



AsReader Combo SDK

SDK マニュアル V1.2

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

変更履歴

序号	バージョン	変更内容	日付
1	1.0	新規作成	2018/03/22
2	1.1	getOutputPowerLevel / setOutputPowerLevel メソッドの説明の修正 txPowerLevelReceived サンプルコードの追加	2018/06/05
3	1.2	メソッドの説明を追加	2018/12/21

目 次

1 SDK を使用するために	5
1.1 使用する SDK のヘッダーファイルのインポート	5
1.2 ExternalAccessory.Framework の追加	7
1.3 libAreteUart.a のインポート	8
1.4 AsReader protocol の追加	9
1.5 SDK のインポート	9
1.6 注意事項	9
2 使い方	10
2.1 ComboBarcodeApi	10
2.1.1 sharedInstance	10
2.1.2 startScan	10
2.1.3 stopScan	10
2.1.4 setFactoryReset	11
2.1.5 setSymbologyPrefix	11
2.2 ComboNFCApi Class	12
2.2.1 sharedInstance	12
2.2.2 sendRawData	12
2.2.3 startScan	12
2.2.4 stopScan	13
2.3 ComboRFIDApi	14
2.3.1 sharedInstance	14
2.3.2 startScan	14
2.3.3 stopScan	15
2.3.4 startReadTagsWithRssi	15
2.3.5 startReadTagsWithTid	16
2.3.6 setSelectParam	16
2.3.7 getSelectParam	17
2.3.8 getChannel	17
2.3.9 setChannel	17
2.3.10 setOnOffTimeSetOnTime	18
2.3.11 setFhLbtParam	18
2.3.12 getFhLbtParam	19
2.3.13 getOutputPowerLevel	19
2.3.14 setOutputPowerLevel	20
2.3.15 writeToTagMemory	20
2.3.16 killTag	21
2.3.17 lockTagMemory	21
2.3.18 getFreqHoppingTable	22
2.3.19 setFreqHoppingTable	22
2.3.20 getSession	22
2.3.21 setSession	23
2.3.22 getAnticollision	23
2.3.23 setAnticollision	23
2.3.24 updateRegistry	24
2.3.25 getRFIDOnOffTime	24
2.3.26 setStopConditionTagNum	25

2.3.27	getRFIDModuleVersion	25
2.3.28	setHoppingOnOff	25
2.3.29	writeToTagMemory	26
2.3.30	getQueryParam	26
2.3.31	setReaderProgMode	26
2.3.32	readFromTagMemory	27
2.3.33	setQueryParam	27
2.3.34	setRfCw	28
2.3.35	getRegistryItem	28
2.3.36	setOptimumFrequencyHoppingTable	28
2.3.37	SetFrequencyHoppingMode	29
2.3.38	getFrequencyHoppingMode	29
2.3.39	getStopCondition	30
2.3.40	setSmarrHoppingOnOff	30
2.3.41	setRSSIThreshold	31
2.3.42	getRSSIThreshold	31
2.4	CommonDevice	32
2.4.1	showPrintNSLog	32
2.4.2	setTriggerModeDefault	32
2.5	SDeviceApi	33
2.5.1	getSDKVersion	33
2.5.2	isOpened	33
2.5.3	isConnected	33
2.5.4	getRegion	33
2.5.5	getReaderInfomation	34
2.5.6	getReaderInfo	34
2.5.7	getCurrentBat	34
2.5.8	setBeep	35
2.5.9	setReaderPower	36
2.5.10	setReaderProgMode	36
2.5.11	setChargingControl	37
2.5.12	setTagCount	37
2.6	CommonReaderInfo	38
2.6.1	sharedInstance	38
2.6.2	プロパティ	38
2.7	HWEventDelegate	40
2.7.1	resPowerOnOff	40
2.7.2	readerConnected	40
2.7.3	checkTriggerStatus	40
2.7.4	plugged	40
2.7.5	pushedTriggerButton	41
2.7.6	releasedTriggerButton	41
2.8	RcpCommonDelegate	42
2.8.1	errReceived	42
2.8.2	adcReceved	42
2.8.3	receivedScanData	42
2.8.4	resFactoryRset	43
2.8.5	nfcRawDataReceived	43

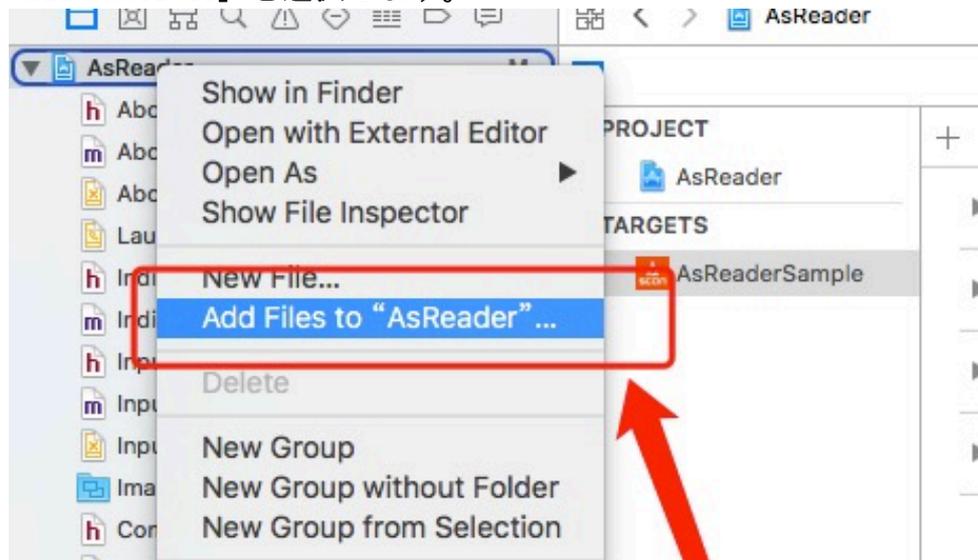
2.8.6	allRawDataReceived	43
2.8.7	startedReadScan	43
2.8.8	stopReadScan	43
2.8.9	barcodeRawDataReceived	44
2.8.10	rfidRawDataReceived	44
2.9	RcpRFIDnDelegate	45
2.9.1	epcReceived	45
2.9.2	rssiReceived	45
2.9.3	didSetOutputPowerLevel	45
2.9.4	didSetChParamReceived	45
2.9.5	didSetAntiCol	45
2.9.6	didSetSession	46
2.9.7	sessionReceived	46
2.9.8	channelReceived	46
2.9.9	anticolParamReceived	46
2.9.10	txPowerLevelReceived	46
2.9.11	regionReceived	47
2.9.12	onOffTimeChanged	47
2.9.13	fhLbtReceived	47
2.9.14	hoppingTableReceived	47
2.9.15	didSetFhLbt	48
2.9.16	didSetOptiFreqHPTable	48
2.9.17	didSetFHmodeChanged	48
2.9.18	resGetFHmode	48
2.9.19	rfidModuleVersionReceived	48
2.9.20	rfidOnOffTimeReceived	49
2.9.21	writedReceived	49
2.9.22	tagMemoryReceived	49
2.9.23	killedReceived	49
2.9.24	lockedReceived	50
2.9.25	updatedRegistry	50
2.9.26	pcEpcReceived	50
2.9.27	pcEpcRssiReceived	50
2.9.28	didSetBeep	50
2.9.29	didSetStopCon	51
2.9.30	stopConditionsReceived	51
2.9.31	selectParamReceived	51
2.9.32	queryParamReceived	51
2.9.33	freqHPTableReceived	51
2.9.34	didSetFreqHPTable	52
2.9.35	ackReceived	52
2.9.36	rssiThresholdReceived	52
2.9.37	readerInfoReceived	53

1 SDK を使用するために

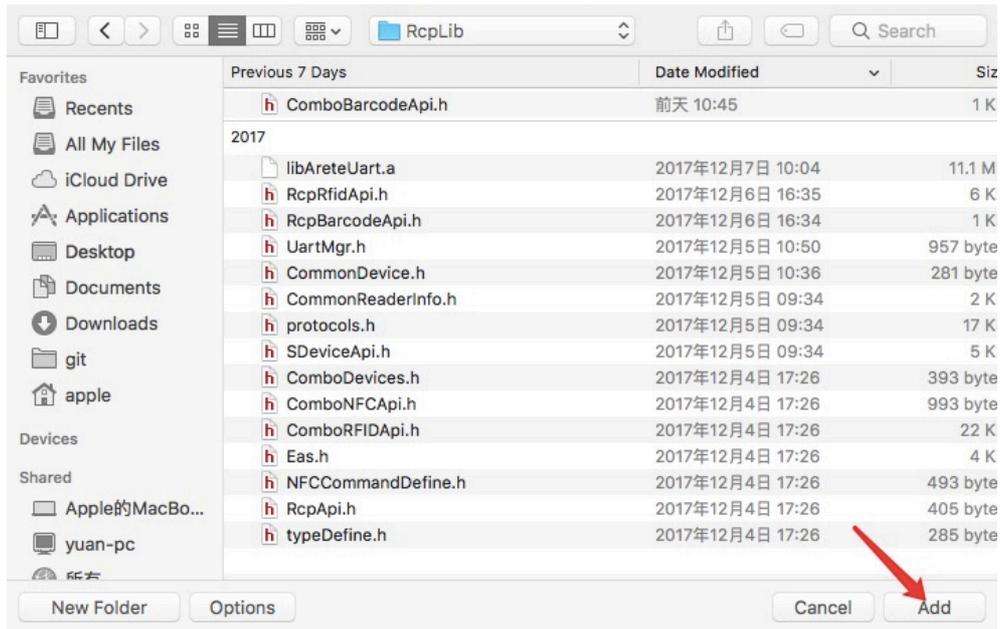
1.1 使用する SDK のヘッダーファイルのインポート

- ・ #import "typeDefine.h"
- ・ #import "protocols.h"
- ・ #import "SDeviceApi.h"
- ・ #import "RcpApi.h"
- ・ #import "ComboNFCApi.h"
- ・ #import "ComboRFIDApi.h"
- ・ #import "ComboBarcodeApi.h"
- ・ #import "CommonDevice.h"
- ・ #import "CommonReaderInfo.h"
- ・ #import "NFCCommandDefine.h"
- ・ #import "ComboDevices.h"

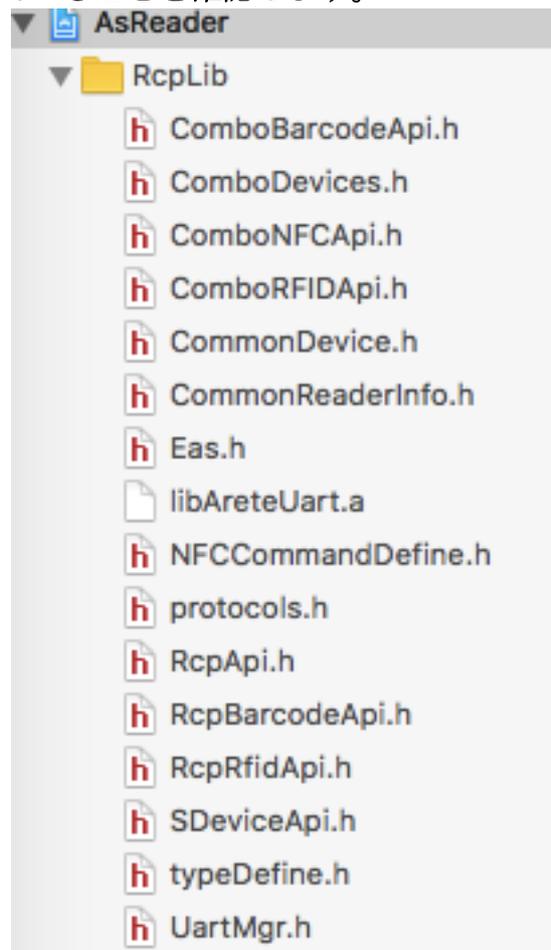
1.1.1 「Add Files to」を選択します。



1. 1. 2 SDK を選択し、「Add」をクリックします。

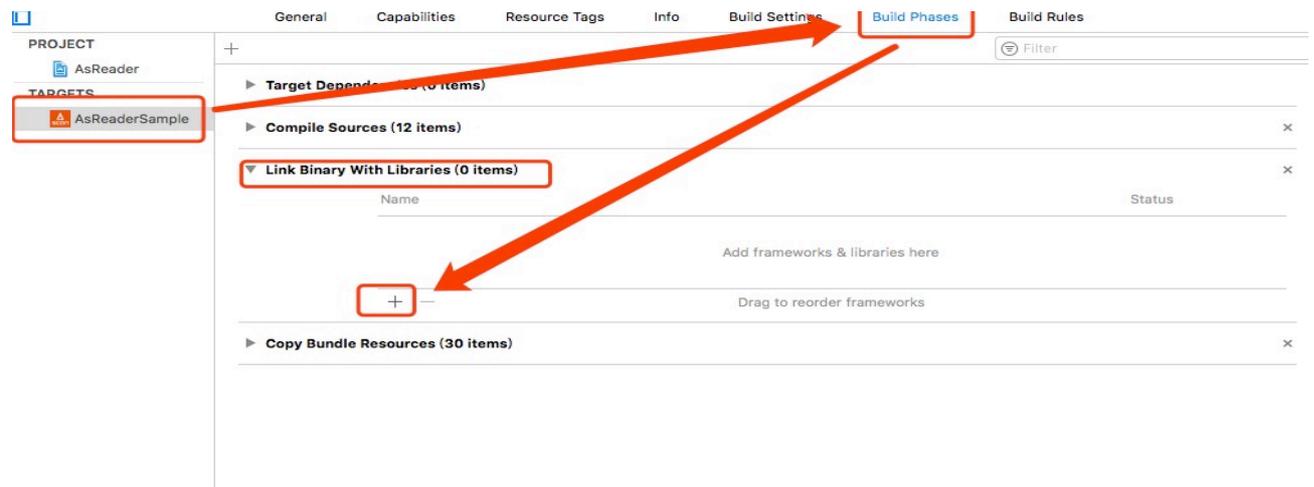


1. 1. 3 SDK が追加されていることを確認します。

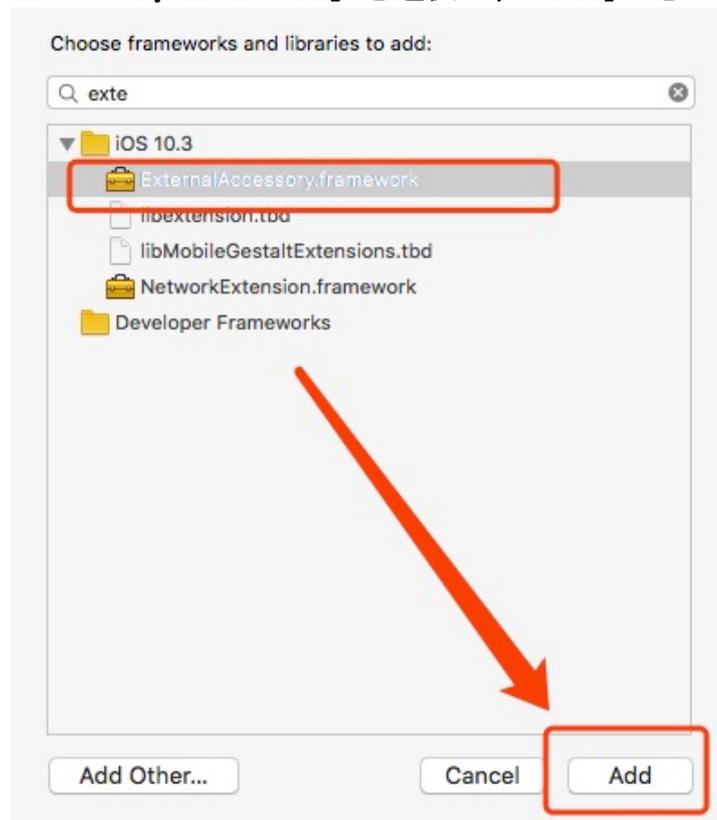


1.2 ExternalAccessory.framework の追加

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



1.2.2 「ExternalAccessory.framework」を選択し、「Add」をクリックします。

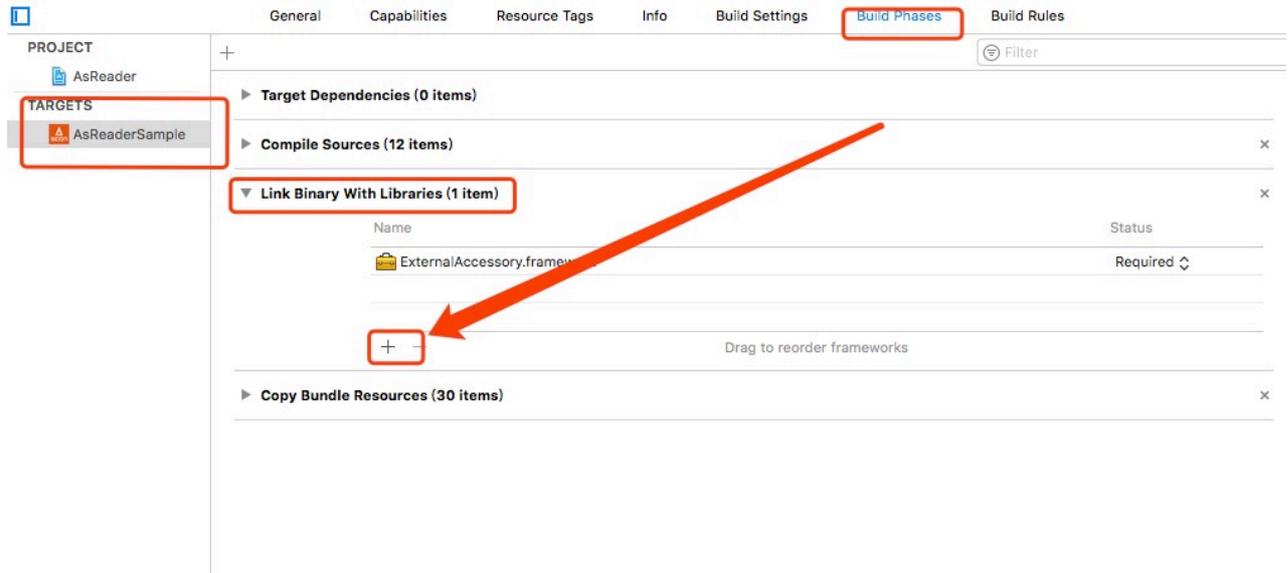


1.2.3 「ExternalAccessory.framework」が追加されていることを確認します。

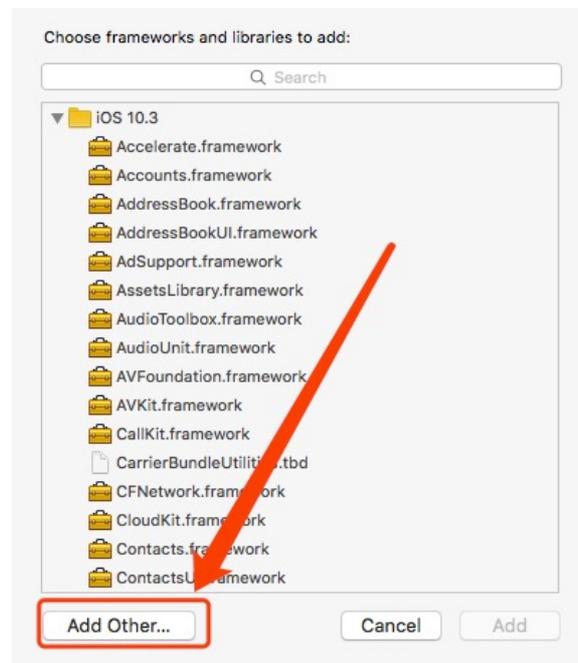
AsReader

1.3 libAreteUart.a のインポート

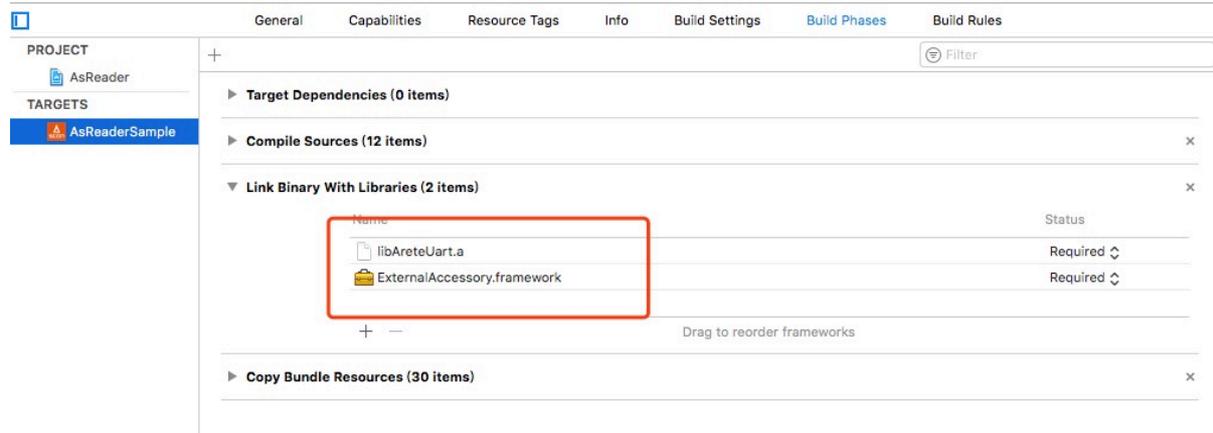
1.3.1 libAreteUart.a を追加します。



1.3.2 「Add Other」 をクリックし、libAreteUart.a を追加します。



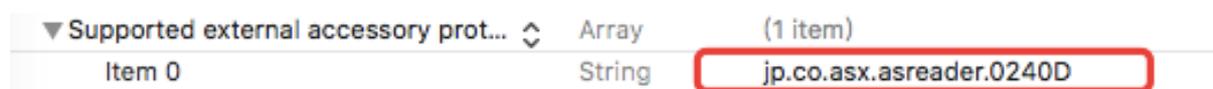
1.3.3 libAreteUart.a が追加されていることを確認します。



1.4 AsReader protocol の追加

info.plist の Supported external accessory protocols に以下を追加します。

- ASX-510R, 520R : jp.co.asx.asreader.barcode
- ASR-010D, 020D : 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.0240D



1.5 SDK のインポート

使用したいプロジェクトの*.m と*.h にヘッダーファイルを import します。
#import "RepBarcodeApi.h"

1.6 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*.m を*.mm に変更します。あるいは Xcode の設定に libc++ を追加します。

(.mm ファイルの中身は空でよい)

2 使い方

2.1 ComboBarcodeApi

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

2.1.1 sharedInstance

```
+ (id)sharedInstance;
```

説明 : ComboBarcodeApi オブジェクトを作成します。(シングルトン)。
戻り値 : ComboBarcodeApi オブジェクト

2.1.2 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

デリゲート :

No.	メソッド	説明	引数	引数値
2.8.3	receivedScanData	スキャンされたデータを返す	readData	スキャンされたデータ
			nDeviceType	0
2.8.7	startedReadScan	「スキャン開始」の実行結果を返す	status	0x00 : 成功
2.8.9	barcodeRawDataReceived	スキャンされたデータを返す	rawData	スキャンされたデータ

2.8.9 デリゲートメソッドを使う場合、バーコード機能のみサポートします。2.8.3 デリゲートメソッドを使う場合、多数の機能 (バーコード、RFID、NFC) をサポートします。

2.1.3 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

デリゲート :

No.	メソッド	説明	引数	引数値
2.8.8	stopReadScan	「スキャン停止」の実行結果を返す	status	0x00 : 成功

2.1.4 setFactoryReset

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

-(BOOL)setFactoryReset;

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

デリゲート：

No.	メソッド	説明	引数	引数値
2.8.4	resFactoryRset	初期設定にリセットの実行結果を返す	status	0x00：リセット開始 0xFF：リセット完成

2.1.5 setSymbologyPrefix

※このメソッドは ASR-0230D、ASR-0231D、ASR-0240D の バーコード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。（接頭辞バーコードがランダムに形成されます。例バーコード「123」、接頭辞を表示する場合「A123」）

戻り値：成功：YES

失敗：NO

2.2 ComboNFCapi Class

サポートされている AsReader : ASR-0240D

2.2.1 sharedInstance

+ (id) sharedInstance;

説明 : ComboNFCapi オブジェクトを作成します。(シングルトン)。

返回值 : ComboNFCapi オブジェクト

2.2.2 sendRawData

**- (BOOL) sendRawData:(Byte*)sendData
len:(int)nLen;**

説明 : AsReader にデータを送信します。

引数 : sendData:データ

nLen : sendData データの長さ (byte)。

※送信するデータは HF モジュールマニュアルを参照ください。

返回值 : 成功 : YES

失敗 : NO

2.2.3 startScan

- (BOOL) startScan;

説明 : NFC スキャンを開始します。

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
2.8.3	receiveScanData	スキャンされたデータを返す	readData	スキャンされたデータ
			nDeviceType	2
2.8.5	nfcRawDataReceived	スキャンされたデータを返す	rawData	スキャンされたデータ

2.8.5 デリゲートメソッドを使う場合 : NFC 機能のみサポートします ; 2.8.3 デリゲートメソッドを使う場合 : 多数の機能 (バーコード、RFID、NFC) をサポートします。

2.2.4 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

デリゲート：

No.	メソッド	説明	引数	引数値
2.8.8	stopReadScan	「スキャン停止」の実行結果を返す	status	0x00：成功

2.3 ComboRFIDApi

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

2.3.1 sharedInstance

```
+ (id) sharedInstance;
```

説明 : ComboRFIDApi オブジェクトを作成します (シングルトン)。

戻り値 : ComboRFIDApi オブジェクト

2.3.2 startScan

```
- (BOOL)startScan:(uint8_t)mtnu mtime:(uint8_t)mtime  
repeatCycle:(uint16_t)repeatCycle;
```

説明 : RFID の読取りを開始します。

引数 : mtneu : 読み取りするタグの最大数、0x00 は制限なし

mtime : 読取りの最大時間、0x00 は制限なし

repeatCycle : 読取りの繰り返し回数、0x00 は制限なし

戻り値 : 成功 : YES

失敗 : NO

デリゲート :

No.	メソッド	説明	引数	引数値
2.8.3	receiveScanData	スキャンされたデータを返す	readData	スキャンされたデータ
			nDeviceType	1
2.8.7	startedReadScan	「スキャン開始」の実行結果を返す	status	0x00:成功
2.8.10	rfidRawDataReceived	スキャンされたデータを返す	rawData	スキャンされたデータ
2.9.26	pcEpcReceived	スキャンされたデータを返す	pcEpc	スキャンされたタグのpcEpc データ

2.9.26 デリゲートメソッドを使う場合、RFID 機能のみサポートします ; 2.8.3 デリゲートメソッドを使う場合、多数の機能 (バーコード、RFID、NFC) をサポートします。

2.3.3 stopScan

- (BOOL)stopScan;

説明：RFID の読取りを停止します。

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.8.8	stopReadScan	「スキャン停止」の実行結果を返す	status	0x00：成功

2.3.4 startReadTagsWithRssi

- (BOOL)startReadTagsWithRssi:(uint8_t)maxTags mtime:(uint8_t)maxTime repeatCycle:(uint16_t)repeatCycle;

説明：RFID の読取りを開始します。同時に RSSI データも読取りします。

引数：maxTags：読み取りするタグの最大数、0x00 は制限なし

mtime：読取りの最大時間、0x00 は制限なし

repeatCycle：読取りの繰り返し回数、0x00 は制限なし

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.27	pcEpcRssiReceived	タグの pcEpc と RSSI データを返す	pcEpc	スキャンされたタグの pcEpc データ
			rssi	Rssi データ

2.3.5 startReadTagsWithTid

**- (BOOL)startReadTagsWithTid:(uint8_t)mtnu mtime:(uint8_t)mtime
repeatCycle:(uint16_t)repeatCycle;**

説明：RFID の読取りを開始します。同時に tid データも読取ります。

引数：mtnu：読み取りするタグの最大数、0x00 は制限なし

mtime：読取りの最大時間、0x00 は制限なし

repeatCycle：読取りの繰り返し回数、0x00 は制限なし

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.1	epcReceived	タグの epc と Tid データを返す	epc	スキャンされたタグの epc データ
			tid	Tid データ

2.3.6 setSelectParam

**- (BOOL)setSelectParam:(uint8_t)target
action:(uint8_t)action
memoryBank:(uint8_t)memoryBank
pointer:(uint32_t)pointer
length:(uint8_t)length
truncate:(uint8_t)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：ISO18000-6C

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

戻り値：成功：YES

失敗：NO

2.3.7 getSelectParam

- (BOOL)getSelectParam;

説明：select 機能のコンフィグ引数を取得します。

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.31	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)

2.3.8 getChannel

- (BOOL)getChannel;

説明：RF チャンネルを取得します。

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.8	channelReceived	コンフィグ引数を返す チャンネル値を CommonReaderInfo.nRFIDchannel にセット		

2.3.9 setChannel

**- (BOOL)setChannel:(uint8_t)channel
channelOffset:(uint8_t)channelOffset;**

説明：AsReader に「Set current RF channel」コマンドを送信します。

引数：Channel (8 bit)、Channel Offset for Miller Subcarrier (8 bit)

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.4	didSetChParamReceived	実行結果を返す	statusCode	0x00:成功

2.3.10 setOnOffTimeSetOnTime

**- (BOOL)setOnOffTimeSetOnTime:(uint16_t)ReadTime
setOffTime:(uint16_t)IdleTime;**

説明：読み取り時間とアイドル時間を設定します。

引数：ReadTime：読取り時間、10 ~ 40000 (ms)

IdleTime：アイドル時間、10 ~ 40000 (ms)

返り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.12	onOffTimeChanged	成功に設定した場合、デリゲートを返す	-	-
2.9.15	didSetFhLbt	実行結果を返す	status	0x00:成功

2.3.11 setFhLbtParam

**- (BOOL)setFhLbtParam:(uint16_t)ReadTime
idleTime:(uint16_t)IdleTime
carrierSenseTime:(uint16_t) carrierSenseTime
rfLevel:(uint16_t)rfLevel
frequencyHopping:(uint8_t)frequencyHopping
listenBeforeTalk:(uint8_t)listenBeforeTalk
continuousWave:(uint8_t)continuousWave;**

説明：FH と LBT の引数を設定します。

引数：readTime (onTime) 読み取り時間 (ms)。

idleTime (offTime) アイドル時間 (ms)。

carrierSenseTime: キャリアセンスタイム。固定値：50。

rfLevel: ターゲットホッピングのパワー。固定値：-740。

frequencyHopping: 起動：0x01 または 0x01 以上/使用禁止：0x00。

listenBeforeTalk: 起動：0x01 または 0x01 以上/使用禁止：0x00。

continuousWave: 固定値：0x00。

注意：Hopping をオンにしたい場合、frequencyHopping を 0x02 に設定し、listenBeforeTalk を 0x01 に設定する必要があります。

Hopping をオフにしたい場合、frequencyHopping を 0x01 に設定し、listenBeforeTalk を 0x02 に設定する必要があります。

返り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.15	didSetFhLbt	実行結果を返す	status	0x00:成功

2.3.12 getFhLbtParam

- (BOOL)getFhLbtParam;

説明：FH と LBT の引数を取得します。

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.13	fhLbtReceived	コンフィグ引数を返す	fhLb	データの構成： ReadTime(16bit) Idle Time(16bit) Carrier Sense Time(16bit) Target RF power level(16bit) FH(8bit) LBT(8bit) CW(8bit)
2.9.20	rfidOnOffTimeReceived	コンフィグ引数を返す CommonReaderInfo クラ スに値をセット	data	無効
2.9.33	freqHPTableReceived	コンフィグ引数を返す	status	0x00：使用禁止 0x01：起動

2.3.13 getOutputPowerLevel

- (BOOL)getOutputPowerLevel;

説明：現在の Tx パワーレベル及び最大と最小の Tx パワーレベルを取得します。

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.10	txPowerLevelReceived	コンフィグ引数を返す RFID の Tx パワーレベル 値を CommonReaderInfo クラスにセット	power	無効

2.3.14 setOutputPowerLevel

- (BOOL)setOutputPowerLevel:(uint16_t)power;

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

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.3	didSetOutputPowerLevel	実行結果を返す	status	0x00：成功

2.3.15 writeToTagMemory

**- (BOOL)writeToTagMemory:(uint32_t)accessPassword
epc:(NSData *)epc
memoryBank:(uint8_t)memoryBank
startAddress:(uint16_t)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を入力

初期値：0x00000000

epc：ターゲットタグ

memoryBank：エリア。RFU (0x00)、EPC (0x01)、TID (0x02)、User (0x03)

startAddress：スタートアドレス、(Word) で区切り

dataToWrite：書き込みデータ

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.21	writtenReceived	実行結果を返す	statusCode	0x00：成功

2.3.16 killTag

```
- (BOOL)killTag:(uint32_t)killpassword  
    epc:(NSData *)epc;
```

説明：タグをキルします。

注意：タグをキルする時、アクセスパスワードを修正しなければなりません。

修正方法：writeToTagMemory (2.3.15) で RFU エリアの最初 2 Word のデータを修正します。

引数：killpassword：タグキルパスワード。killpassword を 0x00000000 に設定された場合、タグキルは無効になります。

epc：タグの EPC。

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
2.9.23	killedReceived	実行結果を返す	statusCode	0x00：成功

2.3.17 lockTagMemory

```
- (BOOL)lockTagMemory:(uint32_t)accessPassword  
    epc:(NSData *)epc  
    lockData:(uint32_t)lockData;
```

説明：タグをロックします。ロックされたエリアが改修できません。

引数：accessPassword：アクセスパスワード

初期値：0x00000000

epc：ターゲットタグの EPC

lockData：ロックするエリア

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
2.9.24	lockedReceived	実行結果を返す	statusCode	0x00：成功

2.3.18 getFreqHoppingTable

- (BOOL)getFreqHoppingTable;

説明：FH テーブルを取得します。

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.14	hoppingTableReceived	コンフィグ引数を返す	table	テーブルサイズ (8bit)

2.3.19 setFreqHoppingTable

**- (BOOL)setFreqHoppingTable:(uint8_t)tableSize
channels:(NSData*)channels;**

説明：FH テーブルを設定します。

引数：tableSize : FH テーブルサイズ (8bit)

channels : チャンネル値

戻り値：成功：YES

失敗：NO

2.3.20 getSession

- (BOOL)getSession;

説明：session を取得します。

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.7	sessionReceived	コンフィグ引数を返す	session	0x00:S0 0x01:S1 0x02:S2 0x03:S3 0xF0:開発モード

2.3.21 setSession

- (BOOL)setSession:(uint8_t)session;

説明：sessionを設定します。

引数：session: S0(0x00)、S1(0x01)、S2(0x02)、S3(0x03)、開発モード (0xF0)

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.6	didSetSession	実行結果を返す	status	0x00:成功

2.3.22 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

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.9	anticolParamReceived	実行結果を返す	mode	0x00：成功
			counter	カウンタ

2.3.23 setAnticollision

**- (BOOL)setAnticollision:(uint8_t)mode
Counter:(uint8_t)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(0x00)、ダイナミック Q(0x01)

counter：カウンタ (初期値 1)

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.5	didSetAntiCol	実行結果を返す	status	0x00：成功

2.3.24 updateRegistry

- (BOOL)updateRegistry;

説明：レジストリを更新します。RFID の設定を永久に保存する時に使います。

返り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.25	updatedRegistry	実行結果を返す	statusCode	0x00:成功

2.3.25 getRFIDOnOffTime

- (BOOL)getRFIDOnOffTime;

説明：読み取り時間とアイドル時間を取得します。

返り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.13	fhLbtReceived	コンフィグ引数を返す	fhLb	データの構成： ReadTime(16bit) Idle Time(16bit) Carrier Sense Time(16bit) Target RF power level(16bit) FH(8bit) LBT(8bit) CW(8bit)
2.9.20	rfidOnOffTimeReceived	コンフィグ引数を返す RFID の RF 値を CommonReaderInfo クラ スにセット	data	無効
2.9.33	freqHPTableReceived	コンフィグ引数を返す	status	0x00：使用禁止 0x01：起動

2.3.26 setStopConditionTagNum

**- (BOOL)setStopConditionTagNum:(Byte)mtnu
Time:(Byte) mtime
Recycle:(UInt16)repeatCycle;**

説明：スキャン停止を設定します。

引数：mtnu：読み取りするタグの最大数、0x00は制限なし

mtime：読取りする最大時間、0x00 は制限なし

repeatCycle：読取りの繰り返し回数、0x00 は制限なし

返り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.26	didSetStopCon	実行結果を返す	status	0x00:成功

2.3.27 getRFIDModuleVersion

- (BOOL)getRFIDModuleVersion;

説明：リーダーのRFIDモジュールバージョンを取得します。

返り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明
2.9.19	rfidModuleVersionReceived	実行結果を返す RFIDのモジュールバージョン値を CommonReaderInfo.strRFIDModuleVersionにセット

2.3.28 setHoppingOnOff

- (BOOL)setHoppingOnOff:(BOOL)isOn;

説明：HPとLBTの引数を設定します。

引数：isOn：YES：HPは2、LBTは1に設定されます。

NO：HPは1、LBTは2に設定されます。

返り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.15	didSetFhLbt	実行結果を返す	status	0x00:成功

2.3.29 writeToTagMemory

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

説明：タグを書き込みます。

引数：epc：タグの EPC。

dataToWrite：Ascii の形でデータを書き込みます。

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.21	writtenReceived	実行結果を返す	statusCode	0x00:成功

2.3.30 getQueryParam

- (BOOL)getQueryParam;

説明：クエリの引数を取得します。

戻り値：成功：YES

失敗：NO

デリゲート

No.	メソッド	説明	引数	引数値
2.9.32	queryParamReceived	コンフィグ引数を返す	qryParam	データの構成： DR(1bit) M(2bit) TRext(1bit) Sel(2bit) Session(2bit) Target(1bit) Q(4bit)

2.3.31 setReaderProgMode

- (BOOL)setReaderProgMode:(uint8_t)mode;

説明：FW の更新モードを設定します。（FW を更新する時のみ使う）

引数：mode：0（MCU）、1（RFIDモード）

戻り値：成功：YES

失敗：NO

2. 3. 32 readFromTagMemory

```
- (BOOL)readFromTagMemory:(uint32_t)accessPassword
                        epc:(NSData*)epc
                memoryBank:(uint8_t)memoryBank
        startAddress:(uint16_t)startAddress
        dataLength:(uint16_t)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

初期値：0x00000000

epc：ターゲットタグの EPC データ

memoryBank：読取りするタグのエリア、RFU (0x00)、EPC (0x01)、TID (0x02)、User (0x03)

startAddress：スタートアドレス。(Word) で区切り

dataLength：データの長さ、(Word) の数

返り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2. 9. 22	tagMemoryReceived	読み取った結果を返す	data	読み取ったデータ

2. 3. 33 setQueryParam

```
- (BOOL)setQueryParam:(uint8_t)dr
                    m:(uint8_t)m
                trext:(uint8_t)trext
                sel:(uint8_t)sel
        session:(uint8_t)session
        target:(uint8_t)target
                q:(uint8_t)q;
```

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、インベントリーサイクルのスロット数です。

返り値：成功：YES

失敗：NO

2.3.34 setRfCw

- (BOOL)setRfCw:(uint8_t)on;

説明：持続波 (CW) 信号のスイッチを設定します。この設定はアイドルモードのみに有効。

引数：on：アイドル時間がタグに持続波を送送します (YES) ; アイドル時間がタグに持続波 (CW) を送送しません (NO) 。

返り値：成功：YES
失敗：NO

2.3.35 getRegistryItem

- (BOOL)getRegistryItem:(uint16_t)registryItem;

説明：レジストリ情報を取得します。

引数：registryItem：バージョン (0x0000)
FW 日付 (0x0001)
band (0x0002)
Tx パワー (0x0003)
FH/LBT (0x0004)
アンチコリジョンモード (0x0005)
調整モード (0x0006)
クエリ (Q) (0x0007)
ホッピングテーブル (0x0008)
Tx パワーテーブル (0x0009)

返り値：成功：YES
失敗：NO

2.3.36 setOptimumFrequencyHoppingTable

- (BOOL) setOptimumFrequencyHoppingTable;

説明：最適な FH テーブルを設定します。

返り値：成功：YES
失敗：NO

デリゲート :ASX-300R の開発を行う時、以下のデリゲートをコールバックします。

No.	メソッド	説明	引数	引数値
2.9.15	didSetFhLbt	実行結果を取得する	status	0x00 : 成功
2.9.12	onOffTimeChanged	成功に設定した場合、デリゲートを返す	-	-
2.9.34	didSetFreqHPTable	実行結果を返す	status	0x00 : 成功

ASX-300R 以外の開発を行う時、以下のデリゲートをコールバックします。

No.	メソッド	説明	引数	引数値
2.9.16	didSetOptiFreqHPTable	実行結果を返す	status	0x00 : 成功

2.3.37 SetFrequencyHoppingMode

- (BOOL) SetFrequencyHoppingMode:(uint8_t)mode;

説明 : FH モードを設定します。

引数 : mode : FH モード。 0: 通常モード ; 1: スマート FH モード

返り値 : 成功 : YES

失敗 : NO

デリゲート :

No.	メソッド	説明
2.9.17	didSetFHmodeChanged	成功に実行した場合、デリゲートを返す

2.3.38 getFrequencyHoppingMode

- (BOOL) getFrequencyHoppingMode;

説明 : FH モードを取得します。

返り値 : 成功 : YES

失敗 : NO

デリゲート : ASX-300R の開発を行う時、以下のデリゲートをコールバックします。

No.	メソッド	説明	引数	引数値
2.9.13	fhLbtReceived	コンフィグ引数を返す	fhLb	データの構成 : ReadTime(16bit) Idle Time(16bit) Carrier Sense Time(16bit) Target RF power level(16bit) FH(8bit) LBT(8bit) CW(8bit)
2.9.20	rfidOnOffTimeReceived	コンフィグ引数を返す。 RFID の RF 値を CommonReaderInfo クラスにセット	data	無効
2.9.33	freqHPTableReceived	コンフィグ引数を返す	status	0x00 : 使用禁止 0x01 : 起動

ASX-300R 以外の開発を行う時、以下のデリゲートをコールバックします。

No.	メソッド	説明	引数	引数値
2.9.18	resGetFHmode	実行結果を返す	引数なし	RFID のスマート FH 値を CommonReaderInfo.bSmartHopping に セット

2.3.39 getStopCondition

- (BOOL) getStopCondition;

説明：setStopCondition の値を取得します。

戻り値：成功：YES

失敗：NO

デリゲート：ASX-300R 以外の開発を行う時、以下のデリゲートを返します。

No.	メソッド	説明	引数	引数値
2.9.30	stopConditionsReceived	コンフィグ引数を返す	data	データの構成： Tag count(1byte) Scan time(1byte) Cycle(1byte)

2.3.40 setSmartHoppingOnOff

- (BOOL)setSmartHoppingOnOff:(BOOL)isOn;

説明：スマート FH モードを設定します。

引数：isOn：YES（スマート FH モード）、NO（通常モード）。（ASX-300R の場合、スマート FH モードに設定できません）

戻り値：成功：YES

失敗：NO

デリゲート：ASX-300R の開発を行う時、以下のデリゲートをコールバックします。

No.	メソッド	説明	引数	引数値
2.9.12	onOffTimeChanged	成功に設定した場合、デリゲートを返す	-	-
2.9.15	didSetFhLbt	実行結果を返す	status	0x00：成功
2.9.34	didSetFreqHPTable	実行結果を返す	status	0x00：成功

ASX-300R 以外の開発を行う時、以下のデリゲートをコールバックします。

No.	メソッド	説明
2.9.17	didSetFHmodeChanged	成功に設定した場合、 デリゲートを返す

2.3.41 setRSSIThreshold

注意：このメソッドは ASR-030Dv2、ASR-031Dv2 のみサポートします。

-(BOOL) setRSSIThreshold:(uint16_t)threshold;

説明：RSSI フィルターを設定します。

引数：threshold：RSSI フィルター値。50 を設定する場合、RSSI<-50dbm のタグが全部フィルターされ、スキャンできなくなります。

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.35	ackReceived	実行結果を返す	commandCode	0x00：成功

2.3.42 getRSSIThreshold

注意：このメソッドは ASR-030Dv2、ASR-031Dv2 のみサポートします。

-(BOOL) getRSSIThreshold;

説明：RSSI フィルター引数を取得します。

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.37	rssThresholdReceived	実行結果を返す	threshold	データを返す

2.4 CommonDevice

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

2.4.1 showPrintNSLog

```
+(void)showPrintNSLog:(BOOL)isShow;
```

説明：ログを出力します。（Xcode のコントロールバーに出力）

引数：isShow：ログ表示：YES

ログ表示しない：NO

2.4.2 setTriggerModeDefault

注意：このメソッドは ASR-0230D、ASR-0231D、ASR-0240D 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：カスタムモード

2.5 SDeviceApi

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

2.5.1 getSDKVersion

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

説明 : SDK バージョンを取得します。
返り値 : SDK バージョン。例 : 2.3.21

2.5.2 isOpened

```
- (BOOL)isOpened;
```

説明 : 通信可能かを取得します。
返り値 : 通信可能 : YES
 通信不可 : NO

2.5.3 isConnected

```
- (BOOL)isConnected;
```

説明 : 接続状態を取得します。
返り値 : 接続成功 : YES
 未接続 : NO

2.5.4 getRegion

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

説明 : RFID モジュールのエリア情報を取得します。
返り値 : 成功 : YES
 失敗 : NO

デリゲート :

No.	メソッド	説明	引数	引数値
2.9.11	regionReceived	コンフィグ引数を返す	region	korea(0x11)、 North america(0x21)、 US(0x22)、 Europe(0x31)、 Japan(0x41)、 China1(0x51)、 China2(0x52)、 Brazil(0x61)

2.5.5 getReaderInfomation

- (BOOL)getReaderInfomation;

説明：RFID モジュールのエリア情報とバージョンを取得します。

戻り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.11	regionReceived	コンフィグ引数を返す	region	korea(0x11)、 North america(0x21) US(0x22) Europe(0x31) Japan(0x41) China1(0x51) China2(0x52) Brazil(0x61)
2.9.19	rfidModuleVersionReceived	コンフィグ引数を返す。 RFID モジュールのバージョン値を CommonReaderInfo.strRFIDModuleVersion にセット		

2.5.6 getReaderInfo

- (BOOL)getReaderInfo:(uint8_t)infoType;

説明：AsReader のデバイス情報を取得します。

引数：infoType：型番 (0x00)、シリアル No. (0x01)、メーカー (0x02)、頻度 (0x03) タグのタイプ (0x04)

戻り値：YES：RFID モード

NO：RFID モードではない

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.19	rfidModuleVersionReceived	コンフィグ引数を返す RFID モジュールバージョン値を CommonReaderInfo. strRFIDModuleVersion にセット		
2.9.38	readerInfoReceived	AsReader 情報を返す	data	無効

2.5.7 getCurrentBat

- (int) getCurrentBat;

説明：AsReader のバッテリー残量を取得します。

戻り値：0 (0%~25%)、1 (26%~50%)、2 (51%~75%)、3 (76%~100%) ;
最低は0です。

2.5.8 setBeep

```
- (BOOL)setBeep:(uint8_t)beepOn  
  setVibration:(uint8_t)vibrationOn  
  setIllumination:(uint8_t)illuminationOn  
  setLED:(uint8_t)led;
```

説明：AsReader のビープ音、バイブレーション、レーザ、LED を設定します。

引数：beepOn：ビープ音。オン (0x01)、オフ (0x00)

vibrationOn：バイブレーション。オン (0x01)、オフ (0x00)

illuminationOn：レーザ。オン (0x01)、オフ (0x00)

led：LED。オン (0x01)、オフ (0x00)

返り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.9.28	didSetBeep	成功に実行した場合、デリゲートを返す	status	成功：0x00

2.5.9 setReaderPower

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

説明：AsReader の電源をオン、ビープ音、バイブレーション、レーザ、LED を設定します。

引数：isOn：電源。オン (YES)、オフ (NO)

isBeep：ビープ音。オン (YES)、オフ (NO)

isVib：バイブレーション。オン (YES)、オフ (NO)

isLed：LED。オン (YES)、オフ (NO)

isIllu：レーザ。オン (YES)、オフ (NO)

nDeviceType：デバイスタイプ。barcode (0)、RFID (1)、NFC (2)

返り値：成功：YES

失敗：NO

デリゲート：

No.	メソッド	説明	引数	引数値
2.7.1	resPowerOnOff	コンフィグ引数を返す	isOn	On (YES)、Off (NO)
			nDeviceType	Unknown(99)、Barcode(0)、RFID(1)、NFC(2)
			bIsHWMModeChange	YES (When mode is changed by H/W switch)
2.7.2	readerConnected	コンフィグ引数を返す	status	Connected (0xFF)、Disconnected(0x00)

2.5.10 setReaderProgMode

```
- (BOOL)setReaderProgMode:(uint8_t)mode;
```

説明：FW の更新モジュールと更新方法を設定します。

引数：mode：APP または外部デバイスで MPU モジュールを更新します (0) ; APP で RFID モジュールを更新します (1) ; 外部デバイスで RFID モジュールを更新します (2) 。

返り値：成功：YES

失敗：NO

2.5.11 setChargingControl

- (BOOL) setChargingControl:(BOOL)isOn;

説明： AsReader を充電する時にスマートデバイスに充電させるかどうかを設定します。

引数： isOn： 充電させる (YES) 、充電させない (NO)

戻り値： 成功： YES

失敗： NO

2.5.12 setTagCount

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

説明： AsReader のスキャン数量、スキャン時間、スキャン回数を設定します。

引数： mtnu： 読み取りタグの最大数、0 の場合制限なし

mtime： 読み取りの最大時間、0 の場合制限なし

repeatCycle： 読み取りの繰り返し回数、0 の場合制限なし

戻り値： 成功： YES

失敗： NO

2.6 CommonReaderInfo

2.6.1 sharedInstance

```
+ (id)sharedInstance;
```

説明： CommonReaderInfo オブジェクトを作成します（シングルトン）。
戻り値： CommonReaderInfo オブジェクト

2.6.2 プロパティ

```
@property(nonatomic, readonly) NSString *strName; //デバイス名称
@property(nonatomic, readonly) NSString *strfirmware; //デバイス FW
@property(nonatomic, readonly) NSString *strhardware; //デバイス HW
@property(nonatomic, readonly) NSString *strID; //デバイス ID
@property(nonatomic, readonly) NSString *strmanufacturer ;//デバイスメーカー
@property(nonatomic, readonly) NSString *strmodelName;//デバイス型番
@property(nonatomic, readonly) NSString *strserialNumber;//デバイスシリアル
@property(nonatomic, readonly) NSString *strProtocol;// プロトコル
@property(readonly, assign) int m_nReaderType;// リーダータイプ 0 : バーコード ; 1 : RFID ; 2 : NFC
@property(readonly, assign) int m_nCurrentSelectDevice; //選択されているデバイス
@property(readonly, assign) BOOL bCanUseRFID;// RFID アクティブ
@property(readonly, assign) BOOL bCanUseBarcode;// Barcode アクティブ
@property(readonly, assign) BOOL bCanUseNFC;// NFC アクティブ
@property(readonly, assign) BOOL bIsPowerOn;// 給電中
@property(readonly, assign) BOOL bBeep;// beep アクティブ
@property(readonly, assign) BOOL bVibration;//バイブアクティブ
@property(readonly, assign) BOOL bLED;//LED アクティブ
@property(readonly, assign) BOOL bIllumination;//イルミネーションアクティブ
@property(readonly, assign) BOOL bSymbologyPrefix;// タグ先頭文字
@property(readonly, assign) BOOL bIsTriggerModeDefault;// デフォルトトリガーモード
@property(readonly, assign) float fRFIDpower;//RFID アウトプットパワー
@property(readonly, assign) float fRFIDpowerMax;//RFID 最大アウトプットパワー
@property(readonly, assign) float fRFIDpowerMin;//RFID 最小アウトプットパワー
@property(readonly, assign) int nRFIDonTime;//RFID On タイム
@property(readonly, assign) int nRFIDoffTime;//RFID Off タイム
@property(readonly, assign) int nRFIDchannel;//RFID チャンネル
@property(readonly, assign) int nCount;//タグカウント
@property(readonly, assign) int nScanTime;//スキャン時間
@property(readonly, assign) int nCycle;//スキャンサイクル
@property(readonly, assign) int nCst;//キャリアセンスタイム
```

AsReader

```
@property(readonly, assign) int    nRfl;//RF パワーレベル
@property(readonly, assign) int    nLbt;//LBT
@property(readonly, assign) int    nFh;//RFID ホッピング
@property(readonly, assign) int    nCw;//RFID 連続スキャン
@property(readonly, assign) BOOL   bSmartHopping;//スマートホッピング
@property(nonatomic, readonly) NSString *strRFIDModuleVersion;RFID//モジュールバージョン
```

2.7 HWEventDelegate

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

2.7.1 resPowerOnOff

```
- (void)resPowerOnOff:(BOOL)isOn  
                  Device:(int)nDeviceType  
          IsHWModeChange:(BOOL)bIsHWModeChange;
```

説明 : AsReader 接続され、電源状態変化時、ステータスをコールバックします。
setReaderPower 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 : isOn : 開く (YES)、 閉じる (NO)

 nDeviceType : 未知 (99)、 Barcode (0)、 RFID (1)、 NFC (2)

 bIsHWModeChange : YES : ハードのスイッチでモードチェンジ時

2.7.2 readerConnected

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

説明 : AsReader 接続状態変化時、ステータスをコールバックします。

setReaderPower を呼び出すと、実行結果をコールバックします。

引数 : status : 接続 (0xFF)、 切断 (0x00)

2.7.3 checkTriggerStatus

```
- (void)checkTriggerStatus:(NSString*)strStatus;
```

説明 : AsReader のトリガーを押下/押下解除の実行結果をコールバックします。

引数 :strStatus :RFID (RFID startScan) ; barcode (Barcode startScan) ; NFC (NFC startScan)

2.7.4 plugged

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

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

引数 : plug : 接続 (YES)、 切断 (NO)

2.7.5 pushedTriggerButton

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

説明：AsReader のトリガーを押すと、変化結果をコールバックします。

2.7.6 releasedTriggerButton

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

説明：AsReader のトリガー押下を解除すると、変化結果をコールバックします。

2.8 RcpCommonDelegate

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

2.8.1 errReceived

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

説明: コマンド実行のエラー情報を取得します。コマンド実行エラーが発生時、結果をコールバックします。

引数: errorCode: コマンドコード

エラーコマンド情報:

0x09	Failure to read the tag memory
0x10	Failure to write data
0x0B	'Read Type C Tag ID Multiple' in Operation
0x0D	Not in mode 'Read Type C Tag ID Multiple'
0x0E	Invalid parameter
0x12	Failure to kill a tag
0x13	Failure to lock a tag
0x15	Failure to read a tag
0x18	Not supported command
0xFF	CRC Error

2.8.2 adcReceived

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

説明: バッテリー残量を返す時、コールバックします。

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

注意: 10s 毎にデータを返します。

引数: data: 電池残量

2.8.3 receivedScanData

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

説明: AsReader のトリガー押下または「startScan」実行時にスキャンされたデータをコールバックします。

引数: readData: スキャンされたデータ (NSData 型)

nDeviceType: 未知(99) ; バーコード(0) ; RFID(1) ; NFC(2)

2.8.4 resFactoryRset

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

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

説明：バーコードモード再起動の時、コールバックします。

Barcode Factory Reset 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：status:リセット起動(0x00)；リセット完成(0xFF)

2.8.5 nfcRawDataReceived

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

説明：NFC データ返す時、コールバックします。

startScan (ComboNFCApi Class) 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：rawData:NFC タグデータ

2.8.6 allRawDataReceived

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

説明：全てタイプのデータを返す時、コールバックします。

startScan 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：rawData:読み取ったデータ (NSData 型)

2.8.7 startedReadScan

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

説明：startScan 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：status：成功 (0x00)

2.8.8 stopReadScan

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

説明：stopScan 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：status：成功 (0x00)。

2.8.9 barcodeRawDataReceived

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

説明：バーコードデータを返す時、コールバックします。
starScan(ComboBarcodeApi)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：rawData：Barcode タグデータ。

2.8.10 rfidRawDataReceived

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

説明：RFID データを返す時、コールバックします。
starScan(ComboRFIDApi)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：rawData:rfid タグデータ。

2.9 RcpRFIDnDelegate

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

2.9.1 epcReceived

```
- (void)epcReceived:(NSData *)epc tid:(NSData *)tid;
```

説明 : 読み取れた epc および tid データをコールバックします。
startReadTagsWithTid 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 : epc : epc データ
tid : tid データ

2.9.2 rssiReceived

```
- (void)rssiReceived:(uint16_t)rssi;
```

説明 : RSSI 返す時、コールバックします。
startReadTagsWithRssi 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 : rssi : rssi データ

2.9.3 didSetOutputPowerLevel

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

説明 : Tx パワーを設定する時、コールバックします。
setOutputPowerLevel メソッドを呼び出すと、実行結果をコールバックします。
デリゲートメソッド内に 2S の遅延が経った後、次の処理を行います。

引数 : status : 成功 (0x00) ; 失敗 (0x00 以外)

2.9.4 didSetChParamReceived

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

説明 : チャンネルを設定する時、コールバックします。
setChannel 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 : status : 成功 (0x00) ; 失敗 (0x00 以外)

2.9.5 didSetAntiCol

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

説明 : アンチコリジョンモードを設定する時、コールバックします。
setAnticollision 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 : status : 成功 (0x00) ; 失敗 (0x00 以外)

2.9.6 didSetSession

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

説明：session を設定する時、コールバックします。
SetSession 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： status：成功(0x00)；失敗(0x00 以外)

2.9.7 sessionReceived

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

説明：session を取得する時、コールバックします。
getSession 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： session： S0(0x00)、S1(0x01)、S2(0x02)、S3(0x03)、Dev. mode(0xF0)。

2.9.8 channelReceived

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

説明：チャンネルを取得する時、コールバックします。
getChannel メソッドを呼び出すと、実行結果をコールバックします。
RFID のチャンネル値を CommonReaderInfo.nRFIDchannel にセットします。
引数：無効

2.9.9 anticollParamReceived

```
- (void)anticollParamReceived:(uint8_t)mode Counter:(uint8_t)counter;
```

説明：アンチコリジョンを取得する時、コールバックします。
getAnticollision 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： mode：成功(0x00)、(失敗(0x00 以外))
counter：カウンタ

2.9.10 txPowerLevelReceived

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

説明：OutputPowerLevel を取得する時、コールバックします。
getOutputPowerLevel メソッドを呼び出すと、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 クラスにセットします。
rRFIDpower：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

注意 : ASX-300R の開発を行う時、CommonReaderInfo クラスは fRFIDpower 値のみ返します。

2.9.11 regionReceived

```
- (void)regionReceived: (uint8_t)region;
```

説明 : RFID モジュールエリアを取得する時、コールバックします。
getRegion 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 : region : korea(0x11)、North america(0x21)、US(0x22)、Europe(0x31)、Japan(0x41)、China1(0x51)、China2(0x52)、Brazil(0x61)

2.9.12 onOffTimeChanged

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

説明 : ON/OFF time を設定する時、コールバックします。
getRFIDOnOffTime メソッドを呼び出すと、実行結果をコールバックします。
デリゲートメソッド内に 2S の遅延が経った後、次の処理を行います。

2.9.13 fhLbtReceived

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

説明 : FH と LBT を取得する時、コールバックします。
getFhLbtParam/getRFIDOnOffTime 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 : fhLb : 読取時間(16bit)、アイドル時間(16bit)、キャリアセンスタイム(16bit)、ターゲット RF パワーレベル (16bit)、FH(8bit)、LBT(8bit)、CW(8bit)

2.9.14 hoppingTableReceived

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

説明 : FH テーブルを取得する時、コールバックします。
getFreqHoppingTable 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 : table : テーブルサイズ (8bit) 。

2.9.15 didSetFhLbt

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

説明：FH と LBT を設定する時、コールバックします。

setFreqHoppingTable メソッドを呼び出すと、実行結果をコールバックします。
デリゲートメソッド内に 2S の遅延が経った後、次の処理を行います。

引数： status：成功(0x00)；失敗(0x00 以外)

2.9.16 didSetOptiFreqHPTable

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

説明：FH テーブルを設定する時、コールバックします。

setFreqHoppingTable 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： status：成功(0x00)；失敗(0x01)

2.9.17 didSetFHmodeChanged

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

説明：FH モードを設定する時、コールバックします。

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

2.9.18 resGetFHmode

```
- (void) resGetFHmode;
```

説明：FH モードを取得する時、コールバックします。

getFrequencyHoppingMode メソッドを呼び出すと、実行結果をコールバックします。
RFID のホッピングモードを CommonReaderInfo. bSmartHopping にセットします。

2.9.19 rfidModuleVersionReceived

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

説明：RFID モジュールバージョンを取得する時、コールバックします。

getRFIDModuleVersion メソッドを呼び出すと、実行結果をコールバックします。
RFID のモジュールバージョンを CommonReaderInfo. strRFIDModuleVersion にセットします。

2.9.20 rfidOnOffTimeReceived

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

説明 : getRFIDOnOffTime/getFhLbtParam メソッドを呼び出すと、実行結果をコールバックします。RFID の RF 値を CommonReaderInfo クラスにセットします。

nRFIDonTime : 読み取り時間
nRFIDoffTime : アイドル時間
nCst : キャリアセンスタイム
nRfl : RF パワーレベル
nLbt : LBT
nFh : RFID ホッピング
nCw : RFID 連続スキャン

引数 : 無効

2.9.21 writedReceived

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

説明 : タグを書き込み時、コールバックします。

writeToTagMemory 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 : statusCode: 成功 (0x00) ; 失敗 (0x00 以外)

2.9.22 tagMemoryReceived

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

説明 : タグデータを読み取り時、コールバックします。

readFromTagMemory を呼び出すと、実行結果をコールバックします。

引数 : RFU エリアのデータを返す。

2.9.23 killedReceived

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

説明 : タグをキル時、コールバックします。

killTag 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 : statusCode: 成功 (0x00) ; 失敗 (0x00 以外)

2.9.24 lockedReceived

- (void)lockedReceived:(uint8_t)statusCode;

説明：タグをロックする時、コールバックします。
lockTagMemory 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：statusCode: 成功(0x00) ; 失敗(0x0 以外)

2.9.25 updatedRegistry

- (void)updatedRegistry:(uint8_t)statusCode;

説明：レジストリを更新する時、コールバックします。
updateRegistry 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：statusCode: 成功(0x00) ; 失敗(0x00 以外)

2.9.26 pcEpcReceived

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

説明：タグを読み取り時、コールバックします。
startScan (ComboRFIDApi) 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：pcEpc データ

2.9.27 pcEpcRssiReceived

- (void)pcEpcRssiReceived:(NSData *)pcEpc rssi:(int8_t)rssi;

説明：タグを読み取り時、コールバックします。
startReadTagsWithRssi を呼び出すと、実行結果をコールバックします。
引数：rssi:rssi 数据。

2.9.28 didSetBeep

- (void)didSetBeep:(uint8_t)status;

説明：setBeep を呼び出すと、実行結果をコールバックします。
引数：成功(0x00)

2.9.29 didSetStopCon

- (void)didSetStopCon:(uint8_t)status;

説明 : setStopConditionTagNum 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 : 成功 (0x00)

2.9.30 stopConditionsReceived

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

説明 : getStopCondition 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 : タグカウント (1 バイト)、スキャン時間 (1 バイト)、サイクル (1 バイト)。

2.9.31 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。

getSelectParam 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)。

2.9.32 queryParamReceived

- (void)queryParamReceived:(NSData *)qryParam;

説明 : クエリコマンドを取得する時、コールバックします。

getQueryParam 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 : DR (1 桁)、M (2 桁)、TRext (1 桁)、Sel (2 桁)、会話 (2 桁)、ターゲット (1 桁)、Q (4 桁)。

2.9.33 freqHPTableReceived

- (void)freqHPTableReceived:(uint8_t)status;

説明 : FH と LBT を取得する時、コールバックします。

getFhLbtParam 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 : FH : 起動 (0x01 または 0x01 以上)、使用禁止 (0x00)。

2.9.34 didSetFreqHPTable

- (void)didSetFreqHPTable:(uint8_t)status;

説明：FH と LBT を設定する時、コールバックします。

setOptimumFrequencyHoppingTable/setSmartHoppingOnOff 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：status：成功 (0x00)

2.9.35 ackReceived

- (void)ackReceived:(uint8_t)commandCode;

説明：RSSI フィルターもしくは PA ゲインモードを設定する時、コールバックします。

setRSSIThreshold/setPAGainMode 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：commandCode：成功 (0x00)

2.9.36 rssiThresholdReceived

- (void)rssiThresholdReceived:(uint16_t)threshold;

説明：RSSI フィルター設定を取得する時、コールバックします。

getRSSIThreshold 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：threshold：RSSI フィルター値

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

説明：AsReader デバイス情報を取得する時、コールバックします。
getReaderInfo メソッドを呼び出すと、実行結果をコールバックします。AsReader の情報を CommonReaderInfo クラスにセットします。

ASX-300R の開発を行う場合：

- bBeep：ピープ音
- fRFIDpower：現在の RFID アウトプットパワー
- fRFIDpowerMax：RFID 最大アウトプットパワー
- fRFIDpowerMin：RFID 最小アウトプットパワー
- nRFIDonTime：読み取り時間
- nRFIDoffTime：アイドル時間

ASX-300R 以外の開発を行う場合：

- nRFIDonTime：読み取り時間
- nRFIDoffTime：アイドル時間
- bBeep：ピープ音
- fRFIDpower：現在の RFID アウトプットパワー
- fRFIDpowerMax：RFID 最大アウトプットパワー
- fRFIDpowerMin：RFID 最小アウトプットパワー
- nCst：キャリアセンスタイム
- nRfl：RF パワーレベル
- nFh：RFID ホッピング
- nLbt：LBT
- nCw：RFID 連続スキャン

引数：data：無効