AsReaderGUN SDK

Reference Guide V1.4

ASR-R250G、ASR-L251G 共通開発マニュアル

Asterisk, Inc.
2024-09-05

修正履歴

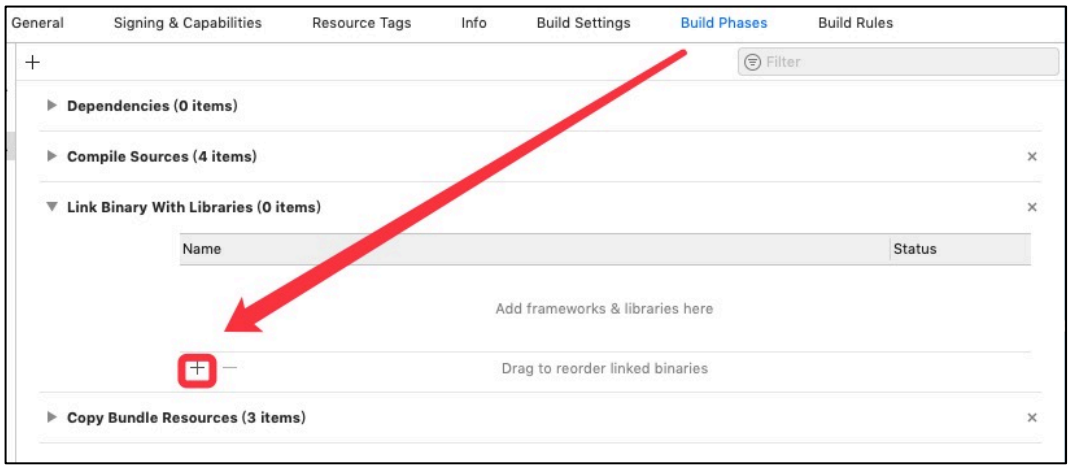
No.	バージョン	修正内容	日付
1	1.0	新規作成	2020/08/24
2	1.1	附録を追加	2020/10/16
3	1.2	メソッド「stopDecode」を削除 メソッド「setBarcodeMode (With Custom prefix suffix)」を追加	2021/05/06
4	1.3	「continuousMode」の説明を追加	2022/4/1
5	1.4	3.2.4、4.1 章節に注意事項を追加 getTagState、TagState の説明を追加	2024/9/5

目　次

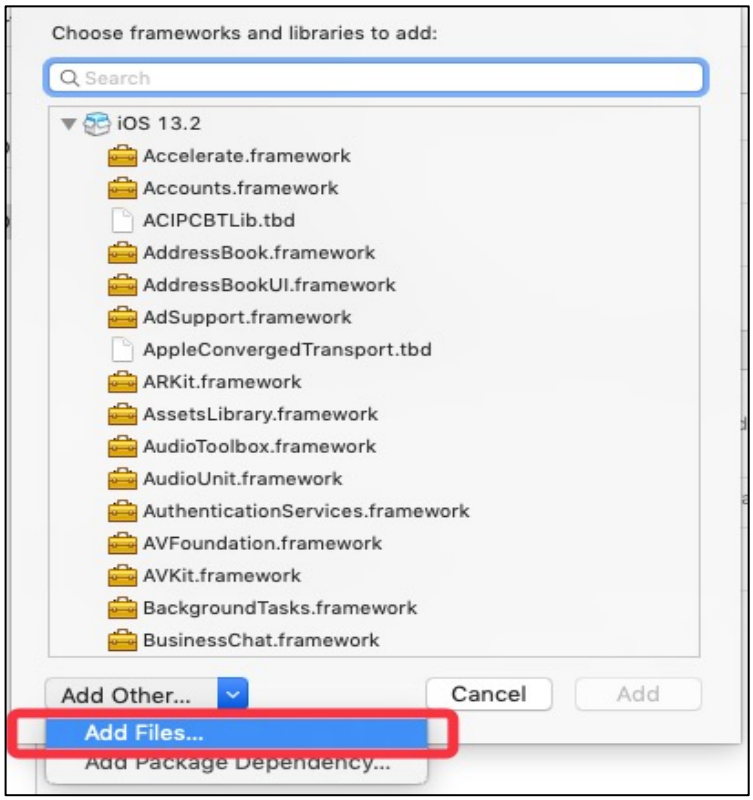
1 SDK の使用

1.1 SDK を追加

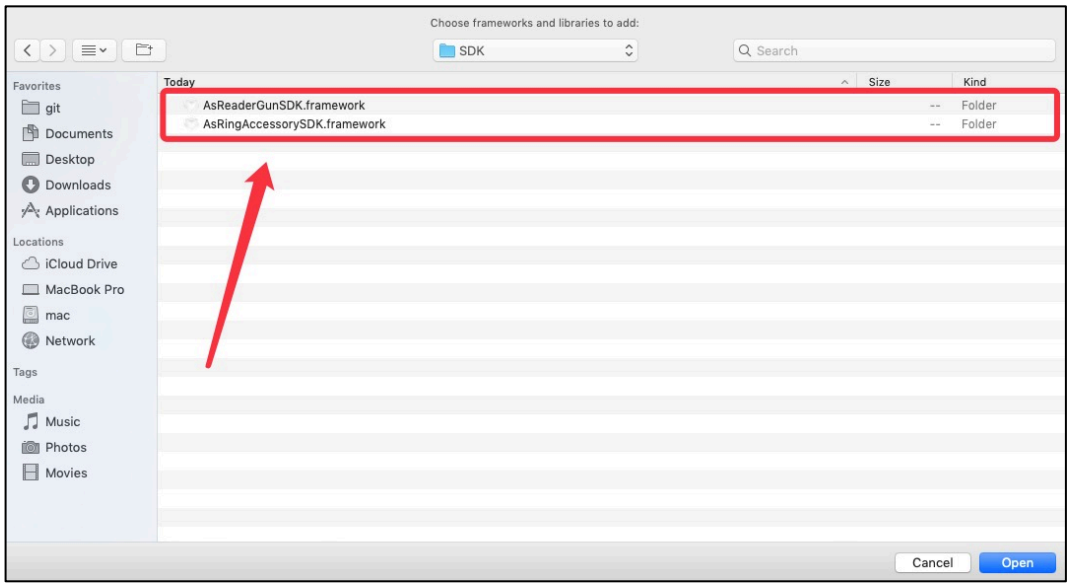
1.1.1 TARGET -> Build phases -> Link Binary With Libraries



1.1.2「Add Other…」、「Add Files…」を選択



1.1.3 AsReaderGUNSDK.framework、AsRingAccessorySDK.framework を追加



追加後、下記のようになります。

1.1.4 プロトコルの追加

info.plist ファイルの「Supported external accessory protocols」オプションで以下のデバイスに対応するプロトコルを追加します。

jp.co.asx.asring.plus

jp.co.asx.asreader.gun

1.2.SDK の Import

ObjectC オブジェクトは SDK を使うクラスにヘッダーファイルを引用する必要があります。

以下をサンプルとしてご参照ください。

#import <AsReaderGUNSDK/AsReaderGUNSDK.h>

2 AsReaderGUN Class

2.1 Properties

2.1.1 @property(strong, nonatomic) NSString *deviceModel;

説明：
AsReaderGUN アクセサリの接続プロトコルを取得・設定します。（例：com.asreader.gun）

2.1.2 @property (assign,nonatomic,readonly) BOOL isConnect;

説明：
AsReaderGUN の接続ステータスを取得します。
YES：接続中
NO：未接続

2.2 Function

2.2.1 initWithDeviceModel

関数名	- (instancetype)initWithDeviceModel:(NSString *)deviceModel;		
引数名	IN/OUT	型	説明
deviceModel	IN	NSString	AsReaderGUN アクセサリの接続プロトコル（例：com.asreader.gun）
-	OUT	AsReaderGUN	AsReaderGUN オブジェクト
■メソッドの説明： 本メソッドで AsReaderGUN オブジェクトを初期化します。 ■サンプルコード： AsReaderGUN * AsReaderGUN = [[AsReaderGUN alloc] initWithDeviceModel:@"com.asreader.gun"];			

2.2.2 deviceModel

関数名	- (NSString *)deviceModel;		
引数名	IN/OUT	型	説明
-	OUT	NSString	AsReaderGUN アクセサリの接続プロトコル
■メソッドの説明： AsReaderGUN アクセサリの接続プロトコルを取得します。 ■サンプルコード： NSString * deviceModel = [AsReaderGUN deviceModel];			

2.2.3 address

関数名	- (NSString *)address;		
引数名	IN/OUT	型	説明
-	OUT	NSString	AsRing+が AsReaderGUN に割り当てたアドレス
■メソッドの説明： AsRing+が AsReaderGUN に割り当てたアドレスを取得します。 注：L250G 専用メソッド ■サンプルコード： NSString * address = [AsReaderGUN address];			

2.2.4 getAsReaderGUNVersion

関数名	- (NSString *)getAsReaderGUNVersion;		
引数名	IN/OUT	型	説明
-	OUT	NSString	SDK バージョン
■メソッドの説明： SDK バージョンを取得します。 ■サンプルコード： NSString * sdkVersion= [AsReaderGUN getAsReaderGUNVersion];			

3 AsReader Class

3.1 Properties

パラメータの保存位置及びディフォルト値について「附録 11」をご参照

3.1.1 @property (nonatomic, strong) AsReaderGUN *mAsReaderGUN;

説明：
AsReaderGUN のインスタンスを取得・設定します。

3.1.2 @property (nonatomic, assign) BuzzerState buzzer;

説明：
AsReader のブザーステータスを取得・設定します。
列挙型 BuzzerState（5.1.3 をご参照ください）

3.1.3 @property (nonatomic, assign) VibratorState vibrator;

説明：
AsReader のバイブレーターのステータスを取得・設定します。
列挙型 VibratorState（5.1.4 をご参照ください）

3.1.4 @property (nonatomic, assign) int operationTime;

説明：
AsReaderGUN の RFID モジュールの稼働時間を取得・設定します。（単位：ms）

3.1.5 @property (nonatomic, assign) int inventoryTime;

説明：
AsReaderGUN の RFID モジュールのインベントリタイムを取得・設定します。（単位：ms）

3.1.6 @property (nonatomic, assign) int idleTime;

説明：
AsReaderGUN の RFID モジュールのアイドルタイムを取得・設定します。（単位：ms）

3.1.7 @property (nonatomic, assign) int sleepTime;

説明：
AsReaderGUN の自動スリープタイムを取得・設定します。（単位：s）
注：ASR-R250G の場合、デバイスとの接続を切断してからカウントします。
ASR-L251G の場合、接続成功の状態で、インベントリをしない時点からカウントします。

3.1.8 @property (nonatomic, assign) int autoOffTime;

説明：
AsReaderGUN の接続を切断後のオートオフの時間を取得・設定します。（単位：s）

3.1.9 @property (nonatomic, strong) NSString *accessPassword;

説明：
AsReaderGUN はロックされたタグをアクセスした時のアクセスパスワードを取得・設定します。

3.1.10 @property (nonatomic, assign) SessionType inventorySession;

説明：
AsReaderGUN の RFID モジュールの Session 値を取得・設定します。
列挙型 SessionType（5.1.5 をご参照ください）

3.1.11 @property (nonatomic, assign) SessionFlag sessionFlag;

説明：
AsReaderGUN の RFID モジュールの Session Flag を取得・設定します。
列挙型 SessionFlag（5.1.6 をご参照ください。）

3.1.12 @property (nonatomic, strong) NSString *serialNumber;

説明：
AsReaderGUN デバイスのシリアルコードを取得・設定します。

3.1.13 @property (nonatomic, assign) BOOL continuousMode;

説明：
AsReaderGUN の RFID モジュールは連続読取可否を取得・設定します。
連続読取モードでご使用の際は、アプリ側で明示的に ContinuousMode=YES を設定してください。
ハードの特性上、電源 ON 時または AsReader とアプリが接続された時、連続読取モードは無効になります。
YES：連続読取を有効
NO：連続読取を無効

3.1.14 @property (nonatomic, assign) int powerGain;

説明：
AsReaderGUN の RFID モジュールで読み取る際の Power 値を取得・設定します。
構造体 CMinMaxValue（5.2.1 をご参照ください）

3.1.15 @property (nonatomic, assign) BOOL isUseKeyAction;

説明：
AsReaderGUN はハードウェア秘密鍵を使う要否を取得・設定します。
YES：ハードウェア秘密鍵を使う
NO：ハードウェア秘密鍵を使わない

3.1.16 @property (nonatomic, assign) SelectFlag useSelectionMask;

説明：
AsReaderGUN の RFID モジュールの Select Flag を取得・設定します。
列挙型 SelectFlag（5.1.7 をご参照ください）

3.1.17 @property (nonatomic, assign) BOOL rssiMode;

説明：
AsReaderGUN の RFID モジュールはタグを読み取る時に RSSI データの取得要否を取得・設定します。
YES：RSSI データを取得
NO：RSSI データを取得しない

3.1.18 @property (nonatomic, assign) BOOL epcMaskMatchMode;

説明：
AsReaderGUN の RFID モジュールは EPC Mask フィルターモードを使う要否を取得・設定します。
YES：EPCMask フィルターモードを使う
NO：EPCMask フィルターモードを使わない

3.1.19 @property (nonatomic, assign) AlgorithmType algorithm;

説明：
AsReaderGUN の RFID モジュールの Q 値タイプを取得・設定します。
列挙型 AlgorithmType（3.3.1 をご参照ください）

3.1.20 @property (nonatomic, assign) int minQ;

説明：
AsReaderGUN の RFID モジュールの最小 Q 値を取得・設定します。
値の範囲：0~15

3.1.21 @property (nonatomic, assign) int maxQ;

説明：
AsReaderGUN の RFID モジュールの最大 Q 値を取得・設定します。
値の範囲：0~15

3.1.22 @property (nonatomic, assign) int qValue;

説明：
AsReaderGUN の RFID モジュールの Q 値を取得・設定します。
値の範囲：0~15

3.1.23 @property (nonatomic, assign) int linkProfileValue;

説明：
AsReaderGUN の RFID モジュールの Link Profile 値を取得・設定します。
値の範囲：0~3

3.1.24 @property (nonatomic, assign) int defaultLinkProfileValue;

説明：
AsReaderGUN の RFID モジュールの Default Link Profile 値を取得・設定します。
値の範囲：0~3
注：ASR-R250G のみサポート

3.1.25 @property (nonatomic, assign) int maskTypeValue;

説明：
AsReaderGUN の RFID モジュールの Mask タイプを取得・設定します。
0：NoMask、1：SelectionMask、2：EPCMask

3.2 Function

3.2.1 initWithAsReaderGUN

関数名	-(id)initWithAsReaderGUN:(AsReaderGUN*)device delegate:(id<AsReaderDelegate>)delegate;		
引数名	IN/OUT	型	説明
device	IN	AsReaderGUN	AsReaderGUN オブジェクト
delegate	IN	AsReaderDelegate	AsReaderGUN デバイスからの情報または通知を受け取るデリゲートを設定します。AsReaderGUN のステータス変更などのイベントを処理した際に使います。
-	OUT	AsReader	AsReader オブジェクト

■メソッドの説明：
初期化メソッドです。
注：本メソッドは「AsReaderGUNConnected」通知を受け取ってから呼び出してください。
■サンプルコード：（注：AsReaderGUN は AsReaderGUN クラスのオブジェクト）

```
- (void)AsReaderGUNConnected:(NSNotification *)notification {
    dispatch_async(dispatch_get_main_queue(), ^{
        AsReader * asReader = [[AsReader alloc]
initWithAsReaderGUN:AsReaderGUN delegate:self];
    });
}
```

3.2.2 disconnect

関数名	- (void)disconnect;		
引数名	IN/OUT	型	説明
-	-	-	-

■メソッドの説明：
AsReaderGUN との接続を切断します。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）

```
[asReader disconnect];
```

3.2.3 getAction

関数名	- (CommandType)getAction;		
引数名	IN/OUT	型	説明
-	OUT	CommandType	コマンドタイプ 列挙型 CommandType（6.1.1 をご参照ください）
■メソッドの説明： AsReader インスタンスのアクションステータスを取得します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） CommandType type = [asReader getAction];			

3.2.4 setDelegate

関数名	- (void)setDelegate:(id<AsReaderDelegate>)delegate;		
引数名	IN/OUT	型	説明
delegate	IN	AsReaderDelegate	AsReaderDelegate
■メソッドの説明： AsReaderDelegate を設定します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） [asReader setDelegate:self]; 注：本デリゲートを設定しないと、データを返しません。			

3.2.5 setScanMode

関数名	- (void)setScanMode:(ScanMode)scanMode;		
引数名	IN/OUT	型	説明
scanMode	IN	ScanMode	スキャンモード 列挙型 ScanMode（6.1.2 をご参照ください）
■メソッドの説明： AsReaderGUN のスキャンモードを設定します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） [asReader setScanMode: RFIDScanMode];			

3.2.6 getScanMode

関数名	- (ScanMode)getScanMode;		
引数名	IN/OUT	型	説明
-	OUT	ScanMode	スキャンモード 列挙型 ScanMode（6.1.2 をご参照ください）
■メソッドの説明： AsReaderGUN のスキャンモードを取得します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） ScanMode mode = [asReader getScanMode];			

3.2.7 inventory

関数名	- (ResultCode)inventory;		
引数名	IN/OUT	型	説明
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
インベントリを開始します。
changeActionState（4.4 をご参照）デリゲート、readTag（4.3 をご参照）デリゲートにコールバックされます。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）
ResultCode resultCode = [asraeder inventory];
if (resultCode == ResultNoError) {
　　//メソッドの呼び出しに成功
}else{
　　//メソッドの呼び出しに失敗
}			

3.2.8 readMemory

関数名	-(ResultCode)readMemory:(MemoryBank)bank offset:(int)offset length:(int)length;		
引数名	IN/OUT	型	説明
bank	IN	MemoryBank	タグのメモリーバンク 列挙型 MemoryBank（5.1.2 をご参照ください）
offset	IN	int	タグのスタートアドレス（単位：word）
length	IN	int	タグの長さ(単位：word)
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：

タグの特定メモリーバンクのデータを読み取ります。

accessResult（4.7 をご参照）のデリゲートメソッドにコールバックされます。

■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder readMemory:Bank_EPC offset:16 length:4];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.9 writeMemory

関数名	-(ResultCode)writeMemory:(MemoryBank)bank offset:(int)offset value:(NSString *)value;		
引数名	IN/OUT	型	説明
bank	IN	MemoryBank	タグのメモリーバンク 列挙型 MemoryBank（5.1.2 をご参照ください）
offset	IN	int	タグのスタートアドレス （単位：word）
value	IN	NSString	書き込むデータ（16 進数の文字列）
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：

タグの特定メモリーバンクのデータを変更します。

accessResult（4.7 をご参照）のデリゲートメソッドにコールバックされます。

■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder writeMemory:Bank_EPC offset:16
value:@"1234"];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.10 lock

関数名	- (ResultCode)lock:(LockParam *)param;		
引数名	IN/OUT	型	説明
param	IN	LockParam	LockParam オブジェクト
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
タグの特定エリアのデータをロックします。
accessResult（4.7 をご参照）のデリゲートメソッドにコールバックされます。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト、param は LockParam クラスのオブジェクト）

```
ResultCode resultCode = [asraeder lock:param];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.11 unlock

関数名	- (ResultCode) unlock:(LockParam *)param;		
引数名	IN/OUT	型	説明
param	IN	LockParam	LockParam オブジェクト
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
タグのロックされたメモリーバンクをアンロックします。アンロックした後、デフォルトパスワードでタグのデータを変更することができます。
accessResult（4.7 をご参照）のデリゲートメソッドにコールバックされます。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト、param は LockParam クラスのオブジェクト）

```
ResultCode resultCode = [asraeder unlock:param];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.12 permaLock

関数名	- (ResultCode)permaLock:(LockParam *)param;		
引数名	IN/OUT	型	説明
param	IN	LockParam	LockParam オブジェクト
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
タグの特定メモリーバンクのデータを永久にロックします。永久ロックをした後、タグのデータは変更できなくなり、アンロックもできません。
accessResult（4.7 をご参照）のデリゲートメソッドにコールバックされます。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト、param は LockParam クラスのオブジェクト）

```
ResultCode resultCode = [asraeder permaLock:param];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.13 kill

関数名	- (ResultCode)kill:(NSString *)killPassword;		
引数名	IN/OUT	型	説明
killPassword	IN	NSString	Kill パスワード
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
タグをキルします。
キルされたタグは使えなくなります。
accessResult（4.7 をご参照）のデリゲートメソッドにコールバックされます。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder kill: @"00000000"];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.14 stop

関数名	- (ResultCode)stop;		
引数名	IN/OUT	型	説明
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
インベントリを停止します。
changedActionState（4.4 をご参照）のデリゲートメソッドにコールバックされます。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder stop];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.15 stopSync

関数名	- (ResultCode)stopSync;		
引数名	IN/OUT	型	説明
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode (5.1.1 をご参照ください)

■メソッドの説明：
スキャン停止を同期に実行します。
changedActionState (4.4 をご参照) のデリゲートメソッドにコールバックされます。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder stopSync];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.16 defaultParameter

関数名	- (ResultCode)defaultParameter;		
引数名	IN/OUT	型	説明
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode (5.1.1 をご参照ください)

■メソッドの説明：
全てパラメータの設定をデフォルトに復帰します。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder defaultParameter];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.17 saveParameter

関数名	- (ResultCode)saveParameter;		
引数名	IN/OUT	型	説明
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
全てのパラメータをメモリーバンクに保存します。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）
ResultCode resultCode = [asraeder saveParameter];
if (resultCode == ResultNoError) {
　//メソッドの呼び出しに成功
}else{
　//メソッドの呼び出しに失敗
}			

3.2.18 regionName

関数名	- (NSString *)regionName;		
引数名	IN/OUT	型	説明
-	OUT	NSString	地域名
■メソッドの説明： AsReaderGUN の地域情報を取得します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） NSSting * regionName = [asraeder regionName];			

3.2.19 serialNumber

関数名	- (NSString *)serialNumber;		
引数名	IN/OUT	型	説明
-	OUT	NSString	シリアル番号
■メソッドの説明： AsReaderGUN のシリアル番号を取得します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） NSSting * serialNumber = [asraeder serialNumber];			

3.2.20 rFModuleVersion

関数名	- (NSString *)rFModuleVersion;		
引数名	IN/OUT	型	説明
-	OUT	NSString	RF モジュールのバージョン
■メソッドの説明： AsReaderGUN の RF モジュールのバージョンを取得します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） NSSting * rFModuleVersion = [asraeder rFModuleVersion];			

3.2.21 firmwareVersion

関数名	- (NSString *)firmwareVersion;		
引数名	IN/OUT	型	説明
-	OUT	NSString	FW のバージョン
■メソッドの説明： AsReaderGUN の FW バージョンを取得します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） NSSting * firmwareVersion = [asraeder firmwareVersion];			

3.2.22 powerGainRange

関数名	- (CMinMaxValue)powerGainRange;		
引数名	IN/OUT	型	説明
-	OUT	CMinMaxValue	Power の設定範囲 構造体 CMinMaxValue（5.21 をご参照ください）

■メソッドの説明：
AsReaderGUN は設定できる最大と最小のパワー値を取得します。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）
int min = asReader.powerGainRange.min; //最小パワー値
int max = asReader.powerGainRange.max; //最大パワー値			

3.2.23 batteryStatus

関数名	- (int)batteryStatus;		
引数名	IN/OUT	型	説明
-	OUT	int	バッテリー残量 値の範囲：0、1、2、3、4
■メソッドの説明： AsReaderGUN のバッテリー残量を取得します。 ■サンプルコード： int battery = [asReader batteryStatus];			

3.2.24 clearEpcMask

関数名	- (ResultCode)clearEpcMask;		
引数名	IN/OUT	型	説明
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
AsReaderGUN メモリーにある EPC マスクデータを削除します。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）
ResultCode resultCode = [asraeder clearEpcMask];
if (resultCode == ResultNoError) {
　　//メソッドの呼び出しに成功
}else{
　　//メソッドの呼び出しに失敗
}			

3.2.25 saveEpcMask

関数名	- (ResultCode)saveEpcMask;		
引数名	IN/OUT	型	説明
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
追加された EPC マスクデータを AsReaderGUN のメモリーに保存します。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder saveEpcMask];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.26 epcMaskCount

関数名	- (int)epcMaskCount;		
引数名	IN/OUT	型	説明
-	OUT	int	EPC マスクの数量

■メソッドの説明：
AsReaderGUN メモリーに保存されている EPC マスクの数量を取得します。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）

```
int epcMaskCount = [asReader epcMaskCount];
```

3.2.27 addEpcMask

関数名	-(ResultCode)addEpcMask:(int)offset length:(int)length mask:(NSString *)mask;		
引数名	IN/OUT	型	説明
offset	IN	int	マスクのスタートアドレス（単位:bit）
length	IN	int	マスクの長さ（単位:bit）
mask	IN	NSString	マスクデータ（16 進数の文字列）
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
EPC マスクを追加します。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder addEpcMask:32 length:16 mask:@"1234"];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.28 addEpcMask

関数名	- (ResultCode)addEpcMask:(AsSelectMaskEPCParam *)mask;		
引数名	IN/OUT	型	説明
mask	IN	AsSelectMaskEPCParam	AsSelectMaskEPCParam オブジェクト
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
AsSelectMaskEPCParam オブジェクトの形式で EPC マスクを追加します。
■サンプルコード：（注：asReader は AsReader クラスのオブジェクト、mask は AsSelectMaskEPCParam クラスのオブジェクト）

```
ResultCode resultCode = [asraeder addEpcMask:mask];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.29 getEpcMask

関数名	- (AsSelectMaskEPCParam *)getEpcMask:(int)index;		
引数名	IN/OUT	型	説明
index	IN	int	EPC マスクのインデックス
-	OUT	AsSelectMaskEPCParam	AsSelectMaskEPCParam オブジェクト
■メソッドの説明： 指定したインデックスの EPC マスクを取得します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） AsSelectMaskEPCParam * mask= [asraeder getEpcMask:0];			

3.2.30 getLBTMask

関数名	- (NSArray *)getLBTMask;		
引数名	IN/OUT	型	説明
-	OUT	NSArray	マスク配列
■メソッドの説明： 周波数テーブル情報のマスク配列を取得します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） NSArray * lbtArray = [asraeder getLBTMask];			

3.2.31 getLBT

関数名	- (NSArray *)getLBT;		
引数名	IN/OUT	型	説明
-	OUT	NSArray	周波数テーブル情報
■メソッドの説明： 周波数テーブル情報を取得します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） NSArray * lbtArray = [asraeder getLBT];			

3.2.32 setLBT

関数名	- (void)setLBT:(NSArray *)table;		
引数名	IN/OUT	型	説明
table	IN	NSArray	周波数テーブル情報 配列要素は LbtItem オブジェクト（5.7 をご参照ください）
■メソッドの説明： 周波数テーブル情報を設定します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト、array は NSArray（要素は LbtItem オブジェクト）クラスのオブジェクト） [asraeder setLBT:array];			

3.2.33 getLBTFrequency

関数名	- (NSString *)getLBTFrequency:(int)slot;		
引数名	IN/OUT	型	説明
slot	IN	int	LBT 周波数テーブルの周波数位置
-	OUT	NSString	周波数情報
■メソッドの説明： 周波数情報を取得します。 ■サンプルコード：（注：asReader は AsReader クラスのオブジェクト） NSString * frequency =[asraeder getLBTFrequency:0];			

3.2.34 startBuzzerWithBuzzerTime

関数名	- (ResultCode)startBuzzerWithBuzzerTime:(int)buzzerTime;		
引数名	IN/OUT	型	説明
buzzerTime	IN	int	ブザータイム （単位：ms）
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
ブザーの再生開始。
※再生されたブザー音は一定時間再生され停止します。
■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder startBuzzerWithBuzzerTime:100];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.35 startVibratorWithVibratorTime

関数名	- (ResultCode)startVibratorWithVibratorTime:(int)vibratorTime;		
引数名	IN/OUT	型	説明
vibratorTime	IN	int	バイブレーションタイム （単位：ms）
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
バイブレーションの開始。
※バイブレーションは一定時間動作した後、停止します。
■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder startVibratorWithVibratorTime:100];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.36 startDecode

関数名	- (ResultCode)startDecode;		
引数名	IN/OUT	型	説明
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
バーコードスキャンを開始します。
changedActionState デリゲートメソッド（4.4 をご参照）、detectBarcode デリゲートメソッド（4.5、4.6 をご参照）にコールバックされます。
注：
バーコードスキャンを停止させる場合は、「3.2.15 stopSync」を使用してください。

■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder startDecode];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.37 setBarcodeParam

関数名	- (ResultCode)setBarcodeParam:(AsParamValue *)paramData;		
引数名	IN/OUT	型	説明
paramData	IN	AsParamValue	AsParamValue オブジェクト
-	OUT	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
バーコードのコンフィグ情報を設定します。
■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト、asParamValue は AsParamValue クラスのオブジェクト）

```
ResultCode resultCode = [asraeder setBarcodeParam: asParamValue];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.38 getBarcodeParam

関数名	- (AsParamValue *)getBarcodeParam:(NSNumber *)paramData;		
引数名	IN/OUT	型	説明
paramData	IN	NSNumber	取得したいバーコードパラメータ。指定する値は「8.2.1 ParamName をご参照ください」
-	OUT	AsParamValue	AsParamValue オブジェクト

■メソッドの説明：
バーコードのコンフィグ情報を取得します。
■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト）

```
AsParamValue * asParamValue= [asraeder getBarcodeParam: [NSNumber numberWithInt:UPCA]];
```

3.2.39 usedSelectionMask

関数名	- (BOOL)usedSelectionMask☹int)index;		
引数名	IN/OUT	型	説明
index	IN	int	マスクのインデックス
-	OUT	BOOL	指定 Index の Mask 使用可否設定 YES：使用中 NO：未使用

■メソッドの説明：
AsReaderGUN に指定した Mask は使用可否を取得します。
■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト）

```
BOOL isUsed = [asraeder usedSelectionMask:0];
if (isUsed) {
   //接続中の AsReaderGUN の mask は使用されている
}else{
   //接続中の AsReaderGUN の mask は使用されていない
}
```

3.2.40 getSelectionMask

関数名	- (AsSelectMaskParam *)getSelectionMask:(int)index;		
引数名	IN/OUT	型	説明
index	IN	int	マスクインデックス
-	OUT	AsSelectMaskParam	AsSelectMaskParam オブジェクト

■メソッドの説明：
指定したインデックスの Selection マスクを取得します。
■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト）

```
AsSelectMaskParam * asSelectMaskParam = [asraeder getSelectionMask:0];
```

3.2.41 setSelectionMask

関数名	- (void)setSelectionMask:(int)index withParam:(AsSelectMaskParam *)param;		
引数名	IN/OUT	型	説明
index	IN	int	マスクインデックス
param	IN	AsSelectMaskParam	AsSelectMaskParam オブジェクト
■メソッドの説明： 指定したインデックスに Selection マスク情報を設定します。 ■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト、asSelectMaskParam は AsSelectMaskParam クラスのオブジェクト） [asraeder setSelectionMask:0 withParam: asSelectMaskParam];			

3.2.42 removeSelectionMask

関数名	- (void)removeSelectionMask:(int)index;		
引数名	IN/OUT	型	説明
index	IN	int	マスクインデックス
■メソッドの説明： AsReaderGUN に指定したインデックスの Selection マスク情報を削除します。 ■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト） [asraeder removeSelectionMask:0];			

3.2.43 clearSelectionMask

関数名	- (void)clearSelectionMask;		
引数名	IN/OUT	型	説明
-	-	-	-
■メソッドの説明： AsReaderGUN の全ての Selection マスク情報を削除します。 ■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト） [asraeder clearSelectionMask];			

3.2.44 getAlgorithm

関数名	- (AlgorithmType)getAlgorithm;		
引数名	IN/OUT	型	説明
-	OUT	AlgorithmType	Q 値のタイプ 列挙型 AlgorithmType（3.3.1 をご参照ください）

■メソッドの説明：
現在のアンチコリジョンモードを取得します。
■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト）

```
AlgorithmType type= [asraeder getAlgorithm];
if (type = FixedQ) {
   //FixedQ モード
}else if (type = DynamicQ) {
   //DynamicQ モード
}
```

3.2.45 setBarcodeMode

関数名	- (ResultCode)setBarcodeMode:(BOOL)enabled isKeyAction:(BOOL)isKeyOn;		
引数名	IN/OUT	型	説明
enabled	IN	BOOL	バーコードモードを有効・無効する YES：有効 NO：無効
isKeyOn	IN	BOOL	使用しない
-	OUT	ResultCode	メソッドの実行結果

■メソッドの説明：
現在のモードをバーコードモードに設定します。
注：
バーコードにプレフィックス・サフィックスを追加する場合は、このメソッドを使用しないでください。
「3.2.46 setBarcodeMode(With Custom prefix suffix)」を使用してください。
バーコードモードを有効に設定した場合、RFID 関連コマンドを実行できなくなり、バーコード関連のコマンドしか実行できません。
また、バーコードモードを無効に設定した場合は同様に、RFID 関連コマンドのみが実行できます。
■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder setBarcodeMode:YES isKeyAction:YES];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.46 setBarcodeMode(With Custom prefix suffix)

関数名	- (ResultCode)setBarcodeMode:(BOOL)enabled isKeyAction:(BOOL)isKeyOn isCustomPreSuffixOn:(BOOL)isCustomPreSuffixOn;		
引数名	IN/OUT	型	説明
enabled	IN	BOOL	バーコードモードを有効・無効する YES：有効 NO：無効
isKeyOn	IN	BOOL	使用しない
isCustomPreSuffixOn	IN	BOOL	カスタマイズプレフィックス・サフィックスを無効・有効
-	OUT	ResultCode	メソッドの実行結果

■メソッドの説明：

バーコードモードに設定し、且つカスタマイズプレフィックス/サフィックスできるように設定します。

注：

バーコードにプレフィックス・サフィックスを追加する場合は、このメソッドを使用してください。前述の「3.2.45 setBarcodeMode」を使用しないでください。

バーコードモードを有効に設定した場合、RFID 関連コマンドを実行できなくなり、バーコード関連のコマンドしか実行できません。

また、バーコードモードを無効に設定した場合は同様に、RFID 関連コマンドのみが実行できます。

■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト）

```
ResultCode resultCode = [asraeder setBarcodeMode:YES isKeyAction:YES isCustomPreSuffixOn:YES];
if (resultCode == ResultNoError) {
    //メソッドの呼び出しに成功
}else{
    //メソッドの呼び出しに失敗
}
```

3.2.47 isBarcodeModule

関数名	- (BOOL)isBarcodeModule;		
引数名	IN/OUT	型	説明
-	OUT	BOOL	Barcode モジュールのサポート状態の取得 YES：サポート NO：サポートしていない

■メソッドの説明：
AsReaderGUN は Barcode モジュールのサポート状態を取得します。
■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト）

```
BOOL isSupportBarcodeModule = [asraeder isBarcodeModule];
if (isSupportBarcodeModule) {
   // Barcode モジュールをサポート
}else{
   // Barcode モジュールを未サポート
}
```

3.2.48 isRFIDModule

関数名	- (BOOL)isRFIDModule;		
引数名	IN/OUT	型	説明
-	OUT	BOOL	RFID モジュールのサポート状態の取得 YES：サポート NO：サポートしない

■メソッドの説明：
AsReaderGUN は RFID モジュールのサポート状態を取得します。
■サンプルコード：（注：asraeder は AsReader クラスのオブジェクト）

```
BOOL isSupportRFIDModule = [asraeder isRFIDModule];
if (isSupportRFIDModule) {
   //RFID モジュールをサポート
}else{
   //RFID モジュールを未サポート
}
```

3.2.49 getTagState

関数名	- (TagState)getTagState:(NSString *)pcEPCData;		
引数名	IN/OUT	型	説明
pcEPCData	IN	NSString	タグの PCEPC データ
-	OUT	TagState	メソッドの呼び出し結果 列挙型 TagState（5.1.11 を参照）
■メソッドの説明： RF タグの EPC の Lock ステータスを取得します。 ■サンプルコード（注：asReader は AsReader クラスのオブジェクト，pcEpc は String クラスのオブジェクト） TagState state = [mReader getTagState:pcEpc];			

3.3 Enum

3.3.1 AlgorithmType

定義	説明
FixedQ = 0	Fixed Q
DynamicQ = 1	Dynamic Q

4 AsReaderDelegate Class

4.1 readerInitialized

関数名	- (void)readerInitialized:(AsReader *)reader;		
引数名	IN/OUT	型	説明
reader	IN	AsReader	AsReader オブジェクト

■メソッドの説明：
AsReaderGUN との接続が成功した際に呼び出され、初期化された AsReader オブジェクトを取得することができます。
注：本メソッドのコールバックにメソッド「SetDelegate（3.2.4）」を呼び出さないとデータを返しません。
■サンプルコード：

```
- (void)readerInitialized:(AsReader *)reader {
}
```

4.2 updateDeviceState

関数名	- (void)updateDeviceState:(ResultCode)error;		
引数名	IN/OUT	型	説明
error	IN	ResultCode	列挙型 ResultCode（5.1.1 をご参照ください）

■メソッドの説明：
メソッド・プロパティを実行後にエラーが発生した場合コールバックされます。
※コールバックされるメソッド・プロパティは以下の項目をご参照下さい。
3.1.2~3.1.25、3.2.1、3.2.16、3.2.18~3.2.21、3.2.23~3.2.33、3.2.37~3.2.46
■サンプルコード：

```
- (void)updateDeviceState:(ResultCode)error {
    NSString * errorMessage = [AsResultCode msg:error];
}
```

4.3 readTag

関数名	- (void)readTag㊍NSString *)tag rssi:(float)rssi phase:(float)phase frequency:(float)frequency;		
引数名	IN/OUT	型	説明
tag	IN	NSString	タグの PCEPC 値（16 進数の文字列）
rssi	IN	float	タグの RSSI 値
phase	IN	float	タグの Phase 値
frequency	IN	float	タグの周波数
■メソッドの説明： 「Inventory」のメソッドを実行後/Trigger キーを押下した際にコールバックされます。 ※「Inventory」の詳細は Inventory（3.2.7）をご参照ください。			

4.4 changedActionState

関数名	- (void)changedActionState:(CommandType)action resultCode:(NSInteger)resultCode;		
引数名	IN/OUT	型	説明
action	IN	CommandType	アクションタイプ 列挙型 CommandType（6.1.1 をご参照ください）
resultCode	IN	NSInteger	コマンドの実行結果 列挙型 ResultCode（5.1.1 をご参照ください）
■メソッドの説明： 「inventory」「stop」のメソッドを実行後にコールバックされ、実行結果を受け取ることができます。 ※「inventory」「stop」の詳細は inventory（3.2.7）、stop（3.2.14）をご参照ください。			

4.5 detectBarcode

関数名	- (void)detectBarcode:(BarcodeType)barcodeType codeId:(NSString *)codeId barcode:(NSString *)barcode;		
引数名	IN/OUT	型	説明
barcodeType	IN	BarcodeType	バーコードタイプ 列挙型 BarcodeType（7.2.1 をご参照ください）
codeId	IN	NSString	バーコードのコード ID
barcode	IN	NSString	バーコードデータ
■メソッドの説明： 「startDecode」 のメソッドを実行後にコールバックされます。 ※「startDecode」の詳細は（3.2.36）をご参照ください）			

4.6 detectBarcode

関数名	- (void)detectBarcode:(BarcodeType)barcodeType codeId:(NSString *)codeId barcodeData:(NSData *)barcodeData;		
引数名	IN/OUT	型	説明
barcodeType	IN	BarcodeType	バーコードタイプ 列挙型 BarcodeType（7.2.1 をご参照ください）
codeId	IN	NSString	バーコードのコード ID
barcodeData	IN	NSData	バーコードデータ
■メソッドの説明： 「startDecode」のメソッドを実行後/Trigger キーを押下した際コールバックされます。 ※「startDecode」の詳細は（3.2.36）をご参照ください）。			

4.7 accessResult

関数名	**- (void)accessResult:(ResultCode)error actionState:(CommandType)action epc:(NSString *)epc data:(NSString *)data rssi:(float)rssi phase:(float)phase frequency:(float)frequency;**		
引数名	**IN/OUT**	**型**	**説明**
error	IN	ResultCode	メソッドの実行結果 列挙型 ResultCode（5.1.1 をご参照ください）
action	IN	CommandType	AsReaderGUN のアクションコマンド 列挙型 CommandType（6.1.1 をご参照ください）
epc	IN	NSString	タグの EPC 値（16 進数）
data	IN	NSString	読み取ったタグのデータ（16 進数）
rssi	IN	float	タグの RSSI 値
phase	IN	float	タグの Phase 値
frequency	IN	float	タグの周波数
■メソッドの説明： 「writeMemory」、「readMemory」、「lock」、「unlock」、「permalock」、「kill」のメソッドを実行後にコールバックされます。 ※「writeMemory」、「readMemory」、「lock」、「unlock」、「permalock」、「kill」の詳細は（3.2.9）（3.2.8）（3.2.10）（3.2.11）（3.2.12）（3.2.13）をご参照ください。			

4.8 onAsReaderLeftModeKeyEvent

関数名	- (BOOL)onAsReaderLeftModeKeyEvent:(BOOL)status;		
引数名	IN/OUT	型	説明
status	IN	BOOL	AsReaderGUN 本体の Mode キー(左)のステータス YES：Mode キーを押下 NO：Mode キーを離す
-	OUT	BOOL	YES：Barcode・RFID モードを切り替える NO：現在のモードを維持する

■メソッドの説明：
AsReaderGUN 本体の左側の Mode キーを押下した際にコールバックされます。
■サンプルコード：
- (BOOL)onAsReaderLeftModeKeyEvent:(BOOL)status{
return YES;
}			

4.9 onAsReaderRightModeKeyEvent

関数名	- (BOOL)onAsReaderRightModeKeyEvent:(BOOL)status;		
引数名	IN/OUT	型	説明
status	IN	BOOL	AsReaderGUN の Mode キー(右)のアクションステータス YES：Mode キーを押下 NO：Mode キーを離した
-	OUT	BOOL	YES：Barcode・RFID モードを切り替える NO：現在のモードを維持する

■メソッドの説明：
AsReaderGUN 本体の右側の Mode キーの押下/離すアクション時にコールバックされます。
■サンプルコード：
- (BOOL) onAsReaderRightModeKeyEvent:(BOOL)status{
return YES;
}			

4.10 onAsReaderTriggerKeyEvent

関数名	- (BOOL)onAsReaderTriggerKeyEvent:(BOOL)status;		
引数名	IN/OUT	型	説明
status	IN	BOOL	AsReaderGUN の Trigger キーのステータス YES：Trigger キーを押下 NO：Trigger キーを離した
-	OUT	BOOL	YES：SDK のデフォルト動作を行う Trigger キーを押下：スキャン開始 Trigger キーを離す：スキャン停止 NO：SDK のデフォルト動作を行わない

■メソッドの説明：
AsReaderGUN の Trigger キーの押下/離すアクション時にコールバックされます。
■サンプルコード：
- (BOOL) onAsReaderTriggerKeyEvent:(BOOL)status{
return YES;
}			

5 AsRfidValues Class

5.1 Enum

5.1.1 ResultCode

定義	説明
ResultNoError = 0x0000	成功
ResultOtherError = 0x0001	未知のエラー
ResultUndefined = 0x0002	未定義エラー
ResultMemoryOverrun = 0x0003	アクセスするメモリーは範囲を超えた
ResultMemoryLocked = 0x0004	タグがロックされた
ResultInsufficientPower = 0x000B	バッテリー不足
ResultNonSpecificError = 0x000F	未定義エラー
ResultInOperation = 0xE000	動作中
ResultOutOfRange = 0xE001	範囲外
ResultNotConnected = 0xE100	デバイス未接続
ResultInvalidParameter = 0xE200	無効なパラメータが送信された
ResultInvalidResponse = 0xE300	無効なパラメータを返った
ResultNotSupportFirmware = 0xEE00	サポートしていないフォームウェア
ResultTimeout = 0xEFFF	タイムアウト
ResultHandleMismatch = 0xF001	ハンドラーが不一致。
ResultCRCError = 0xF002	CRC エラー
ResultNoTagReply = 0xF003	タグからの応答なし
ResultInvalidPassword = 0xF004	無効なパスワード
ResultZeroKillPassword = 0xF005	kill パスワード未設定
ResultTagLost = 0xF006	タグをロスト
ResultCommandFormatError = 0xF007	コマンドフォーマットエラー
ResultReadCountInvalid = 0xF008	読取カウント無効
ResultOutOfRetries = 0xF009	リトライ失敗
ResultParamError = 0xFFFB	パラメータエラー
ResultBusy = 0xFFFC	ビジー状態
ResultInvalidCommand = 0xFFFD	無効なコマンド
ResultLowBattery = 0xFFFE	バッテリー残量が少ない
ResultOperationFailed = 0xFFFF	オペレーション失敗
ResultOverHeated = 0xFFF6	オーバーヒート

ResultModuleOverHeated = 0x0307	モジュールでのオーバーヒート

5.1.2 MemoryBank

定義	説明
Bank_Reserved	Reseved メモリーバンク
Bank_EPC	EPC メモリーバンク
Bank_TID	TID メモリーバンク
Bank_User	User メモリーバンク

5.1.3 BuzzerState

定義	説明
Buzzer_Off	ビープ音オフ
Buzzer_Low	ビープ音小
Buzzer_High	ビープ音大

5.1.4 VibratorState

定義	説明
Vibrator_Off	バイブレーションをオフ
Vibrator_On	バイブレーションをオン

5.1.5 SessionType

定義	説明
Session_S0	inventoried S0
Session_S1	inventoried S1
Session_S2	inventoried S2
Session_S3	inventoried S3

5.1.6 SessionFlag

定義	説明
SessionFlag_A	A only
SessionFlag_B	B only
SessionFlag_AB	A or B

5.1.7 SelectFlag

定義	説明
SelectFlag_NotUsed	フラグを使用しない
SelectFlag_SL	SL を有効にする
SelectFlag_NOT_SL	SL を無効にする
SelectFlag_All	全てを有効にする

5.1.8 MaskTargetType

定義	説明
MaskTarget_S0	inventoried S0
MaskTarget_S1	inventoried S1
MaskTarget_S2	inventoried S2
MaskTarget_S3	inventoried S3
MaskTarget_SL	Selection Flags

5.1.9 MaskActionType

定義	説明
MaskAction_AB	Tag Matching : assert SL or inventoried → A タグがマッチングした場合： ・SL Flag を 1 に設定 ・Inventoried Flag はステータス A に設定される Tag Not-Matching : retract SL or inventoried → B タグがマッチングされない場合： ・SL Flag を 0 に設定 ・Inventoried Flag はステータス B に設定される
MaskAction_AN	Tag Matching : assert SL or inventoried → A タグがマッチングした場合： ・SL Flag を 1 に設定 ・Inventoried Flag はステータス A に設定される Tag Not-Matching：何もしない
MaskAction_NB	Tag Matching：何もしない Tag Not-Matching : retract SL or inventoried → B タグがマッチングされない場合： ・SL Flag を 0 に設定 ・Inventoried Flag はステータス B に設定される
MaskAction_MN	Tag Matching : negate SL or (A → B, B → A) タグがマッチングされない場合： ・SL Flag を反転

	・ Inventoried Flag のステータスは A の場合、B に設定され、ステータスは B の場合、A に設定されます。 Tag Not-Matching： 何もしない
MaskAction_BA	Tag Matching : retract SL or inventoried → B タグがマッチングした場合： ・SL Flag を 0 に設定 ・Inventoried Flag はステータス B に設定される Tag Not-Matching : assert SL or inventoried → A タグがマッチングされない場合： ・SL Flag を 1 に設定 ・Inventoried Flag はステータス A に設定される
MaskAction_BN	Tag Matching : retract SL or inventoried → B タグがマッチングした場合： ・SL Flag を 0 に設定 ・Inventoried Flag はステータス B に設定される Tag Not-Matching： 何もしない
MaskAction_NA	Tag Matching： 何もしない Tag Not-Matching : assert SL or inventoried → A タグがマッチングされない場合： ・SL Flag を 1 に設定 ・Inventoried Flag はステータス A に設定されます。
MaskAction_NM	Tag Matching： 何もしない Tag Not-Matching : negate SL or (A → B, B → A) タグがマッチングされない場合： ・SL Flag を反転 ・ Inventoried Flag のステータスは A の場合、B に設定され、ステータスは B の場合、A に設定されます。

5.1.10 MaskType

定義	説明
MaskType_NO_MASK	No MASK.
MaskType_Selection	Selection MASK.
MaskType_EPC	EPC MASK.

5.1.11 TagState

定義	説明
State_Unknown	対象タグの EPC 情報の取得エラー
State_Error	その他のエラー
State_NoReply	Lock ステータスの取得処理のエラー
State_UnLock	RF タグの EPC エリアはロックされてない
State_Lock	RF タグの EPC エリアはロックされている
State_PermaLock	RF タグの EPC エリアは永久ロックされている

5.2 Struct

5.2.1 CMinMaxValue

RFID の Output Power 設定範囲

定義	型	説明
min	int	設定できる最小パワー
max	int	設定できる最大パワー

5.3 LockParam

5.3.1 Properties

5.3.1.1 @property (nonatomic) BOOL killPassword;

説明：
Kill パスワードの制御可否の取得・設定。
YES：制御
NO：制御しない

5.3.1.2 @property (nonatomic) BOOL accessPassword;

説明：
Access パスワードの制御可否の取得・設定。
YES：制御
NO：制御しない

5.3.1.3 @property (nonatomic) BOOL epc;

説明：
EPC エリアの制御可否の取得・設定。
YES：制御
NO：制御しない

5.3.1.4 @property (nonatomic) BOOL tid;

説明：
TID エリアの制御可否の取得・設定。
YES：制御
NO：制御しない

5.3.1.5 @property (nonatomic) BOOL user;

説明：
User エリアの制御可否の取得・設定。
YES：制御
NO：制御しない

5.4 AsResultCode

5.4.1 Function

5.4.1.1 msg

関数名	+(NSString *)msg:(ResultCode)code;		
引数名	IN/OUT	型	説明
code	IN	ResultCode	メソッドの呼び出し結果 列挙型 ResultCode（5.1.1 をご参照ください）
-	OUT	NSString	ResultCode に紐づく説明文
■メソッドの説明： ResultCode から対応した文字列を返します。 ■サンプルコード： NSString * errorMessage = [AsResultCode msg:ResultNoError];			

5.5 AsSelectMaskParam

5.5.1 Properties

5.5.1.1 @property (nonatomic) MaskTargetType target;

説明：
Mask の MaskTargetType を取得・設定します。
列挙型 MaskTargetType（5.1.8 をご参照ください）

5.5.1.2 @property (nonatomic) MaskActionType action;

説明：
Mask の MaskActionType を取得・設定します。
列挙型 MaskActionType（5.1.9 をご参照ください）

5.5.1.3 @property (nonatomic) MemoryBank bank;

説明：
Mask の MemoryBank を取得・設定します。
列挙型 MemoryBank（5.1.2 をご参照ください）

5.5.1.4 @property (nonatomic) int offset;

説明：
Mask のオフセットを取得・設定します。

5.5.1.5 @property (strong, nonatomic) NSString *mask;

説明：
Mask を取得・設定します（16 進数）。

5.5.1.6 @property (nonatomic) int length;

説明：
Mask の長さを取得・設定します。

5.5.1.7 @property (nonatomic) BOOL used;

説明：
Mask の使用可否を取得・設定します。
YES：Mask を使用
NO：Mask を使用しない

5.5.2 Function

5.5.2.1 index

関数名	- (int)index;		
引数名	IN/OUT	型	説明
-	OUT	int	インデックス
■メソッドの説明： AsSelectMaskParam オブジェクトのマスクのインデックスを取得します。 ■サンプルコード： int index = [param index];			

5.5.2.2 initWithIndex

関数名	- (id)initWithIndex:(int)index;		
引数名	IN/OUT	型	説明
index	IN	int	マスク順序
-	OUT	AsSelectMaskParam	AsSelectMaskParam オブジェクト
■メソッドの説明： マスク順序で AsSelectMaskParam オブジェクトを作成します。 ■サンプルコード： AsSelectMaskParam *param = [[AsSelectMaskParam alloc]initWithIndex:0];			

5.5.2.3 initWithParameterIndex

関数名	**- (id)initWithParameterIndex:(int)index target:(MaskTargetType)maskTarget action:(MaskActionType)maskAction bank:(MemoryBank)maskBank offset:(int)maskOffset mask:(NSString *)maskData used:(BOOL)usedMask;**		
引数名	**IN/OUT**	**型**	**説明**
index	IN	int	マスク順序（0~7）
maskTarget	IN	MaskTargetType	Selection Mask の Target 値 列挙型 MaskTargetType（5.1.8 をご参照ください）
maskAction	IN	MaskActionType	Selection Mask の Action 値 列挙型 MaskActionType（5.1.9 をご参照ください）
maskBank	IN	MemoryBank	Selection Mask の MemoryBank 値 列挙型 MemoryBank（5.1.2 をご参照ください）
maskOffset	IN	int	スタート位置
maskData	IN	NSString	Mask データ（16 進数）
usedMask	IN	BOOL	Mask の使用可否

■メソッドの説明：
Selection マスク情報を構成して AsSelectMaskParam オブジェクトを作成します。
■サンプルコード：

```
AsSelectMaskParam *param = [[AsSelectMaskParam alloc]
initWithParameterIndex:0 target:MaskTarget_SL action:MaskAction_AB
bank:Bank_EPC offset:32 mask:@"1234"used:YES];
```

5.5.2.4 initWithParameterLength

関数名	**- (id)initWithParameterLength:(int)index target:(MaskTargetType)maskTarget action:(MaskActionType)maskAction bank:(MemoryBank)maskBank offset:(int)maskOffset mask:(NSString *)maskData length:(int)maskLength used:(BOOL)usedMask;**		
引数名	**IN/OUT**	型	説明
index	IN	int	マスク順序（0~7）
maskTarget	IN	MaskTargetType	Selection Mask の Target 値 列挙型 MaskTargetType（5.1.8 をご参照ください）
maskAction	IN	MaskActionType	Selection Mask の Action 値 列挙型 MaskActionType（5.1.9 をご参照ください）
maskBank	IN	MemoryBank	Selection Mask の MemoryBank 値 列挙型 MemoryBank（5.1.2 をご参照ください）
maskOffset	IN	int	スタート位置
maskData	IN	NSString	Mask データ（16 進数）
maskLength	IN	int	Mask の長さ（単位：bit）
usedMask	IN	BOOL	Mask の使用可否

■メソッドの説明：

Selection マスク情報を構成して AsSelectMaskParam オブジェクトを作成します。

■サンプルコード：

```
AsSelectMaskParam *param = [[AsSelectMaskParam alloc]
initWithParameterIndex:0 target:MaskTarget_SL action:MaskAction_AB
bank:Bank_EPC offset:32 mask:@"1234" length:16 used:YES];
```

5.6 AsSelectMaskEPCParam

5.6.1 Properties

5.6.1.1 @property (nonatomic) int offset;

説明：
Selection マスクデータのスタート位置を取得・設定します。

5.6.1.2 @property (nonatomic) int length;

説明：
ターゲットタグのデータ長さを取得・設定します。

5.6.1.3 @property (strong, nonatomic) NSString *mask;

説明：
ターゲットタグのマスクデータを取得・設定します（16 進数）。

5.7 LbtItem

5.7.1 Properties

5.7.1.1 @property (nonatomic) int mSlot;

説明：
LBT 周波数テーブルの周波数位置を取得・設定します。

5.7.1.2 @property (nonatomic) BOOL mIsUsed;

説明：
LBT 周波数テーブルの特定周波数の使用可否を取得・設定します。
YES：使用
NO：使用しない

5.7.1.3 @property (strong, nonatomic) NSString *frequency;

説明：
LBT 周波数テーブルの周波数を取得・設定します。

5.7.2 Function

5.7.2.1 init

関数名	- (id)init;		
引数名	IN/OUT	型	説明
-	OUT	LbtItem	LbtItem オブジェクト
■メソッドの説明： LbtItem オブジェクトを初期化します。 ■サンプルコード： LbtItem *lbtI = [[LbtItem alloc] init];			

5.7.2.2 initWithSlot

関数名	- (id)initWithSlot:(int)slot isUsed:(BOOL)isUsed;		
引数名	IN/OUT	型	説明
slot	IN	int	LBT 周波数テーブルの周波数位置
isUsed	IN	BOOL	LBT 周波数テーブルの指定スロット周波数の使用可否 YES：使用 NO：使用しない
-	OUT	id	LbtItem オブジェクト
■メソッドの説明： LbtItem を初期化・作成します。 ■サンプルコード： LbtItem *lbtI = [LbtItem alloc] initWithSlot:1 isUsed:YES];			

6 AsPacket Class

6.1 Enum

6.1.1 CommandType

定義	説明
CommandInventory = 0x66	インベントリ中
CommandReadMemory = 0x72	メモリ読取
CommandWriteMemory = 0x77	メモリ書込
CommandKill = 0x6B	タグキル
CommandLock = 0x6C	タグロック
CommandUnlock = 0x6D	タグアンロック
CommandPermaLock = 0x70	タグ永久ロック
CommandBlockWrite = 0x57	メモリブロック書込
CommandBlockErase = 0x45	メモリブロック消去
CommandStop = 0x73	オプレーション停止
CommandDefaultParam = 0x61	デフォルトパラメータ復帰
CommandSaveParam = 0x53	パラメーター保存
CommandDecodeStart = 0x64	バーコードデコード開始
CommandBuzzerStart = 0x75	ブザー開始
CommandVibratorStart = 0x76	バイブレーションを開始中

6.1.2 ScanMode

定義	説明
RFIDScanMode	RFID モード
BarcodeScanMode	Barcode モード

7 AsBarcodeType Class

7.1 Function

7.1.1 getBarcodeString

関数名	+(NSString *)getBarcodeString:(BarcodeType)barcodeType;		
引数名	**IN/OUT**	型	説明
barcodeType	IN	BarcodeType	列挙型 BarcodeType（7.2.1 をご参照ください）
-	OUT	NSString	バーコードタイプ
■メソッドの説明： 指定した BarcodeType（7.2.1 をご参照）列挙型からバーコードタイプを文字列で取得できます。 ■サンプルコード： NSString * barcodeName = [AsBarcodeType getBarcodeString: BarcodeTypeEAN13];			

7.2 Enum

7.2.1 BarcodeType

読み取ったバーコードタイプの ENUM 定義

列挙子名	説明
BarcodeTypeNoRead	NR (バーコードを一定時間読み込まれない場合に返ってきます。)
BarcodeTypeAustralianPost	Australian Post
BarcodeTypeAztecCode	Aztec Code
BarcodeTypeBooklandEAN	Bookland EAN
BarcodeTypeBritishPost	British Post
BarcodeTypeCanadianPost	Canadian Post
BarcodeTypeChinaPost	China Post
BarcodeTypeCodabar	Codabar
BarcodeTypeCodablockF	Codablock F
BarcodeTypeCode11	Code 11
BarcodeTypeCode128	Code 128
BarcodeTypeCode16K	Code 16K
BarcodeTypeCode32	Code 32
BarcodeTypeCode39	Code 39
BarcodeTypeCode49	Code 49
BarcodeTypeCode93	Code 93
BarcodeTypeComposite	EAN-UCC Composite Code
BarcodeTypeD2of5	Discreate 2 of 5
BarcodeTypeDataMatrix	Data Matrix
BarcodeTypeEAN128	UCC/EAN-128
BarcodeTypeEAN13	EAN-13
BarcodeTypeEAN13CouponCode	EAN-13 with Extended Coupon Code
BarcodeTypeEAN8	EAN-8
BarcodeTypeI2of5	Interleaved 2 of 5
BarcodeTypeIATA	IATA 2 of 5
BarcodeTypeISBT128	ISBT 128
BarcodeTypeISBT128Concat	ISBT-128 Concat.
BarcodeTypeJapanesePost	Japanese Post

BarcodeTypeKixPost	Kix (Netherlands) Post
BarcodeTypeKoreaPost	Korea Post
BarcodeTypeMacroMicroPDF	Macro Micro PDF
BarcodeTypeMaxiCode	MaxiCode
BarcodeTypeMicroPDF	Micro PDF 417
BarcodeTypeMSI	MSI
BarcodeTypeParameterFNC3	Parameter (FNC3)
BarcodeTypePDF417	PDF-417
BarcodeTypePlanetCode	Planet Code
BarcodeTypePlesseyCode	Plessey Code
BarcodeTypePostnet	Postnet
BarcodeTypeQRCode	QR Code
BarcodeTypeR2of5	Straight 2 of 5
BarcodeTypeRSS	RSS
BarcodeTypeScanletWebcode	Scanlet Webcode
BarcodeTypeTelepen	Telepen
BarcodeTypeTLC39	TCIF Linked Code 39
BarcodeTypeTriopticCode	Trioptic Code 39
BarcodeTypeUPCA	UPC-A
BarcodeTypeUPCE	UPC-E
BarcodeTypeVeriCode	VeriCode
BarcodeTypeX2of5	Matrix 2 of 5
BarcodeTypeRSSLimited	DataBar
BarcodeTypeChineseSensible	Chinese-Sensible Code

8 AsParamName Class

8.1 Function

8.1.1 getName

関数名	+(NSString *)getName:(ParamName)paramName;		
引数名	IN/OUT	型	説明
paramName	IN	ParamName	列挙型 ParamName（8.2.1 をご参照ください）
-	OUT	NSString	バーコードタイプ
■メソッドの説明： 引数で設定したパラメータからバーコードタイプが文字列で取得できます。 ■サンプルコード： NSString * barcodeName = [AsParamName getName:EAN13];			

8.2 Enum

8.2.1 ParamName

設定できるバーコードタイプの Enum

定義	説明
Codabar	Codabar
Code39	Code 39
Code32 Pharmaceutical	Code32 Pharmaceutical
I2of5	Interleaved 2 of 5
NEC2of5	NEC 2 of 5
Code93	Code 93
R2of5	Straight 2 of 5 Industrial
A2of5	Straight 2 of 5 IATA
X2of5	Matrix 2 of 5
Code11	Code 11
Code128	Code 128
GS1128	GS1-128
Telepen	Telepen
UPCA	UPC-A
UPCACouponCode	UPC-A/EAN-13 with Extended Coupon Code
CouponGS1DataBarOutput	Coupon GS1 DataBar Output
UPCE0	UPC-E0
UPCE1	UPC-E1
EAN13	EAN/JAP-13
EAN8	EAN/JAP-8
MSI	MSI
RSS14	RSS-14
RSSLimit	RSS Limited
RSSExp	RSS Expanded
TriopticCode	Trioptic Code
CodablockA	Codablock A
CodablockF	Codablock F
PDF417	PDF 417
MacroPDF417	MacroPDF417
MicroPDF	MicroPDF 417

ComCode	EAN/UCC Composite Code
GS1Emulation	GS1 Emulation
TLC39	TCIF Linked Code 39
ChinaPost	China Post
KoreaPost	Korea Post
QRCode	QR Code
Matrix	Data Matrix
MaxiCode	MaxiCode
AztecCode	Aztec Code
HanXinCode	Chinese Sensible (Han Xin) Code
PostalCodes	2D Postal Codes
Code39Pharmaceutical	Code 39 Pharmaceutical

9 AsParamValue Class

9.1 Properties

9.1.1 @property (assign, readwrite) ParamName paramName;

説明：
AsReaderGUN でスキャンできるバーコードタイプを取得・設定します。
列挙型 ParamName（8.2.1 をご参照ください）

9.1.2 @property (assign, readwrite) unsigned int value;

説明：
ParamName で設定されたバーコードのスキャンを有効か、無効かを設定・取得します。
1：スキャンできる
0：スキャンできない

9.2 Function

9.2.1 setEnabled

関数名	- (void)setEnabled:(BOOL)value;		
引数名	**IN/OUT**	型	説明
value	IN	BOOL	バーコードタイプのスキャンの有効可否を設定 YES：有効 NO　：無効
■メソッドの説明： ParamName で設定されたバーコードのスキャンを有効か、無効かを設定します。 ■サンプルコード： AsParamValue * paramValue = [asReader getBarcodeParam: EAN13]; [paramValue setEnabled:NO];			

10 AsMaskActionType Class

10.1 Function

10.1.1 toString

関数名	+(NSString*)toString:(MaskActionType)actionType targetType:(MaskTargetType)targetType;		
引数名	**IN/OUT**	型	説明
actionType	IN	MaskActionType	Mask アクションタイプ 列挙型 MaskActionType（5.1.9 をご参照ください）
targetType	IN	MaskTargetType	Mask タグタイプ 列挙型 MaskTargetType（5.1.8 をご参照ください）
-	OUT	NSString	Mask アクションタイプとタグタイプの説明
■メソッドの説明： MaskActionType と MaskTargetType によって、説明文を文字列で取得します。 ■サンプルコード： NSString * actionMessage = [AsMaskActionType toString:MaskAction_AB targetType:MaskTarget_S0];			

11 附録

11.1 パラメータの保存位置及びデフォルト値

※本附録は ASR-L251G 専用です。

プロパティ	デフォルト値	L251G 本体に保存	SDK のインスタンス内に保存	L251G 本体を再起動すると、デフォルト値に戻る
buzzer	ON (HIGH)	√		
vibrator	ON	√		
operationTime	0 (ms):Not use time		√	
inventoryTime	400(ms)			√
idleTime	300(ms)			√
sleepTime	65535(s)	√		
autoOffTime	65535(s)	√		
accessPassword	"0"			√
inventorySession	0:S0			√
sessionFlag	2:A or B			√
serialNumber		√		
continuousMode	YES			√
powerGain	300(30.0 dBm)			√
isUseKeyAction				
useSelectionMask	No		√	
rssiMode	No		√	
epcMaskMatchMode	No		√	
algorithm	1:Dynmaic			√
minQ	0			√
maxQ	15			√
qValue	4			√
linkProfileValue	1			√
defaultLinkProfileValue				
maskTypeValue	0:No MASK		√	