn: On (YES) / Off (NO)
illuminationon: On (YES) / Off (NO)
led: On (YES) / Off (NO)

返り値: 成功: YES
失敗: NO

2.5 setReaderPower

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

説明: AsReader の電源を On すると同時にビープ音、バイブ、イルミネーション、LED を設定します。

引数: isBeep: On (YES) / Off (NO)
isVib: On (YES) / Off (NO)
isLed: On (YES) / Off (NO)
isIllu : On (YES) / Off (NO)
nDeviceType: デバイスタイプ (int 型): 未知: 99/Barcode:

0/RFID: 1/NFC: 2

返り値: int 型: 99、未知デバイスタイプ

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;
```

説明: AsReader の電源を On すると同時にビープ音、バイブ、イルミネーション、LED を設定します。

引数 : isBeep: On (YES) / Off (NO)

isVib: On (YES) / Off (NO)

isLed: On (YES) / Off (NO)

isIllu: On (YES) / Off (NO)

isConnectedBeep: 接続成功後、ビープ音あり: YES / ビープ音なし: NO

nDeviceType : デバイスタイプ (int 型) : 未知: 99/Barcode: 0/RFID: 1/NFC: 2

返回值: int 型: 99、未知デバイスタイプ

2.7 setTagCount

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

説明: スキャンされたタグをカウントします。

引数: mtnu: 読み取りするタグの最大数

mtime: 読取りの最大時間

repeatCycle: 読取りの繰り返し回数

2.8 startBleScan

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

説明: Bluetooth で AsReader を検索します。

返回值: YES: 成功

NO: 失敗

2.9 disconnectBLE

```
- (void)disconnectBLE;
```

説明: Bluetooth で接続された AsReader との接続を切断します。

2.10 connectBLE

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

説明: Bluetooth で接続可能な AsReader に接続します。

引数: peripheral: Bluetooth を介して接続された AsReader。

2.11 getSleepTimeForBLEDevice

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

説明: Bluetooth で接続された AsReader のスリープ時間を取得します。

返り値: int 型: AsReader のスリープ時間

2.12 setSleepTimeForBLEDevice

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

説明: Bluetooth で接続された AsReader のスリープ時間を設定します

引数: min: AsReader のスリープ時間

type: 列挙型 SaveType (11.5 を参照)

返り値: int 型: 1 成功、0 失敗

3 AsReaderBarcodeDevice Class

サポートされている AsReader: ASX-510R、ASX-520R、ASR-010D、ASR-020D、ASR-0230D、ASR-0231D、ASR-0240D、ASR-022D、ASR-M24D

3.1 startScan

- (BOOL) startScan;

説明: AsReader バーコードスキャンを開始します。

戻り値: 成功: YES

失敗: NO

3.2 stopScan

- (BOOL) stopScan;

説明: AsReader バーコードスキャンを停止します。

戻り値: 成功: YES

失敗: NO

3.3 doFactoryReset

注意: このメソッドは ASR-230D, ASR-0231D, ASR-0240D, ASR-022D の Barcode モードのみサポートします。

- (BOOL) doFactoryReset;

説明: 初期化します (バーコードモジュール)。

戻り値: 成功: YES

失敗: NO

3.4 setSymbologyPrefix

注意: このメソッドは ASR-230D, ASR-0231D, ASR-0240D, ASR-022D の Barcode モードのみサポートします。

-(BOOL) setSymbologyPrefix;

説明: バーコードの接頭辞を表示するかを設定します。(例バーコード「123」、接頭辞を表示する場合「A123」)

戻り値: 成功: YES

該当デバイスはサポートしない or 現在はスキャン状態ではない: NO

3.5 setSleepBeep

注意：このメソッドは ASR-M24D のみサポートします。

- (BOOL)setSleepBeep: (BOOL)isOn;

説明：スリープ機能を使用するかどうかを設定します。

このメソッドを実行後に、コールバックメソッド `receivedSleepBeep` で設定結果を受け取ります。

引数：isOn: On (YES) / Off (NO)

戻り値：成功：YES

失敗：NO

3.6 setCodeID

注意：このメソッドは ASR-M24D のみサポートします。

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

説明：CodeID のタイプを設定します。

引数：type: タイプ、isBeepOn: On (YES) / Off (NO)

戻り値：YES：成功

NO：失敗

3.7 setSleepTime

注意：このメソッドは ASR-M24D のみサポートします。

- (BOOL)setSleepTime: (int)time;

説明：スリープタイムを設定します。

このメソッドを実行後に、コールバックメソッド `receivedSleepBeep` で設定結果を受け取ります。

引数：time: タイム

戻り値：成功：YES

失敗：NO

3.8 setOCRType

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: OCR タイプとステータスを設定します。

引数: type:タイプ、isBeepOn: On (YES)/Off (NO)

戻り値: 成功: YES

失敗: NO

3.9 setAndroidHIDEnable

注意: このメソッドは ASR-M24D のみサポートします。

- (BOOL)setAndroidHIDEnable: (BOOL)isOn;

説明: Android HID モードをオンにするかどうかを設定します。

引数: isOn: On (YES)/Off (NO)

戻り値: 成功: YES

失敗: NO

3.10 setiOSHIDEnable

注意: このメソッドは ASR-M24D のみサポートします。

- (BOOL)setiOSHIDEnable: (BOOL)isOn;

説明: iOS HID モードを使用するかどうかを設定します。

引数: isOn: On (YES)/Off (NO)

戻り値: 成功: YES

失敗: NO

3.11 setPresentationMode

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: プレゼンテーションモードを On に設定します。プレゼンテーションモードに設定すると、読み取り範囲内でバーコードを検出すると、読み取りを実行します。

引数: isOn: プレゼンテーションモード On、isBeepOn: ssiBeep

戻り値: 成功: YES

失敗: NO

3.12 getCodeID

注意：このメソッドは ASR-M24D のみサポートします。

- (BOOL)getCodeID;

説明：CodeID を取得します。このメソッドを実行後に、コールバックメソッド receivedCodeID で CodeID を受け取ります。

戻り値：成功：YES
失敗：NO

3.13 getSleepTime

注意：このメソッドは ASR-M24D のみサポートします。

- (BOOL)getSleepTime;

説明：スリープタイムを取得します。このメソッドを実行後に、コールバックメソッド 8.1.13receivedSleepTime でスリープタイムを受け取ります。

戻り値：成功：YES
失敗：NO

3.14 getOCR

注意：このメソッドは ASR-M24D のみサポートします。

- (BOOL)getOCR;

説明：OCR のステータス (OCR-A, OCR-B) を取得します。このメソッドを実行後に、コールバックメソッド receivedOCR で OCR のステータスを受け取ります。

戻り値：成功：YES
失敗：NO

3.15getHID

注意：このメソッドは ASR-M24D のみサポートします。

- (BOOL)getHID;

説明：HID モードのステータスを取得します。このメソッドを実行後に、コールバックメソッド receivedHID で HID モードのステータスを受け取ります。

戻り値：成功：YES
失敗：NO

3.16 getSymbologies

注意: このメソッドは ASR-M24D のみサポートします。

```
- (BOOL) getSymbologies;
```

説明: 読み取り可能なバーコードシンボルを取得します。このメソッドを実行後に、コールバックメソッド `receivedSymbologies` でバーコードシンボルのステータスを受け取ります。

戻り値: 成功: YES

失敗: NO

3.17 setSsiParamWithData

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: SSI パラメータを設定します。

引数: `symbolData`: データ、`beepOn:On` にするかどうか

戻り値: 成功: YES

失敗: NO

3.18 setSsiParamWithDictionary

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: SSI パラメータを設定します。このメソッドを実行後に、コールバックメソッド `receivedBarcodeSetSsiSuccess` で Ssi パラメータのステータスを受け取ります。

引数: `symbologies`: データ、`beepOn:On` にするかどうか

戻り値: 成功: YES

失敗: NO

3.19 getPresentationMode

注意: このメソッドは ASR-M24D のみサポートします。

- (BOOL)getPresentationMode;

説明: プレゼンテーションモードのステータスを取得します。このメソッドを実行後に、コールバックメソッド `receivedPresentationMode` でプレゼンテーションモードのステータスを受け取ります。

戻り値: 成功: YES

失敗: NO

3.20 getAutoLaunch

注意: このメソッドは ASR-M24D のみサポートします。

- (BOOL)getAutoLaunch;

説明: アプリの自動起動のステータスを取得します。このメソッドを実行後に、コールバックメソッド `receivedGetAutoLaunch` でアプリの自動起動のステータスを受け取ります。

戻り値: 成功: YES

失敗: NO

3.21 setAutoLaunch

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: アプリの自動起動を設定します。

引数: `enable`: 自動起動のステータス、`bundleId`: `bundleId`

戻り値: 成功: YES

失敗: NO

3.22 getSecurity

注意：このメソッドは ASR-M24D のみサポートします。

- (BOOL)getSecurity;

説明：バーコードのセキュリティレベルを取得します。このメソッドを実行後に、コールバックメソッド receivedBarcodeSecurity (10.12 を参照) でバーコードのセキュリティレベルを受け取ります。

戻り値：成功：YES

失敗：NO

4 AsReaderInfo Class

4.1 Properties

```
@property(nonatomic, readonly) NSString *deviceName;//デバイス名称
```

```
@property(nonatomic, readonly) NSString *deviceHardware;//デバイスハードウェア
```

```
@property(nonatomic, readonly) NSString *deviceID;//デバイスID
```

```
@property(nonatomic, readonly) NSString *deviceManufacturer;//デバイスメーカー
```

```
@property(nonatomic, readonly) NSString *deviceModelNumber;//デバイスモデルNo.
```

```
@property(nonatomic, readonly) NSString *deviceSerialNumber;//デバイスシリアルNo.
```

```
@property(nonatomic, readonly) NSString *deviceProtocol;//デバイスプロトコル
```

```
@property(readonly, assign) int readerType;//リーダータイプ
```

```
@property(readonly, assign) int currentSelectDevice;// 選択されたデバイス
```

```
@property(readonly, assign) BOOL isPowerOn;// モジュールパワーON
```

```
@property(readonly, assign) BOOL canUseRFID; //RFIDアクティブ
```

```
@property(readonly, assign) BOOL canUseBarcode; //Barcodeアクティブ
```

```
@property(readonly, assign) BOOL canUseNFC; //NFCアクティブ
```

```
@property(readonly, assign) BOOL isBeep; //ビーブ音アクティブ
```

```
@property(readonly, assign) BOOL isVibration; //バイブレーションアクティブ
```

```
@property(readonly, assign) BOOL isLED; //LEDアクティブ
```

```
@property(readonly, assign) BOOL isIllumination;//イルミネーションアクティブ
```

```
@property(readonly, assign) BOOL isSymbologyPrefix;// タグ先頭文字
```

```
@property(readonly, assign) BOOL isTriggerModeDefault;//デフォルトトリガモード
```

```
@property(readonly, assign) float rfidpower;//RFID アウトプットパワー
```

```
@property(readonly, assign) float rfidPowerMax;//RFID最大アウトプットパワー
```

```
@property(readonly, assign) float rfidPowerMin;//RFID最小アウトプットパワー
```

```
@property(readonly, assign) int rfidOnTime;// RFID オンタイム
```

```
@property(readonly, assign) int rfidOffTime;//RFID オフタイム
```

```
@property(readonly, assign) int nRFIDchannel;// RFID チャンネル
```

```
@property(readonly, assign) int count;// タグカウント
```

```
@property(readonly、 assign) int scanTime;//スキャンタイム
```

```
@property(readonly、 assign) int cycle;// スキャン週期
```

```
@property(readonly、 assign) int carrierSenseTime;// キャリア検知時間
```

```
.....
```

```
@property(readonly、 assign) int targetRFPowerLevel;//RF パワーレベル
```

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

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

```
@property(readonly、 assign) int rfidContinuousWave;//RFID 連続スキャン
```

```
@property(readonly、 assign) BOOL isSmartHopping;// スマートホッピング
```

```
@property(nonatomic、 readonly) NSString *rfidModuleVersion;//RFID モジュールバージョン
```

```
@property(readonly、 assign) BOOL isShowPrintNSLog;//log 出力
```


5 AsReaderRFIDProtocol Class

サポートされている AsReader: ASX-300R、ASX-301R、ASR-030D、ASR-031D、ASR-0230D、ASR-0231D

5.1 AsReaderRFIDDeviceDelegate

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

5.1.1 pcEpcReceived

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

説明: 「startScan」の実行結果をコールバックします。

引数: pcEpc: pcEPC データ

5.1.2 pcEpcRssiReceived

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

説明: 「startReadTagsAndRssiWithTagNum」の実行結果をコールバックします。

引数: pcEpc: pcEPC データ

rssi:RSSI データ

5.1.3 didSetOutputPowerLevel

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

説明: 「setOutputPowerLevel」の実行結果をコールバックします。

引数: status: 成功(0) / 失敗(0以外)

5.1.4 didSetChannelParamReceived

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

説明: 「setChannel」の実行結果をコールバックします。

引数: statusCode: 成功(0) / 失敗(0以外)

5.1.5 didSetAntiCollision

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

説明: 「setAnticollision」の実行結果をコールバックします。

引数: status: 成功(0) / 失敗(0以外)

5.1.6 didSetSession

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

説明: 「setSession」の実行結果をコールバックします。
引数: status: 成功(0) / 失敗(0以外)

5.1.7 channelReceived

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

説明: 「getChannel」の実行結果をコールバックします。
引数: channel: RFID チャンネル
channelOffset: RFID チャンネルオフセット

5.1.8 anticollParamReceived

```
- (void)anticollParamReceived:(int)mode  
Counter:(int)counter;
```

説明: 「getAnticollision」の実行結果をコールバックします。
引数: modo: 固定 Q (0) / ダイナミック Q (1)

5.1.9 txPowerLevelReceived

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

説明: 「getOutputPowerLevel」の実行結果をコールバックします。
コールバック後 RFID TX Power 値を commonReadInfo オブジェクトに保存します。
fRFIDpower: 現在のアウトプットパワー
fRFIDpowerMax: 設定できる最大のアウトプットパワー
fRFIDpowerMin: 設定できる最小のアウトプットパワー

5.1.10 regionReceived

```
- (void)regionReceived:(int)region;
```

説明: 「getRegion」の実行結果をコールバックします。
引数: region: リージョン

5.1.11 onOffTimeChanged

- (void)onOffTimeChanged;

説明: 「setReadTime」の実行結果をコールバックします。

5.1.12 fhLbtReceived

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

説明: 「getFhLbtParameter」の実行結果をコールバックします。

引数: fhLb: 読取時間(16bit)、アイドル時間(16bit)、キャリア監視時間(16bit)、ターゲット RF パワーレベル (16bit)、FH(8bit)、LBT(8bit)、CW(8bit)

5.1.13 hoppingTableReceived

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

説明: 「getFrequencyHoppingTable」の実行結果をコールバックします。

引数: table: テーブルサイズ (8bit)。

5.1.14 didSetFhLbt

- (void)didSetFhLbt:(int)status;

説明: 「setHoppingOnOff」の実行結果をコールバックします。

引数: status: 成功(0) / 失敗 (0 以外)

5.1.15 didSetOptiFreqHPTable

- (void)didSetOptiFreqHPTable:(int)status;

説明: 「setOptimumFrequencyHoppingTable」の実行結果をコールバックします。

引数: status: 成功(0) / 失敗(1)

5.1.16 didSetFHmodeChanged

- (void)didSetFHmodeChanged;

説明: 「setFrequencyHoppingMode」後コールバックします。

5.1.17 rfidModuleVersionReceived

- (void)rfidModuleVersionReceived;

説明: 「getRFIDModuleVersion」後コールバックします。
RFID モジュールバージョンを AsReaderInfo クラスに保存

5.1.18 rfidOnOffTimeReceived

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

説明: 「getFhLbtParameter」の実行結果をコールバックします。
引数: data: RFID on/off データを CommonReaderInfo クラスに保存

5.1.19 writtenReceived

- (void)writtenReceived:(int) statusCode;

説明: 「writeTagMemoryWithEPC」の実行結果をコールバックします。
引数: statusCode: 成功(0) / 失敗(0 以外)

5.1.20 sessionReceived

- (void)sessionReceived:(int) session;

説明: 「getSession」の実行結果をコールバックします。
引数: session: S0(0) / S1(1) / S2(2) / S3(3)

5.1.21 tagMemoryReceived

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

説明: 「readTagWithAccessPassword」の実行結果をコールバックします。
引数: data: タグの保存内容

5.1.22 killedReceived

- (void)killedReceived:(int) statusCode;

説明: 「killTagWithPassword」の実行結果をコールバックします。
引数: statusCode: 成功(0) / 失敗(0 以外)

5.1.23 lockedReceived

- (void)lockedReceived:(int)statusCode;

説明: 「lockTagMemoryWithAccessPassword」の実行結果をコールバックします。

引数: statusCode: 成功(0) / 失敗(0以外)

5.1.24 responseReboot

- (void)responseReboot:(int)status;

説明: デバイス再起動の実行結果をコールバックします (FW 更新時)。

引数: status: 状態コード 成功(0) / 失敗(0以外)

5.1.25 updatedRegistry

- (void)updatedRegistry:(int)statusCode;

説明: 「updateRegistry」の実行結果をコールバックします。

引数: status: 状態コード 成功(0) / 失敗(0以外)

5.1.26 pcEpcSensorDataReceived

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

説明: 呼び出しメソッド: startReadTagsRFM の実行結果をコールバックします。

引数:

pcEpc: 温度タグ/湿度タグデータ

sensorData: 温度/湿度データ

サンプルコード :

```
- (void)pcEpcSensorDataReceived:(NSData *)pcEpc sensorData:(NSData *)sensorData
{
    int codeType;//タグタイプ : 2 (湿度タグ) / 3 (温度タグ)
    int onChipRssiCodeValue;//タグシフトRSSIデータ
    int sensorCodeValue;//温度/湿度データ (16進数)
    double calcTemp;//温度 (摂氏)
    NSMutableString *tmpTagid;//タグpcepcデータ (16進数)
    NSData *tagid = pcEpc;
    NSData *taghex = sensorData;
    //pcepc NSDataデータをNSStringに変換
    tmpTagid = [[NSMutableString alloc] init];
    unsigned char* ptrtagid= (unsigned char*) [tagid bytes];
    for(int i = 0; i < tagid.length; i++)
        [tmpTagid appendFormat:@"%02X", *ptrtagid++ & 0xFF ];

    //温度、湿度データを解析
    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;
```

説明: select 機能を取得する時、コールバックします。

getSelectParameter メソッドを呼び出すと、実行結果をコールバックします。

引数: ターゲット (3桁)、動作 (3桁)、ストレージ (2桁)、インジケータ (32桁)、長さ (8桁)、切断 (1桁)、保留 (7桁)、マスク (0~255桁)。

5.1.28 didSetModulation:

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

説明: メソッド「setModulationBLF」を呼び出した後に実行結果をコールバックします。

引数: status: 成功 (0x00)

6 AsReaderNFCProtocol Class

サポートされている AsReader: ASR-0240D

6.1 AsReaderNFCDeviceDelegate

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

6.1.1 nfcDataReceived

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

説明: NFC タグデータを取得時にコールバックします。

引数: data: NFC タグデータ

7.5 getFhLbtParameter

- (BOOL) getFhLbtParameter;

説明: FH と LBT の引数を取得します。

戻り値: 成功: YES

失敗: NO

7.6 getOutputPowerLevel

- (BOOL) getOutputPowerLevel;

説明: 現在の Tx パワーレベル及び最大と最小 Tx パワーレベルを取得します。(取得された Tx パワーレベルは txPowerLevelReceived デリゲートを通して CommonReaderInfo クラスに値をセットします)

戻り値: 成功: YES

失敗: NO

7.7 setOutputPowerLevel

- (BOOL) setOutputPowerLevel:(int)powerLevel;

説明: Tx パワーレベルを設定します。

引数: power: Tx パワーレベル(日本版の Tx パワーレベル範囲: 18~24dBm、日本版以外の Tx パワーレベル範囲: 18~25dBm)。

戻り値: 成功: YES

失敗: NO

7.8 writeTagMemoryWithAccessPassword

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

説明: タグデータを書き込みます。

引数: accessPassword: アクセスパスワード 00000000

epc: タグ

memoryBank: RFU(0) / EPC(1) / TID(2) / User(3)

startAddress: スタートアドレス

dataToWrite: 書込みデータ

返回值: 成功: YES

失敗: NO

7.9 killTagWithPassword

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

説明: タグをキルします。

注意: タグをキルする時、必ずパスワードを設定してください。

引数: password: パスワード 00000000 に設定された場合、タグキル無効

epc: タグの EPC

返回值: 成功: YES

失敗: NO

7.10 lockTagMemoryWithAccessPassword

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

説明: タグをロックします。

注意: タグをロックする時、必ずパスワードを設定してください。

引数: accessPassword: アクセスパスワード 0x00000000 に設定された場合、タグロック無効。

epc: タグの EPC

lockData: ロックデータ

返回值: 成功: YES

失敗: NO

7.11 getSession

- (BOOL)getSession;

説明: session を取得します。

戻り値: 成功: YES

失敗: NO

7.12 setSession

- (BOOL)setSession:(int)session:

説明: session を設定します。

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

戻り値: 成功: YES

失敗: NO

7.13 getAnticollision

- (BOOL)getAnticollision;

説明: アンチコリジョンモードを取得します。

戻り値: 成功: YES

失敗: NO

7.14 setAnticollision

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

説明: アンチコリジョンモードを設定します。

引数: mode: 固定 Q(0) / ダイナミック Q(1)

counter: カウンター (初期値 1)

戻り値: 成功: YES

失敗: NO

7.15 updateRegistry

– (BOOL)updateRegistry;

説明: レジストリを更新します。

戻り値: 成功: YES

失敗: NO

7.16 getRFIDModuleVersion

– (BOOL)getRFIDModuleVersion;

説明: RFID モジュールバージョンを取得します。

戻り値: 成功: YES

失敗: NO

7.17 setHoppingOnOff

– (BOOL)setHoppingOnOff:(BOOL)isOn;

説明: FH と LBT を設定します。

引数: isOn: YES: ホッピング引数を 2、LBT 引数を 1 に設定

NO: ホッピング引数を 1、LBT 引数を 2 に設定

注: On/OffTime、Hopping を順番に設定する必要な場合、

setFhLbtParameter メソッドをお勧めします。

戻り値: YES: メソッドの実行に成功

NO: メソッドの実行に失敗

7.18 writeTagMemory

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

説明: タグデータを書き込みます。

引数: epc: タグ EPC。

dataToWrite: 書込みデータ

戻り値: 成功: YES

失敗: NO

7.19 readTagWithAccessPassword

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

説明: 指定されるメモリの Type C タグデータを読取ります。

引数: accessPassword: アクセスパスワード

epc: タグ

memoryBank: RFU (0) / EPC (1) / TID (2) / User (3)

startAddress: スタートアドレス

dataLength: データの長さ

戻り値: 成功: YES

失敗: NO

7.20 setOptimumFrequencyHoppingTable

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

説明: 最適な FH チャンネルテーブルを自動設定します。

戻り値: 成功: YES

失敗: NO

7.21 getFrequencyHoppingMode

```
- (BOOL)getFrequencyHoppingMode;
```

説明: FH モードを取得します。

戻り値: 成功: YES

失敗: NO

7.22 getStopCondition

```
- (BOOL)getStopCondition;
```

説明: StopCondition の値を取得します。

戻り値: 成功: YES

失敗: NO

7.23 setSmartHoppingOnOff

- (BOOL)setSmartHoppingOnOff:(BOOL)isOn;

説明: スマート FH モードを設定します。

引数: isOn: スマート FM モード スマートモード有効(YES) / スマート FH モード無効(NO)

返回值: 成功: YES

失敗: NO

7.24 getRegion

- (BOOL)getRegion;

説明: リージョン情報を取得します。

7.25 startReadTagsRFM

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

説明: RFID 温度タグ /湿度タグ の読み取りを開始します。

引数: codeType: タグのタイプ

温度タグ: 3、湿度タグ: 2

mtnu: 読み取りするタグ の最大数

mtime: s を単位として読取りの最大時間

repeatCycle: 読取りの繰り返し回数

返回值: YES: メソッドの実行に成功

NO: メソッドの実行に失敗

7.26 setReadTime

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

説明: 読取タイムとアイドルタイムを設定します。

引数: ReadTime: 読取タイム (ms)

IdleTime: アイドルタイム (ms)

注: On/OffTime、Hopping を順番に設定する必要な場合、setFhLbtParameter メソッドをお勧めします。

返り値: YES: メソッドの実行に成功

NO: メソッドの実行に失敗

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;
```

説明: FH と LBT の引数を設定します。

引数: readTime: 読取り時間 (ms)

idleTime: アイドル時間 (ms)

carrierSenseTime: キャリア検知時間 (ms)。固定値: 50

targetRFPowerLevel: ターゲット RF パワーレベル。固定値: -740

frequencyHopping: 起動: 1 及び以上/禁止: 0

listenBeforeTalk: 起動: 1 及び以上/禁止: 0

continuousWave: 固定値: 0

注: Hopping をオンにする場合、引数 frequencyHopping を 2 に設定、引数 listenBeforeTalk を 1 に設定する必要; Hopping をオフにする場合、引数 frequencyHopping を 1 に設定、引数 listenBeforeTalk を 2 に設定する必要があります。

返り値: YES: メソッドの実行に成功

NO: メソッドの実行に失敗

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;
```

説明: フィルター機能を設定します。

引数: target :session: S0(000b)、S1(001b)、S2(010b)、S3(011b)、SL(100b)

action : 参照標準: IS018000-6C

memoryBank : エリア: RFU (00b)、EPC (01b)、TID (10b)、User (11b)

pointer : フィルターのスタートアドレス

length : フィルターデータの長さ

truncate : タグを切断、切断しない (0)

mask : フィルターするデータ

返回值: 成功: YES

失敗: NO

7.29 getSelectParameter

```
- (BOOL)getSelectParameter;
```

説明: select 機能のコンフィグ引数を取得します。

返回值: 成功: YES

失敗: NO

デリゲート:

No.	メソッド	説明	引数	引数値
5.1.27	selectParamReceived	コンフィグ引数を返す	selParam	データの構成: 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;
```

説明: クエリ引数を設定します。

引数: 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 : 0-15、インベントリサイクルのスロット数です。

返り値: 成功: YES

失敗: NO

7.31 setModulationBLF

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

説明: RFID モジュールの変調方式などを設定します。

引数: blf : BLF_160 (160)、BLF_250 (250)、BLF_320 (320)、BLF_640 (640)

rxMod : FM0 (0)、M2 (1)、M4 (2)、M8 (3)

dr : 8 (0)、64/3 (1)

返り値: 成功: YES

失敗: NO

8 AsReaderDeviceProtocol Class

サポートされている 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、ASR-M24D

8.1 AsReaderDeviceProtocol

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

8.1.1 responsePowerOnOff

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

説明: 「setReaderPower」の実行結果をコールバックします。

引数: isOn: パワーオン (YES) / パワーオフ (NO)。

isHWMModeChange: HW モード、チェンジ (YES) / チェンジしない (NO)。

8.1.2 releasedTriggerButton

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

説明: 「setTriggerModeDefault」が「no」の時、トリガー離すとコールバックします。

8.1.3 plugged

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

説明: AsReader とデバイスの接続状態変化時、変化結果をコールバックします。

引数: plug: 接続: YES/切断: NO

8.1.4 readerConnected

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

説明: 「setReaderPower」の実行結果をコールバックします。

引数: status: 接続 (255) / 切断 (0)。

8.1.5 pushedTriggerButton

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

説明: 「setTriggerModeDefault」が「no」の時、トリガー押すとコールバックします。

8.1.6 receivedScanData

```
-(void)receivedScanData:(NSData *)readData
```

説明: スキャンされたデータを取得時にコールバックします。

引数: data: 取得したデータ

8.1.7 allDataReceived

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

説明: 全種類のタグデータを取得します。

引数: タグデータ

8.1.8 batteryReceived

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

説明: 電池残量を取得します。

引数: battery: 電池残量

8.1.9 onAsReaderTriggerKeyEventStatus

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

説明: トリガーを押下時にコールバックします。

引数: status: ステータス

8.1.10 errorReceived

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

説明: エラー情報を取得します。

引数: errorCode: エラーコード、指令コード、サブエラーコード

8.1.11 unknownCommandReceived

-(void)unknownCommandReceived:(int)commandCode;

説明: 定義されないコマンドをレスポンスします。

引数: commandCode: コマンドデータ

8.1.12 receivedSleepTime

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

説明: メソッド「getSleepTimeForBLEDevice」または「setSleepTimeForBLEDevice」を呼び出した後に実行結果をコールバックします。

引数: time: Bluetooth で接続された AsReader のスリープ時間

isAck: YES: メソッド「setSleepTimeForBLEDevice」を呼び出した後に「YES」を返します。

NO: メソッド「getSleepTimeForBLEDevice」を呼び出した後に「NO」を返します。

8.1.13 receivedSleepTime

-(void)receivedSleepTime:(int)time;

説明: 「getSleepTime」を呼び出した後に実行結果をコールバックします。

引数: time: スリープタイム

9 AsReaderNFCDevice Class

サポートされている 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: 棚卸コマンド

NFC_CMD_STARTSCAN: スタートスキャンコマンド

NFC_CMD_STOPSCAN: ストップスキャンコマンド

9.1 sendData

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

説明: データを送信します。

戻り値: 成功: YES

失敗: NO

9.2 startScan

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

説明: NFC スキャンを開始します。

戻り値: 成功: YES

失敗: NO

9.3 stopScan

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

説明: NFC スキャンを停止します。

戻り値: 成功: YES

失敗: NO

10 AsReaderBarcodeProtocol Class

サポートされている AsReader: ASX-510R、ASX-520R、ASR-010D、ASR-020D、ASR-0230D、ASR-0231D、ASR-0240D、ASR-022D、ASR-M24D

10.1 barcodeDataReceived

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

説明: バーコードデータを取得します。バーコードモードの場合、トリガー押すもしくは「startScan」メソッドを呼び出し後、結果をコールバックします。

引数: data: バーコードデータ

10.2 receiveFactoryReset

注意: このメソッドは ASR-0230D、ASR-0231D、ASR-0240D、ASR-022D のバーコードモードのみサポートします。

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

説明: 「doFactoryReset」の実行結果をコールバックします。

引数: status: ステータス、リセット開始 (0) / リセット成功 (255)。

10.3 receivedBypassPayload

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

説明: 一部の動作を行ってからコマンドデータを返します。メソッド setCustomPrefix、setDisableCustomPrefix、setCustomSuffix:、setDisableCustomSuffix、setDisableSymbologyPrefix、setBarcodeEngineUserCommand: を呼び出し後、実行結果をコールバックします。

引数: rawData: データ

10.4 receivedCodeID

注意: このメソッドは ASR-M24D のみサポートします。

```
- (void)receivedCodeID:(int)codeID;
```

説明: CodeID タイプを返します。メソッド「getCodeID」を実行後、このメソッドを通してコールバックします。

引数: codeID: codeID データ

10.5 receivedOCR

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: OCR タイプのステータスを返します。メソッド「getOCR」を実行後、このメソッドを通してコールバックします。

引数: isNone isOCRAon isOCRBon

10.6 receivedHID

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: HID のステータスを返します。メソッド「getHID」を実行後、このメソッドを通してコールバックします。

引数: isOn: HID のステータス。オン (YES) / オフ (NO)

引数: iOShidOn: iOS や Android。iOS (YES) / Android (NO)

10.7 receivedPresentationMode

注意: このメソッドは ASR-M24D のみサポートします。

```
- (void)receivedPresentationMode: (BOOL)isOn;
```

説明: デモモードのステータスを返します。メソッド「getPresentationMode」を実行後、このメソッドを通してコールバックします。

引数: isOn: デモモードのステータス。オン (YES) / オフ (NO)

10.8 receivedSleepBeep

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: スリープタイムを返します。メソッド「setSleepBeep」または「setSleepTime」を実行後、このメソッドを通してコールバックします。

引数: data: データ

10.9 receivedBarcodeSetSsiSuccess

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: Ssi の設定結果を返します。メソッド「setSsiParamWithDictionary」を実行後、このメソッドを通してコールバックします。

引数: data: データ

10.10 receivedSymbologies

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: バーコードタイプの設定ステータスを返します。メソッド「getSymbologies」を実行後、このメソッドを通してコールバックします。

引数: symbols: データ

10.11 receivedGetAutoLaunch

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: デモアプリの自動起動のステータスを返します。メソッド「getAutoLaunch」を実行後、このメソッドを通してコールバックします。

引数: isOn: 自動起動のステータス。data: bundleId

10.12 receivedBarcodeSecurity

注意: このメソッドは ASR-M24D のみサポートします。

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

説明: バーコードのセキュリティレベルを返します。メソッド「getSecurity」を実行後、このメソッドを通してコールバックします。

引数: security: データ

11 AsReaderInfoDefine Class

11.1 ReaderMode

AsReader のモード

ReaderModeUnknown = -1

ReaderModeBarcode = 0

ReaderModeRFID

ReaderModeNFC

ReaderModeDual

ReaderModeRFIDLF

11.2 SupportType

サポートしているモード

SupportTypeNone = -1

SupportTypeBarcode = 0

SupportTypeRFID

SupportTypeNFC

SupportTypeDual

SupportTypeRFIDLF

11.3 ReceiveDataType

AsReader 読取データのタイプ

ReceiveDataTypeUnknown = -1

ReceiveDataTypeBarcode = 0

ReceiveDataTypeRFID

ReceiveDataTypeNFC

ReceiveDataTypeRFIDLF

11.4 ConnectionType

接続タイプ

ConnectionTypeUSB

ConnectionTypeBLE

11.5 SaveType

保存タイプ

SaveType_Permanet

SaveType_Temporary