

ASR-M24D SDK

SDK リファレンスガイド V 1.0

対象 OS	Android
対象言語	Java
バージョン	4.5.6

No.	バージョン	修正内容	日付
1	1.0	新規作成	2024/9/30

目 次

概要.....	7
1 開発環境の構築.....	8
1.1 SDK の追加.....	8
1.2 SDK の使用.....	11
2 メソッドの説明.....	13
2.1 AsReader.....	13
2.1.1 getInstance	13
2.1.2 getDeviceManager	13
2.1.3 getBarcodeManager()	13
2.1.4 getVersion()	14
2.1.5 initialize(Context context, ConnectionType connectionType).....	14
2.1.6 setLogLevel	14
2.1.7 getLogLevel.....	15
2.1.8 setLog	15
2.1.9 removeSDKLog().....	15
2.1.10 enableAutoLaunchApp.....	16
2.2 BarcodeManager	17
2.2.1 getInstance	17
2.2.2 getCharacterStatus.....	17
2.2.3 setCallback	17
2.2.4 setDecodeMode.....	18
2.2.5 getDecodeMode	18
2.2.6 setEncodeMode	18
2.2.7 getEncodeMode.....	19
2.2.8 initialize	19
2.2.9 startDecode.....	19
2.2.10 startDecode	20
2.2.11 stopDecode	20
2.2.12 factoryDefault	20
2.2.13 getStopCondition	20
2.2.14 setBarcodeSettings	21
2.2.15 getBarcodeSettings.....	21
2.2.16 setSymbologyAllEnable.....	22
2.2.17 setSymbologySettings	22
2.2.18 setSymbologySettings	23
2.2.19 setOCRSettings	24
2.2.20 setOCRSettings	24

2.2.21	getSymbologySettings	25
2.2.22	getSymbologySettings	25
2.2.23	getOCRSettings	25
2.2.24	getOCRSettings	26
2.2.25	getSystemBeep	26
2.2.26	setNotisEditing	27
2.2.27	getNotisEditing	27
2.2.28	setPresentationMode	28
2.2.29	getPresentationMode.....	28
2.3	BarcodeManagerCallback	29
2.3.1	onReceivedBarcodeDecodeData.....	29
2.3.2	onReceivedBarcodeDecodeData.....	30
2.3.3	onReceivedResponse	30
2.3.4	onBarcodeManagerStandByReady.....	31
2.3.5	onBarcodeManagerError.....	31
2.4	BarcodeManagerError	32
2.4.1	BarcodeManagerError	32
2.5	DeviceManager.....	33
2.5.1	getInstance	33
2.5.2	setPackageName	33
2.5.3	setCallback	33
2.5.4	isConnected.....	34
2.5.5	setTriggerMode	34
2.5.6	getTriggerMode.....	34
2.5.7	setBackgroundRead	35
2.5.8	getBackgroundRead.....	35
2.5.9	initialize	35
2.5.10	isOpen	36
2.5.11	open	36
2.5.12	close.....	36
2.5.13	setReaderSettings.....	37
2.5.14	getReaderSettings	37
2.5.15	getReaderInfo	38
2.5.16	getSleepTime	38
2.5.17	setSleepTime	38
2.5.18	getSleepBeep	39
2.5.19	setSleepBeep	39
2.5.20	getSleepTimeAndBeep.....	39
2.5.21	setSleepTimeAndBeep.....	40
2.5.22	setHIDMode.....	40

2.5.23 getHIDMode.....	40
2.6 DeviceManagerCallback.....	41
2.6.1 onConnect.....	41
2.6.2 onBatteryStateReceived	41
2.6.3 onTriggerEventReceived	42
2.6.4 onReaderInfoReceived	42
2.6.5 onSetReaderSettingSuccess	43
2.6.6 onReaderSettingReceived	43
2.6.7 onSleepSettingReceived	44
2.6.8 onHIDModeReceived	44
2.6.9 onDeviceManagerError	45
2.7 BarcodeConst	46
2.7.1 BarcodeConst.BarcodeSettings	46
2.7.2 BarcodeConst. QREncodeMode	46
2.7.3 BarcodeConst. ScanDataFormatType	46
2.7.4 BarcodeConst. MemoryType	47
2.7.5 BarcodeConst. SystemBeepSoundType.....	47
2.7.6 BarcodeConst. BarcodeCommandAction	48
2.8 BarcodeResult	49
2.8.1 BarcodeResult.Status	49
2.8.2 BarcodeConst. Result	49
2.8.3 BarcodeResult.....	50
2.8.4 BarcodeResult.....	50
2.8.5 getResult.....	51
2.8.6 setResult	51
2.8.7 getMessage	51
2.8.8 setMessage.....	52
2.8.9 getSymbologyInfoArray.....	52
2.8.10 setSymbologyInfoArray	53
2.8.11 getOCRInfoArray	53
2.8.12 setOCRInfoArray.....	54
2.9 DeviceConst	55
2.9.1 DeviceConst. DeviceTriggerEvent	55
2.9.2 DeviceConst.TriggerMode	55
2.9.3 DeviceConst. DeviceCommandType	56
2.9.4 DeviceConst.HIDType	56
2.10 OCR.....	57
2.10.1 OCR.....	57

2.11 OCRInfoModel	57
2.11.1 OCRInfoModel	57
2.12 Symbology	58
2.12.1 Symbology.....	58
2.13 SymbologyInfoModel	60
2.13.1 SymbologyInfoModel.....	60
2.14 LogLevel.....	61
2.14.1 LogLevel.....	61
2.14.2 getCode	61
2.14.3 getName	61
2.15 ConnectionType.....	62
2.15.1 ConnectionType.....	62
2.16 AsReaderError.....	62
2.16.1 AsReaderError.ErrorCode.....	62

概要

本文の主な目的：

- 開発者が A24D SDK ライブラリ※を使用して Android アプリケーションを開発できるように、開発環境の構築方法と SDK ライブラリの説明を記載しています。

開発ツール：

- Android Studio Giraffe | 2022.3.1
- Android SDK10.0
- Android Gradle 7.2

システム要求：

- Android 10+

※ASR-A24D と ASR-M24D の SDK は互換性があります。

ASR-M24D をご利用の際も A24D SDK ライブラリ (A24D-SDK.aar) をご利用ください。

1 開発環境の構築

1.1 SDK の追加

1. 「app」->「libs」->「右クリック」->「Open in Finder」を選択します。（図 1-1-1）

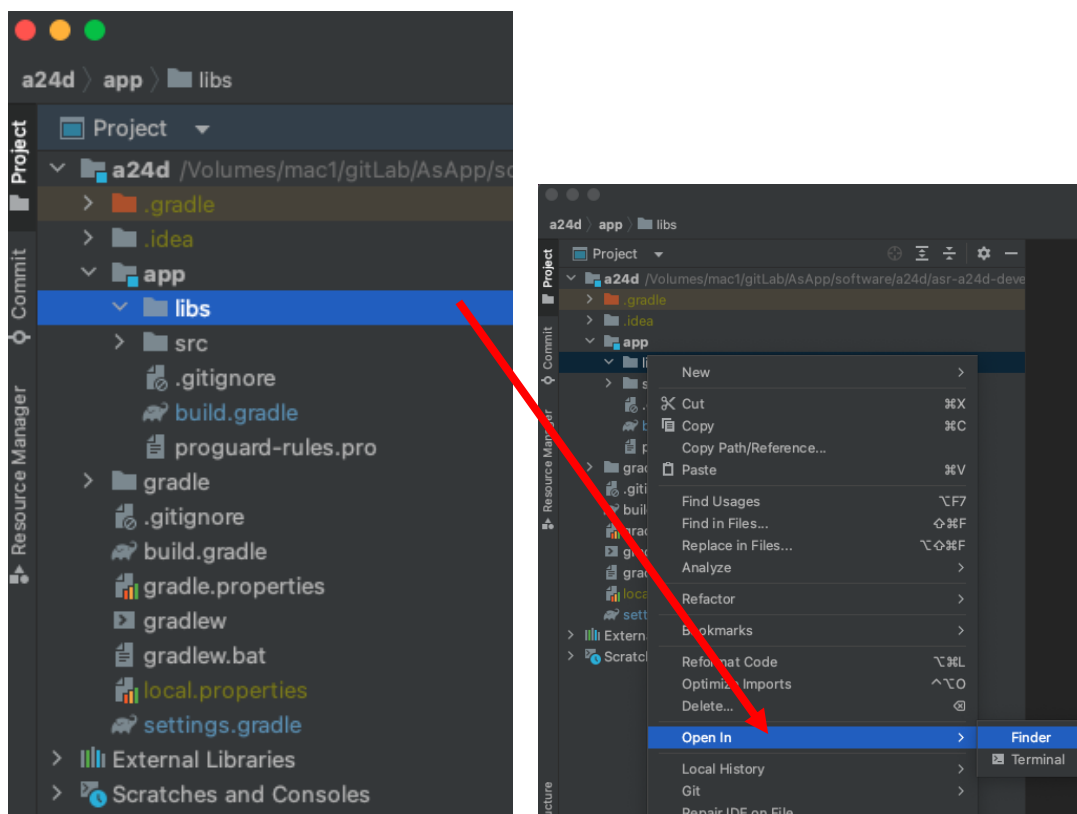


図 1-1-1

2. ポップアップしたウィンドウの「libs」を選択して、このフォルダーに「A24D-SDK.aar」を貼り付けます（図 1-1-2）。「A24D-SDK.aar」はプロジェクトの「libs」に表示されます（図 1-1-3）。

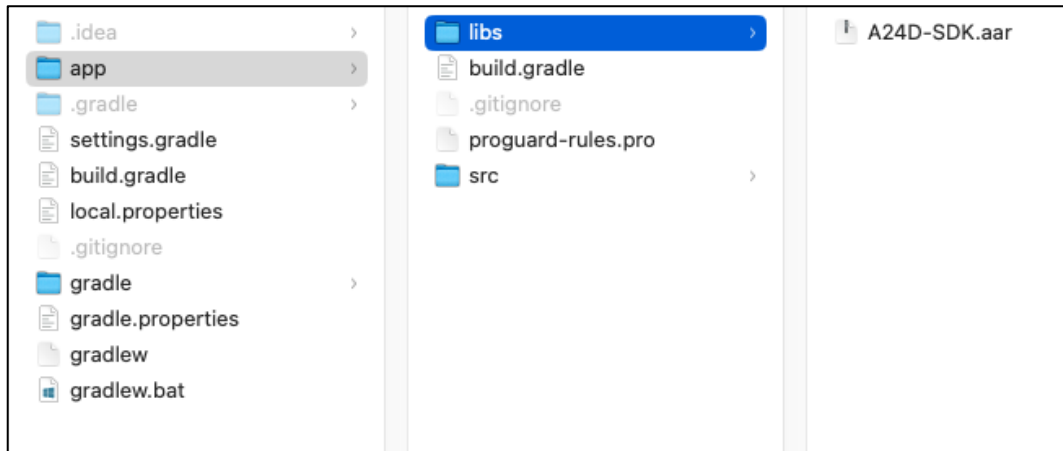


図 1-1-2

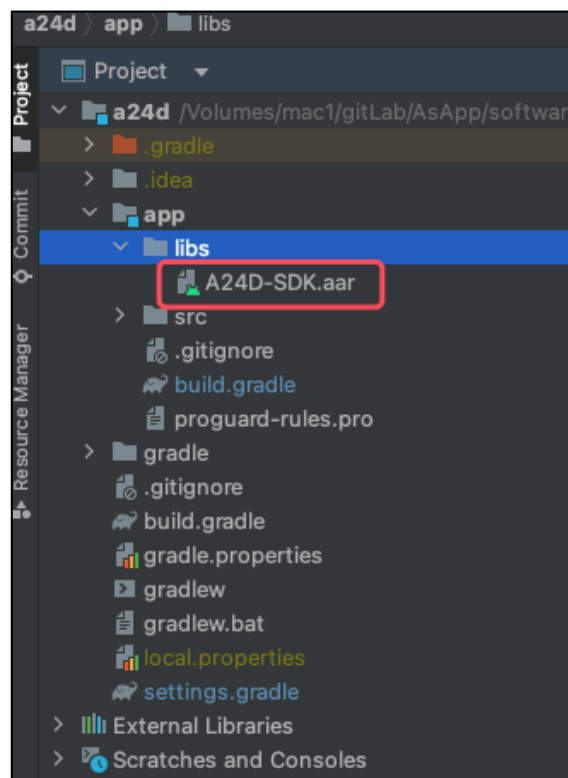


図 1-1-3

3. 「build.gradle」をダブルクリックします。（図 1-1-4）

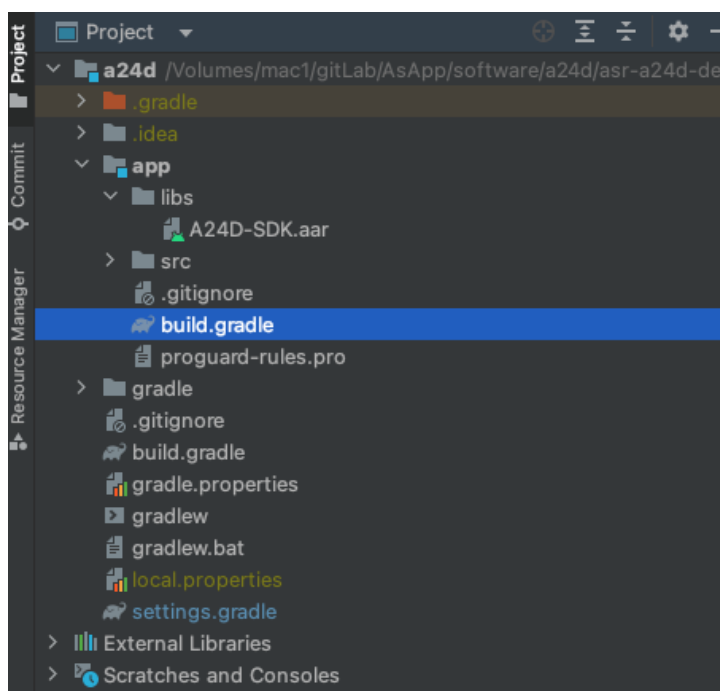


図 1-1-4

4. 下図 1 の通り、依存関係にライブラリをインポートします。
次に「2」の「Sync Now」をクリックし、同期します。

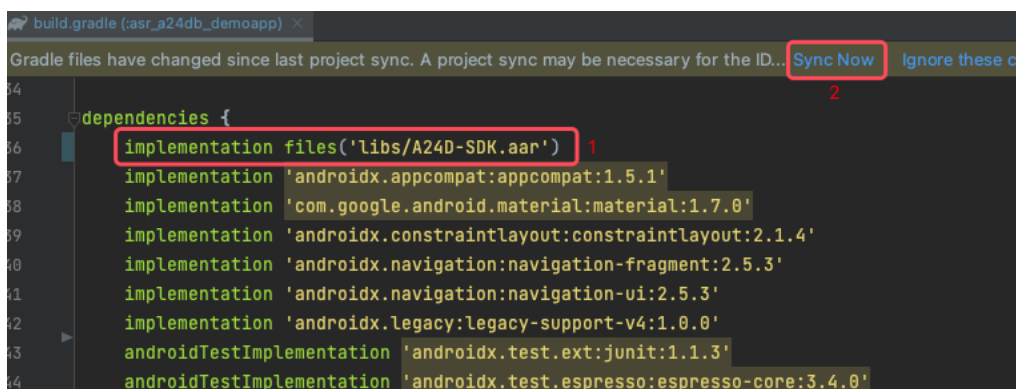


図 1-1-5

5. 同期に成功すると以下のように表示されます（図 1-1-6）。
これで SDK の追加が完了です。

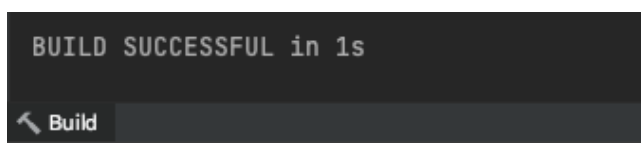


図 1-1-6

1.2 SDK の使用

「import」文でライブラリーを使用するクラスにライブラリーを引用します。（図 1-2-1）

```
import jp.co.asterisk.asreader.a24d.sdk.AsReader;
```

図 1-2-1

オブジェクト「AsReader」を作成して初期化します（図 1-2-2）。

```
AsReader.getInstance().initialize(getApplicationContext(), ConnectionType.USB);
```

図 1-2-2

「DeviceManagerCallback」（図 1-2-3）を実現して、接続成功してから他のコマンドを実行できるようになります。

```
AsReader.getInstance().getDeviceManager().setCallback(new DeviceManagerCallback() {  
    @Override  
    public void onConnect(boolean isConnect) {  
  
    }  
  
    @Override  
    public void onBatteryStateReceived(int battery) {  
  
    }  
  
    @Override  
    public void onTriggerEventReceived(DeviceConst.DeviceTriggerEvent event) {  
  
    }  
  
    @Override  
    public void onDeviceManagerError(AsReaderError.ErrorCode code) {  
  
    }  
  
});
```

図 1-2-3

「BarcodeManagerCallback」(図 1-2-4) を実現して、バーコード解析結果をコールバックします。

```
AsReader.getInstance().getBarcodeManager().setCallback(new BarcodeManagerCallback() {  
    @Override  
    public void onReceivedBarcodeDecodeData(String decodeData, Map<String, String> parameter) {  
  
    }  
  
    @Override  
    public void onBarcodeManagerError(BarcodeManagerError error) {  
  
    }  
});
```

図 1-2-4

バーコード解析開始 (図 1-2-5)

```
if (AsReader.getInstance().getDeviceManager().isConnected() ) {  
    AsReader.getInstance().getBarcodeManager().startDecode();  
}
```

図 1-2-5

2 メソッドの説明

2.1 AsReader

2.1.1 getInstance

関数名	public static AsReader getInstance()		
引数名	IN/OUT	型	説明
	OUT	AsReader	シングルオブジェクト AsReader
<p>■メソッドの説明：</p> <p>シングルトンでオブジェクト AsReader を作成します。</p> <p>■サンプルコード：</p> <pre>AsReader asReader = AsReader.getInstance();</pre>			

2.1.2 getDeviceManager

関数名	public DeviceManager getDeviceManager()		
引数名	IN/OUT	型	説明
	OUT	DeviceManager	DeviceManager のシングルオブジェクトを返す (2.5 を参照)
<p>■メソッドの説明：</p> <p>オブジェクト DeviceManager を取得します。</p> <p>■サンプルコード：</p> <pre>DeviceManager deviceManager = AsReader.getInstance().getDeviceManager();</pre>			

2.1.3 getBarcodeManager()

関数名	public BarcodeManager getBarcodeManager()		
引数名	IN/OUT	型	説明
	OUT	BarcodeManager	BarcodeManager のシングルオブジェクトを返す (2.2 を参照)
<p>■メソッドの説明：</p> <p>BarcodeManager オブジェクトを取得します。</p> <p>■サンプルコード：</p> <pre>BarcodeManager barcodeManager = AsReader.getInstance().getBarcodeManager();</pre>			

2.1.4 getVersion()

関数名	public String getVersion()		
引数名	IN/OUT	型	説明
	OUT	String	SDK バージョン
<p>■メソッドの説明：</p> <p>SDK バージョンを取得します。</p> <p>■サンプルコード：</p> <pre>String version = AsReader.getInstance().getVersion();</pre>			

2.1.5 initialize(Context context, ConnectionType connectionType)

関数名	public void initialize(Context context, ConnectionType connectionType)		
引数名	IN/OUT	型	説明
context	IN	Context	Context
connectType	IN	ConnectionType	ConnectionType (2.15 を参照)
<p>■メソッドの説明：</p> <p>Context と接続タイプによって初期化設定を行います。</p> <p>■サンプルコード：</p> <pre>AsReader.getInstance().initialize(getApplicationContext(), ConnectionType.USB);</pre>			

2.1.6 setLogLevel

関数名	public void setLogLevel(LogLevel level)		
引数名	IN/OUT	型	説明
level	IN	LogLevel	LogLevel (2.14 を参照)
<p>■メソッドの説明：</p> <p>ログレベルを設定します。</p> <p>■サンプルコード：</p> <pre>AsReader.getInstance().setLogLevel(LogLevel.Debug);</pre>			

2.1.7 getLogLevel

関数名	public LogLevel getLogLevel()		
引数名	IN/OUT	型	説明
	OUT	LogLevel	LogLevel (2.14 を参照)
<p>■ メソッドの説明 :</p> <p>ログレベルを取得します。</p> <p>■ サンプルコード :</p> <pre>LogLevel level= AsReader.getInstance().getLogLevel();</pre>			

2.1.8 setLog

関数名	public void setLog(boolean enable)		
引数名	IN/OUT	型	説明
enable	IN	boolean	ログの出力要否 true : 出力 false : 出力しない
<p>■ メソッドの説明 :</p> <p>ログの出力要否を設定します。</p> <p>■ サンプルコード :</p> <pre>AsReader.getInstance().setLog(true);</pre>			

2.1.9 removeSDKLog()

関数名	public void removeSDKLog()		
<p>■ メソッドの説明 :</p> <p>すべてのログファイルを削除します。</p> <p>■ サンプルコード :</p> <pre>AsReader.getInstance().removeSDKLog();</pre>			

2.1.10 enableAutoLaunchApp

関数名	public void enableAutoLaunchApp(boolean enable, String aliasName)		
引数名	IN/OUT	型	説明
enable	IN	boolean	M24D と接続時にデモアプリを自動起動するかどうかの設定 true : 自動起動 false : 自動起動しない
aliasName	IN	String	Activity フラッグ
<p>■メソッドの説明 :</p> <p>M24D と接続時に、アプリを自動起動するかどうかを設定します。</p> <p>■サンプルコード :</p> <pre>AsReader.getInstance().enableAutoLaunchApp(false, "MainActivityAlias");</pre>			

2.2 BarcodeManager

2.2.1 getInstance

関数名	public static BarcodeManager getInstance()		
引数名	IN/OUT	型	説明
	OUT	BarcodeManager	シングルオブジェクト BarcodeManager (2.2 を参照)
<p>■メソッドの説明 :</p> <p>シングルトンでオブジェクト BarcodeManager を作成します。</p> <p>■サンプルコード :</p> <pre>BarcodeManager barcodeManager = BarcodeManager.getInstance();</pre>			

2.2.2 getCharacterStatus

関数名	public BarcodeConst.BarcodeSettings getCharacterStatus()		
引数名	IN/OUT	型	説明
	OUT	BarcodeConst.BarcodeSettings	BarcodeConst.BarcodeSettings (2.7.1 を参照)
<p>■メソッドの説明 :</p> <p>デコードステータスを取得します。</p> <p>■サンプルコード :</p> <pre>BarcodeConst.BarcodeSettings setting = BarcodeManager.getInstance().getCharacterStatus();</pre>			

2.2.3 setCallback

関数名	public void setCallback(BarcodeManagerCallback callback)		
引数名	IN/OUT	型	説明
callback	IN	BarcodeManagerCallback	BarcodeManagerCallback (2.3 を参照)
<p>■メソッドの説明 :</p> <p>コールバックを設定します。</p> <p>■サンプルコード :</p> <pre>BarcodeManager.getInstance().setCallback(this);</pre>			

2.2.4 setDecodeMode

関数名	public void setDecodeMode(BarcodeConst.BarcodeSettings mode)		
引数名	IN/OUT	型	説明
callback	IN	BarcodeConst.BarcodeSettings	BarcodeConst.BarcodeSettings (2.7.1を参照)
<p>■ メソッドの説明 :</p> <p>デコードモードを設定します。 BarcodeConst.BarcodeSettings.DECODE_MODE_CONTINUOUS または BarcodeConst.BarcodeSettings.DECODE_MODE_SINGLE</p> <p>■ サンプルコード :</p> <pre>BarcodeManager.getInstance().setDecodeMode(BarcodeConst.BarcodeSettings.DECODE_MODE_SINGLE);</pre>			

2.2.5 getDecodeMode

関数名	public BarcodeConst.BarcodeSettings getDecodeMode()		
引数名	IN/OUT	型	説明
	OUT	BarcodeConst.BarcodeSettings	BarcodeConst.BarcodeSettings (2.7.1を参照)
<p>■ メソッドの説明 :</p> <p>デコードモードを取得します。 BarcodeConst.BarcodeSettings.DECODE_MODE_CONTINUOUS または BarcodeConst.BarcodeSettings.DECODE_MODE_SINGLE</p> <p>■ サンプルコード :</p> <pre>BarcodeConst.BarcodeSettings mode = BarcodeManager.getInstance().getDecodeMode();</pre>			

2.2.6 setEncodeMode

関数名	public void setEncodeMode(BarcodeConst.QREncodeMode mode)		
引数名	IN/OUT	型	説明
mode	IN	BarcodeConst.QREncodeMode	BarcodeConst.QREncodeMode (2.7.2を参照)
<p>■ メソッドの説明 :</p> <p>エンコードモードを設定します。</p> <p>■ サンプルコード :</p> <pre>BarcodeManager.getInstance().setDecodeMode(BarcodeConst.QREncodeMode.AUTOMATIC);</pre>			

2.2.7 getEncodeMode

関数名	public BarcodeConst.QREncodeMode getEncodeMode()		
引数名	IN/OUT	型	説明
	OUT	BarcodeConst.QREncodeMode	BarcodeConst.QREncodeMode (2.7.2 を参照)
<p>■ メソッドの説明 : エンコードモードを取得します。</p> <p>■ サンプルコード : <pre>BarcodeConst.QREncodeMode mode = BarcodeManager.getInstance().getEncodeMode();</pre></p>			

2.2.8 initialize

関数名	public void initialize(Context context)		
引数名	IN/OUT	型	説明
context	IN	Context	Context
<p>■ メソッドの説明 : Context を設定します。 AsReader クラスの初期化時に自動で呼び出されるため、明示的に呼び出す必要はありません。</p>			

2.2.9 startDecode

関数名	public void startDecode()		
<p>■ メソッドの説明 : デコード開始します。 デコード結果は BarcodeManagerCallback クラスの onReceivedBarcodeDecodeData (2.3.1 を参照) でコールバックします。</p> <p>■ サンプルコード : <pre>BarcodeManager.getInstance().startDecode();</pre></p>			

2.2.10 startDecode

関数名	public void startDecode(@IntRange(from = 0, to = 255) int count, @IntRange(from = 0, to = 255) int time)
<p>■ メソッドの説明 : デコードを開始します。 デコード結果は BarcodeManagerCallback クラスの onReceivedBarcodeDecodeData (2.3.1 を参照) でコールバックします。</p> <p>■ サンプルコード : <pre>BarcodeManager.getInstance().startDecode(10,10);</pre></p>	

2.2.11 stopDecode

関数名	public void stopDecode()
<p>■ メソッドの説明 : デコードを停止します。</p> <p>■ サンプルコード : <pre>BarcodeManager.getInstance().stopDecode();</pre></p>	

2.2.12 factoryDefault

関数名	public void factoryDefault()
<p>■ メソッドの説明 : AsReader のモジュールを工場出荷時に戻します。 結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3 を参照) でコールバックします。</p> <p>■ サンプルコード : <pre>BarcodeManager.getInstance().factoryDefault();</pre></p>	

2.2.13 getStopCondition

関数名	public void getStopCondition()
<p>■ メソッドの説明 : 設定した読取停止条件を取得します。 結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3 を参照) でコールバックします。</p> <p>■ サンプルコード : <pre>BarcodeManager.getInstance().getStopCondition();</pre></p>	

2.2.14 setBarcodeSettings

関数名	public void setBarcodeSettings(BarcodeConst.MemoryType memoryType, BarcodeConst.BarcodeSettings settingsType)		
引数名	IN/OUT	型	説明
memoryType	IN	BarcodeConst.MemoryType	BarcodeConst.MemoryType (2.7.4 を参照)
settingsType	IN	BarcodeConst.BarcodeSettings	BarcodeConst.BarcodeSettings (2.7.1 を参照)
<p>■ メソッドの説明 : バーコードの設定項目を設定します。 設定結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3 を参照) でコールバックします。</p> <p>■ サンプルコード : <pre>BarcodeManager.getInstance().setBarcodeSettings(BarcodeConst.MemoryType.MEMORY_TYPE_PERMANENT, BarcodeConst.BarcodeSettings.CHARACTER_NONE);</pre></p>			

2.2.15 getBarcodeSettings

関数名	public void getBarcodeSettings (BarcodeConst.BarcodeSettings settingsType)		
引数名	IN/OUT	型	説明
settingsType	IN	BarcodeConst.BarcodeSettings	BarcodeConst.BarcodeSettings (2.7.1 を参照)
<p>■ メソッドの説明 : バーコードの設定項目を取得します。 結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3 を参照) でコールバックします。</p> <p>■ サンプルコード : <pre>BarcodeManager.getInstance().setBarcodeSettings(BarcodeConst.BarcodeSettings.CHARACTER_CODE_ID_SYMBOL);</pre></p>			

2.2.16 setSymbologyAllEnable

関数名	public void setSymbologyAllEnable(BarcodeConst.MemoryType memoryType, boolean isEnabled)		
引数名	IN/OUT	型	説明
memoryType	IN	BarcodeConst.MemoryType	BarcodeConst.MemoryType (2.7.4 を参照)
isEnabled	IN	boolean	すべてバーコードタイプの読取 On/Off true:On false:Off
<p>■メソッドの説明： すべてのバーコードタイプの読取の On/Off を設定します。 設定結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3 を参照) でコールバックします。</p> <p>■サンプルコード： BarcodeManager.getInstance().setSymbologyAllEnable(BarcodeConst.MemoryType.MEMORY_TYPE_PERMANENT,true);</p>			

2.2.17 setSymbologySettings

関数名	public void setSymbologySettings(BarcodeConst.MemoryType memoryType, Symbology symbology, boolean isEnabled)		
引数名	IN/OUT	型	説明
memoryType	IN	BarcodeConst.MemoryType	BarcodeConst.MemoryType (2.7.4 を参照)
symbology	IN	Symbology	Symbology (2.12 を参照)
isEnabled	IN	boolean	現在のバーコードタイプの読取の On/Off true:On false:Off
<p>■メソッドの説明： 現在のバーコードタイプの読取の On/Off を設定します。 設定結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3 を参照) でコールバックします。</p> <p>■サンプルコード： BarcodeManager.getInstance().setSymbologySettings(BarcodeConst.MemoryType.MEMORY_TYPE_PERMANENT,Symbology.UPC_A,true);</p>			

2.2.18 setSymbologySettings

関数名	public void setSymbologySettings(BarcodeConst.MemoryType memoryType, ArrayList<SymbologyInfoModel> symbologyInfoList)		
引数名	IN/OUT	型	説明
memoryType	IN	BarcodeConst.MemoryType	BarcodeConst.MemoryType (2.7.4を参照)
symbologyInfoList	IN	ArrayList<SymbologyInfoModel>	SymbologyInfoModel 配列 (2.7.13を参照)
<p>■メソッドの説明： バーコードタイプの読取の On/Off を設定します。 設定結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3を参照) でコールバックします。</p> <p>■サンプルコード：</p> <pre>SymbologyInfoModel model1 = new SymbologyInfoModel(Symbology.UPC_A, true); SymbologyInfoModel model2 = new SymbologyInfoModel(Symbology.UPC_E, true); ArrayList<SymbologyInfoModel> infoModelArrayList = new ArrayList<>(); infoModelArrayList.add(model1); infoModelArrayList.add(model2); BarcodeManager.getInstance().setSymbologySettings(BarcodeConst.MemoryType.MEMORY_TYPE_PERMANENT, infoModelArrayList);</pre>			

2.2.19 setOCRSettings

関数名	public void setOCRSettings(BarcodeConst.MemoryType memoryType, OCR ocr, boolean isEnabled)		
引数名	IN/OUT	型	説明
memoryType	IN	BarcodeConst.MemoryType	BarcodeConst.MemoryType (2.7.4 を参照)
ocr	IN	OCR	OCR (2.10 を参照)
isEnabled	IN	boolean	指定したタイプの OCR の読取の On/Off true:On false:Off
<p>■ メソッドの説明 : 指定したタイプの OCR を読取するかどうかを設定します。</p> <p>■ サンプルコード : BarcodeManager.getInstance().setOCRSettings(BarcodeConst.MemoryType.MEMORY_TYPE_PERMANENT,OCR.OCR_A,true);</p>			

2.2.20 setOCRSettings

関数名	public void setOCRSettings(BarcodeConst.MemoryType memoryType, ArrayList<OCRInfoModel> ocrInfoList)		
引数名	IN/OUT	型	説明
memoryType	IN	BarcodeConst.MemoryType	BarcodeConst.MemoryType (2.7.4 を参照)
ocrInfoList	IN	ArrayList<OCRInfoModel>	OCRInfoModel 配列 (2.11 を参照)
<p>■ メソッドの説明 : 指定したタイプの OCR を読取するかどうかを設定します。 設定結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3 を参照) でコールバックします。</p> <p>■ サンプルコード : OCRInfoModel model = new OCRInfoModel(OCR.OCR_A,true); ArrayList<OCRInfoModel> infoModelArrayList = new ArrayList<>(); infoModelArrayList.add(model); BarcodeManager.getInstance().setOCRSettings(BarcodeConst.MemoryType.MEMORY_TYPE_PERMANENT, infoModelArrayList);</p>			

2.2.21 getSymbologySettings

関数名	public void getSymbologySettings(Symbology symbology)		
引数名	IN/OUT	型	説明
symbology	IN	Symbology	Symbology (2.12を参照)
<p>■メソッドの説明：</p> <p>現在のバーコードの設定を取得します。</p> <p>設定結果は BarcodeManagerCallbasck (2.3.3を参照) クラスの onReceivedResponse でコールバックします。</p> <p>■サンプルコード：</p> <pre>BarcodeManager.getInstance().getSymbologySettings(Symbology.UPC_A);</pre>			

2.2.22 getSymbologySettings

関数名	public void getSymbologySettings(ArrayList<Symbology> symbologyList)		
引数名	IN/OUT	型	説明
symbology	IN	ArrayList<Symbology>	Symbology 配列 (2.12を参照)
<p>■メソッドの説明：</p> <p>バーコードの設定を取得します。</p> <p>設定結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3を参照) でコールバックします。</p> <p>■サンプルコード：</p> <pre>ArrayList<Symbology> symbolArrayList = new ArrayList<>(); symbolArrayList.add(Symbology.UPC_A); BarcodeManager.getInstance().getSymbologySettings(symbolArrayList);</pre>			

2.2.23 getOCRSettings

関数名	public void getOCRSettings(OCR ocr)		
引数名	IN/OUT	型	説明
ocr	IN	OCR	OCR (2.10を参照)
<p>■メソッドの説明：</p> <p>ocr の設定を取得します。</p> <p>設定結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3を参照) でコールバックします。</p> <p>■サンプルコード：</p> <pre>BarcodeManager.getInstance().getOCRSettings(OCR.OCR_A);</pre>			

2.2.24 getOCRSettings

関数名	public void getOCRSettings(ArrayList<OCR> ocrList)		
引数名	IN/OUT	型	説明
ocrList	IN	ArrayList<OCR>	OCR 配列 (2.10 を参照)
<p>■ メソッドの説明 : ocr の設定を取得します。 設定結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3 を参照) でコールバックします。</p> <p>■ サンプルコード : <pre>ArrayList<OCR> ocrArrayList = new ArrayList<>(); ocrArrayList.add(OCR.OCR_A); BarcodeManager.getInstance().getOCRSettings(ocrArrayList);</pre></p>			

2.2.25 getSystemBeep

関数名	public BarcodeConst.SystemBeepSoundType getSystemBeep()		
引数名	IN/OUT	型	説明
	OUT	BarcodeConst.SystemBeepSoundType	BarcodeConst.SystemBeepSoundType (2.7.5 を参照)
<p>■ メソッドの説明 : ビープ音タイプを取得します。</p> <p>■ サンプルコード : <pre>BarcodeConst.SystemBeepSoundType soundType = BarcodeManager.getInstance().getSystemBeep();</pre></p>			

2.2.26 setNotisEditing

関数名	public void setNotisEditing(BarcodeConst.MemoryType memoryType, boolean isEnabled)		
引数名	IN/OUT	型	説明
memoryType	IN	BarcodeConst.MemoryType	BarcodeConst.MemoryType (2.7.4を参照)
isEnabled	IN	boolean	Notis 編集の On / Off true:On false:Off
<p>■ メソッドの説明 : Notis 編集の On / Off を設定します。</p> <p>■ サンプルコード : BarcodeManager.getInstance().setNotisEditing(BarcodeConst.MemoryType.MEMORY_TYPE_PERMANENT,true);</p>			

2.2.27 getNotisEditing

関数名	public void getNotisEditing()
<p>■ メソッドの説明 : Notis 編集の設定を取得します。 結果は BarcodeManagerCallback クラスの onReceivedResponse (2.3.3を参照) でコールバックします。</p> <p>■ サンプルコード : BarcodeManager.getInstance().getNotisEditing();</p>	

2.2.28 setPresentationMode

関数名	public void setPresentationMode(BarcodeConst.MemoryType memoryType, BarcodeConst.BarcodeSettings settingsType)		
引数名	IN/OUT	型	説明
memoryType	IN	BarcodeConst.MemoryType	BarcodeConst.MemoryType (2.7.4 を参照)
settingsType	IN	BarcodeConst.BarcodeSettings	BarcodeConst.BarcodeSettings (2.7.1 を参照)
<p>■ メソッドの説明 : trigger キーモードを設定します。(standard あるいは presentation) TRIGGER_PRESENTATION:trigger モードオン TRIGGER_STANDARD:trigger モードオフ</p> <p>プレゼンテーションモードは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。</p> <p>■ サンプルコード : BarcodeManager.getInstance().setPresentationMode (BarcodeConst.MemoryType.MEMORY_TYPE_PERMANENT, BarcodeConst.BarcodeSettings.TRIGGER_PRESENTATION);</p>			

2.2.29 getPresentationMode

関数名	public void getPresentationMode()
<p>■ メソッドの説明 : trigger キーモードを取得します。</p> <p>■ サンプルコード : BarcodeManager.getInstance().getPresentationMode();</p>	

2.3 BarcodeManagerCallback

2.3.1 onReceivedBarcodeDecodeData

関数名	default void onReceivedBarcodeDecodeData(String decodeData, Map<String, String> parameter)		
引数名	IN/OUT	型	説明
decodeData	OUT	String	バーコードデータ
parameter	OUT	Map<String, String>	バーコードの解析結果
<p>■メソッドの説明： バーコードの解析結果をコールバックします。「2.2.9 startDecode」と「2.2.10 startDecode」を呼び出した後、バーコードを読み取ると、このメソッドをコールバックします。</p> <p>■サンプルコード：</p> <pre>BarcodeManager.getInstance().setCallback(new BarcodeManagerCallback() { @Override public void onReceivedBarcodeDecodeData(String decodeData, Map<String, String> parameter) { // decodeData: Decoded data of the barcode that was scanned. // parameter: Map object containing parameters If no parameters are stored, null is returned KEY_PARAMETER_CODE_ID = Code ID BarcodeConst (2.7を 参照) } });</pre>			

2.3.2 onReceivedBarcodeDecodeData

関数名	default void onReceivedBarcodeDecodeData(byte[] decodeData)		
引数名	IN/OUT	型	説明
decodeData	OUT	byte[]	バーコードデータ
<p>■ メソッドの説明 :</p> <p>バーコードの解析結果をコールバックします。「2.2.9 startDecode」と「2.2.10 startDecode」を呼び出した後、バーコードを読み取ると、このメソッドをコールバックします。</p> <p>■ サンプルコード :</p> <pre>BarcodeManager.getInstance().setCallback(new BarcodeManagerCallback() { @Override public void onReceivedBarcodeDecodeData(byte[] decodeData) { // decodeData: Decoded data of the barcode that was scanned. } });</pre>			

2.3.3 onReceivedResponse

関数名	default void onReceivedResponse(BarcodeConst.BarcodeCommandAction type, BarcodeResult result)		
引数名	IN/OUT	型	説明
type	OUT	BarcodeConst.BarcodeCommandAction	BarcodeConst.BarcodeCommandAction (2.7.6 を参照)
result	OUT	BarcodeResult	BarcodeResult (2.8 を参照)
<p>■ メソッドの説明 :</p> <p>コマンドの実行結果をコールバックします。BarcodeConst.BarcodeCommandAction 関連のコマンドを実行後に、このメソッドをコールバックします。</p> <p>■ サンプルコード :</p> <pre>BarcodeManager.getInstance().setCallback(new BarcodeManagerCallback() { @Override public void onReceivedResponse(BarcodeConst.BarcodeCommandAction type, BarcodeResult result) { // type: BarcodeConst.BarcodeCommandAction // result: BarcodeResult } });</pre>			

2.3.4 onBarcodeManagerStandByReady

関数名	default void onBarcodeManagerStandByReady()
<p>■ メソッドの説明 : AsReader 情報をコールバックします。 AsReader に接続して、端末側 AsReader 情報を取得完了すると、このメソッドをコールバックします。</p> <p>■ サンプルコード :</p> <pre>BarcodeManager.getInstance().setCallback(new BarcodeManagerCallback() { @Override public void onBarcodeManagerStandByReady() { } });</pre>	

2.3.5 onBarcodeManagerError

関数名	void onBarcodeManagerError(BarcodeManagerError error)		
引数名	IN/OUT	型	説明
error	OUT	BarcodeManagerError	BarcodeManagerError (2.4 を参照)
<p>■ メソッドの説明 : エラーをコールバックします。</p> <p>■ サンプルコード :</p> <pre>BarcodeManager.getInstance().setCallback(new BarcodeManagerCallback() { @Override public void onBarcodeManagerError(BarcodeManagerError error) { // error: Error messages. } });</pre>			

2.4 BarcodeManagerError

2.4.1 BarcodeManagerError

定義	説明
COMMAND_ERROR_COMMAND_SEND_FAILED	データパッケージ送信失敗
COMMAND_ERROR_START_DECODE_FAILED	デコード起動失敗
SETTINGS_ERROR_SETTINGS_FAILED	バーコード設定失敗
SETTINGS_ERROR_TIMEOUT	バーコード設定はタイムアウト
COMMAND_ERROR_GET_SYSTEM_BEEP	システムビープの設定を取得できない

2.5 DeviceManager

2.5.1 getInstance

関数名	public static DeviceManager getInstance()		
引数名	IN/OUT	型	説明
	OUT	DeviceManager	シングルオブジェクト
<p>■メソッドの説明：</p> <p>シングルトンで DeviceManager オブジェクトを作成します。</p> <p>■サンプルコード：</p> <pre>DeviceManager deviceManager = DeviceManager.getInstance();</pre>			

2.5.2 setPackageName

関数名	public void setPackageName(String name)		
引数名	IN/OUT	型	説明
name	IN	String	パッケージ名
<p>■メソッドの説明：</p> <p>パッケージ名を設定します。</p> <p>■サンプルコード：</p> <pre>DeviceManager.getInstance().setPackageName(getPackageName())</pre>			

2.5.3 setCallback

関数名	public void setCallback(DeviceManagerCallback callback)		
引数名	IN/OUT	型	説明
callback	IN	DeviceManagerCallback	DeviceManagerCallback (2.6を参照)
<p>■メソッドの説明：</p> <p>コールバックを設定します。</p> <p>■サンプルコード：</p> <pre>DeviceManager.getInstance().setCallback(this);</pre>			

2.5.4 isConnected

関数名	public boolean isConnected()		
引数名	IN/OUT	型	説明
	OUT	boolean	接続ステータス
<p>■メソッドの説明： 接続ステータスを取得します。</p> <p>■サンプルコード： boolean connected = DeviceManager.getInstance().isConnected();</p>			

2.5.5 setTriggerMode

関数名	public void setTriggerMode(DeviceConst.TriggerMode mode)		
引数名	IN/OUT	型	説明
mode	IN	DeviceConst.TriggerMode	DeviceConst.TriggerMode (2.9.2を参照)
<p>■メソッドの説明： トリガーモードを設定します。</p> <p>■サンプルコード： DeviceManager.getInstance().setTriggerMode(DeviceConst.TriggerMode.AUTO);</p>			

2.5.6 getTriggerMode

関数名	public DeviceConst.TriggerMode getTriggerMode()		
引数名	IN/OUT	型	説明
	OUT	DeviceConst.TriggerMode	DeviceConst.TriggerMode (2.9.2を参照)
<p>■メソッドの説明： トリガーモードを取得します。</p> <p>■サンプルコード： DeviceConst.TriggerMode mode = DeviceManager.getInstance().getTriggerMode();</p>			

2.5.7 setBackgroundRead

関数名	public void setBackgroundRead(boolean enable)		
引数名	IN/OUT	型	説明
enable	IN	boolean	バックグラウンドで読取するかどうかの On/Off true:On false:Off
<p>■ メソッドの説明 : バックグラウンドで読取するかどうかの On/Off を設定します。</p> <p>■ サンプルコード : DeviceManager.getInstance().setBackgroundRead(true);</p>			

2.5.8 getBackgroundRead

関数名	public boolean getBackgroundRead()		
引数名	IN/OUT	型	説明
	OUT	boolean	バックグラウンドで読取
<p>■ メソッドの説明 : バックグラウンドで読取するかどうかを取得します。</p> <p>■ サンプルコード : boolean canRead = DeviceManager.getInstance().getBackgroundRead();</p>			

2.5.9 initialize

関数名	public void initialize(Context context, ConnectionType selectConnection)		
引数名	IN/OUT	型	説明
context	IN	Context	Context
selectConnection	IN	ConnectionType	ConnectionType(2.15 を参照)
<p>■ メソッドの説明 : AsReader クラスの初期化時に自動で呼び出されるため、明示的に呼び出す必要はありません。</p>			

2.5.10 isOpen

関数名	public boolean isOpen()		
引数名	IN/OUT	型	説明
	OUT	boolean	接続セッションステータス true: 接続セッション On false: 接続セッション Off
<p>■ メソッドの説明 : 接続セッションステータスを取得します。</p> <p>■ サンプルコード : <code>boolean isOpen = DeviceManager.getInstance().isOpen();</code></p>			

2.5.11 open

関数名	public void open()
<p>■ メソッドの説明 : 接続セッションをオープンします。</p> <p>■ サンプルコード : <code>DeviceManager.getInstance().open();</code></p>	

2.5.12 close

関数名	public void close()
<p>■ メソッドの説明 : 接続セッションを Off します。</p> <p>■ サンプルコード : <code>DeviceManager.getInstance().close();</code></p>	

2.5.13 setReaderSettings

関数名	public void setReaderSettings(boolean beep, boolean vib, boolean aimer, boolean led, boolean powerOnBeep)		
引数名	IN/OUT	型	説明
beep	IN	boolean	ビープ true:On false:Off
vib	IN	boolean	振動 true:On false:Off
aimer	IN	boolean	エイミング true:On false:Off
led	IN	boolean	ライト true:On false:Off
powerOn Beep	IN	boolean	電源 On ビープ true:On false:Off
<p>■ メソッドの説明 : AsReader を設定します。</p> <p>■ サンプルコード : <code>DeviceManager.getInstance().setReaderSettings(true,true,true,true,true);</code></p>			

2.5.14 getReaderSettings

関数名	public void getReaderSettings()
<p>■ メソッドの説明 : AsReader の設定項目を取得します。 結果は BarcodeManagerCallback クラスの onReaderSettingReceived (2.6.6 を参照) でコールバックします。</p> <p>■ サンプルコード : <code>DeviceManager.getInstance().getReaderSettings();</code></p>	

2.5.15 getReaderInfo

関数名	public void getReaderInfo()
<p>■ メソッドの説明 : AsReader 情報を取得します。 結果は BarcodeManagerCallback クラスの onReaderInfoReceived (2.6.4 を参照) でコールバックします。</p> <p>■ サンプルコード : DeviceManager.getInstance().getReaderInfo();</p>	

2.5.16 getSleepTime

関数名	public void getSleepTime()
<p>■ メソッドの説明 : スリープタイムを取得します。 結果は BarcodeManagerCallback クラスの onSleepSettingReceived (2.6.7 を参照) でコールバックします。</p> <p>■ サンプルコード : DeviceManager.getInstance().getSleepTime();</p>	

2.5.17 setSleepTime

関数名	public void setSleepTime(@IntRange(from = 0, to = 255)int sleepTime)		
引数名	IN/OUT	型	説明
sleepTime	IN	int	スリープタイム 設定範囲 : 1~255 (単位 : 分) 、 0 : スリープなし
<p>■ メソッドの説明 : スリープタイムを設定します。</p> <p>■ サンプルコード : DeviceManager.getInstance().setSleepTime(60);</p>			

2.5.18 getSleepBeep

関数名	public void getSleepBeep()
<p>■ メソッドの説明 : スリープ機能を使用するかどうかを取得します。 結果は BarcodeManagerCallback クラスの onSleepSettingReceived (2.6.7 を参照) でコールバックします。</p> <p>■ サンプルコード : DeviceManager.getInstance().getSleepBeep();</p>	

2.5.19 setSleepBeep

関数名	public void setSleepBeep(boolean isSleepBeepOn)		
引数名	IN/OUT	型	説明
isSleepBeepOn	IN	boolean	スリープ機能の On/Off true:On false:Off
<p>■ メソッドの説明 : スリープ機能を使用するかどうかを設定します。</p> <p>■ サンプルコード : DeviceManager.getInstance().setSleepBeep(true);</p>			

2.5.20 getSleepTimeAndBeep

関数名	public void getSleepTimeAndBeep()
<p>■ メソッドの説明 : スリープタイム及びスリープ機能の設定を取得します。 結果は BarcodeManagerCallback クラスの onSleepSettingReceived (2.6.7 を参照) でコールバックします。</p> <p>■ サンプルコード : DeviceManager.getInstance().getSleepTimeAndBeep();</p>	

2.5.21 setSleepTimeAndBeep

関数名	public void setSleepTimeAndBeep(@IntRange(from = 0, to = 255)int sleepTime, boolean isSleepBeepOn)		
引数名	IN/OUT	型	説明
sleepTime	IN	int	スリープタイム
isSleepBeepOn	IN	boolean	スリープ機能の On/Off true:On false:Off
<p>■ メソッドの説明 : スリープタイム及びスリープ機能を使用するかどうかを設定します。</p> <p>■ サンプルコード : DeviceManager.getInstance().setSleepTimeAndBeep(60,true);</p>			

2.5.22 setHIDMode

関数名	public void setHIDMode(boolean enable)		
引数名	IN/OUT	型	説明
enable	IN	boolean	HID モードの設定 true:HID mode false:SDK mode
<p>■ メソッドの説明 : HID モードを使用するかどうかを設定します。</p> <p>■ サンプルコード : DeviceManager.getInstance().setHIDMode(false);</p>			

2.5.23 getHIDMode

関数名	public void getHIDMode()		
引数名	IN/OUT	型	説明
enable	IN	boolean	HID モード
<p>■ メソッドの説明 : HID モードは使用するかどうかを取得します。 結果は BarcodeManagerCallback クラスの onHIDModeReceived (2.6.8 を参照) でコールバックします。</p> <p>■ サンプルコード : DeviceManager.getInstance().setHIDMode(false);</p>			

2.6 DeviceManagerCallback

2.6.1 onConnect

関数名	void onConnect(boolean isConnect)		
引数名	IN/OUT	型	説明
isConnect	OUT	boolean	接続結果 true:接続済み false:接続を切断
<p>■メソッドの説明： 接続結果をコールバックします。</p> <p>■サンプルコード： <pre>DeviceManager.getInstance().setCallback(new DeviceManagerCallback() { @Override public void onConnect(boolean isConnect) { // isConnect: true: Device is connected./false: Device is disconnected. } });</pre> </p>			

2.6.2 onBatteryStateReceived

関数名	void onBatteryStateReceived(final int battery)		
引数名	IN/OUT	型	説明
battery	OUT	int	バッテリー残量 (%形式)
<p>■メソッドの説明： バッテリー残量をコールバックします。</p> <p>■サンプルコード： <pre>DeviceManager.getInstance().setCallback(new DeviceManagerCallback() { @Override public void onBatteryStateReceived(final int battery){ // battery : Battery level(value of %.) } });</pre> </p>			

2.6.3 onTriggerEventReceived

関数名	void onTriggerEventReceived(DeviceConst.DeviceTriggerEvent event)		
引数名	IN/OUT	型	説明
event	OUT	DeviceConst.DeviceTriggerEvent	DeviceConst.DeviceTriggerEvent (2.9.1 を参照)
<p>■ メソッドの説明 : ハード側トリガーキーの押下イベントをコールバックします。</p> <p>■ サンプルコード :</p> <pre>DeviceManager.getInstance().setCallback(new DeviceManagerCallback() { @Override public void onTriggerEventReceived(DeviceConst.DeviceTriggerEvent event) { // event: Trigger action event. } });</pre>			

2.6.4 onReaderInfoReceived

関数名	default void onReaderInfoReceived(Map<String, String> info)		
引数名	IN/OUT	型	説明
info	OUT	Map<String, String>	AsReader 情報
<p>■ メソッドの説明 : AsReader の接続情報をコールバックします。getReaderInfo (2.5.16 を参照) の呼び出しに成功すると、このメソッドをコールバックします。</p> <p>■ サンプルコード :</p> <pre>DeviceManager.getInstance().setCallback(new DeviceManagerCallback() { @Override public void onReaderInfoReceived(Map<String, String> info){ // info: Map object that set each acquired value. DeviceConst(2.9 を参照) } });</pre>			

2.6.5 onSetReaderSettingSuccess

関数名	default void onSetReaderSettingSuccess()
<p>■ メソッドの説明 : 接続した AsReader の情報をコールバックします。 setReaderSettings (2.5.14 を参照) の呼び出しに成功すると、このメソッドをコールバックします。</p>	
<p>■ サンプルコード :</p> <pre>DeviceManager.getInstance().setCallback(new DeviceManagerCallback() { @Override public void onSetReaderSettingSuccess() { } });</pre>	

2.6.6 onReaderSettingReceived

関数名	default void onReaderSettingReceived(Map<String, Boolean> settingParamMap)		
引数名	IN/OUT	型	説明
settingParamMap	OUT	Map<String, Boolean>	AsReader の設定情報
<p>■ メソッドの説明 : 接続中の AsReader の設定情報をコールバックします。 getReaderSettings (2.5.15 を参照) の呼び出しに成功すると、このメソッドをコールバックします。</p>			
<p>■ サンプルコード :</p> <pre>DeviceManager.getInstance().setCallback(new DeviceManagerCallback() { @Override public void onReaderSettingReceived(Map<String, Boolean> settingParamMap){ // settingParamMap: Map object that has each acquired value. DeviceConst (2.9 を参照) } });</pre>			

2.6.7 onSleepSettingReceived

関数名	default void onSleepSettingReceived(Map<String, String> sleepParamMap)		
引数名	IN/OUT	型	説明
sleepParamMap	OUT	Map<String, String>	AsReader のスリープ情報
<p>■ メソッドの説明 : AsReader のスリープ情報をコールバックします。 スリープ関連のコマンドの呼び出しに成功すると、このメソッドをコールバックします。</p> <p>■ サンプルコード :</p> <pre>DeviceManager.getInstance().setCallback(new DeviceManagerCallback() { @Override public void onSleepSettingReceived(Map<String, String> sleepParamMap) { // sleepParamMap: Map object that has each acquired value. DeviceConst // (2.9 を参照) } });</pre>			

2.6.8 onHIDModeReceived

関数名	default void onHIDModeReceived(DeviceConst.HIDType hidType)		
引数名	IN/OUT	型	説明
hidType	OUT	DeviceConst.HIDType	DeviceConst.HIDType(2.9.4 を参照)
<p>■ メソッドの説明 : HID モードの情報をコールバックします。 getHIDMode (2.5.24 を参照) の呼び出しに成功すると、このメソッドをコールバックします。</p> <p>■ サンプルコード :</p> <pre>DeviceManager.getInstance().setCallback(new DeviceManagerCallback() { @Override public void onHIDModeReceived(DeviceConst.HIDType hidType){ // hidType: HID_ON is on, HID_OFF is HID off, NOT_APPLICABLE is when // device HID is not used. } });</pre>			

2.6.9 onDeviceManagerError

関数名	void onDeviceManagerError(AsReaderError.ErrorCode code)		
引数名	IN/OUT	型	説明
code	OUT	AsReaderError.ErrorCode	AsReaderError.ErrorCode (2.16.1 を参照)
<p>■ メソッドの説明 :</p> <p>AsReader のエラー情報をコールバックします。</p> <p>■ サンプルコード :</p> <pre>DeviceManager.getInstance().setCallback(new DeviceManagerCallback() { @Override public void onDeviceManagerError(AsReaderError.ErrorCode code){ // code: Error code. } });</pre>			

2.7 BarcodeConst

2.7.1 BarcodeConst.BarcodeSettings

定義	説明
DECODE_MODE_SINGLE	シングルスキャン
DECODE_MODE_CONTINUOUS	連続スキャン
NOTIS_EDITING	デコードされた Codabar シンボルから開始文字と停止文字を除去または追加する。
CHARACTER_CODE_ID_SYMBOL	Symbol CodeID をバーコードデータに追加する
CHARACTER_CODE_ID_AIM	AIM CodeID をバーコードデータに追加する
CHARACTER_NONE	バーコードデータにキャラクターを追加しない
CHARACTER_GET	キャラクターを取得する
PREFIX_STATUS	バーコードデータにキャラクターを追加しない

2.7.2 BarcodeConst. QREncodeMode

定義	説明
AUTO	自動エンコード
SHIFT_JIS	SHIFT_JIS エンコード
EUC_KR	EUC_KR エンコード
UTF_8	UTF_8 エンコード
UTF_16	UTF_16 エンコード
UTF_32	UTF_32 エンコード

2.7.3 BarcodeConst. ScanDataFormatType

定義	説明
UNKNOWN	未知
ONLY_DATA	BC
DATA_SUFFIX_TYPE1	BC + SUFFIX1
DATA_SUFFIX_TYPE2	BC + SUFFIX2
DATA_SUFFIX_TYPE3	BC + SUFFIX1 + SUFFIX2
PREFIX_DATA	PREFIX + BC
PREFIX_DATA_SUFFIX_TYPE1	PREFIX + BC + SUFFIX1
PREFIX_DATA_SUFFIX_TYPE2	PREFIX + BC + SUFFIX2
PREFIX_DATA_SUFFIX_TYPE3	PREFIX + BC + SUFFIX1 + SUFFIX2

2.7.4 BarcodeConst. MemoryType

定義	説明
MEMORY_TYPE_TEMPORARY	一時保存
MEMORY_TYPE_PERMANENT	永久保存

2.7.5 BarcodeConst. SystemBeepSoundType

定義	説明
NONE	ビープなし
TYPE1	短くて高い音
TYPE2	短く低い音
TYPE3	長く高い音
TYPE4	長く低い音
TYPE5	速いワープ音
TYPE6	遅いワープ音
TYPE7	高いクリック音
TYPE8	低いクリック音

2.7.6 BarcodeConst. BarcodeCommandAction

定義	説明
UNSET	設定していない
GET_INFO_STOP_CONDITION	読取停止条件を取得する
GET_INFO_CHARACTER_STATUS	キャラクターステータスを取得する
GET_INFO_SYBBOLOGIES_ENABLE_STATUS	バーコードの可用状態を取得する
GET_INFO_OCR_ENABLE_STATUS	OCRの可用情報を取得する
GET_INFO_SCAN_DATA_TRANSMISSION_FORMAT	バーコードデータ伝送時のフォーマットを取得する
GET_INFO_SYSTEM_BEEP	システムビープを取得する
GET_INFO_CODE_ID_ENABLE	CODE IDの可用情報を取得する
GET_INFO_NOTIS_EDITING	NOTIS編集の設定を取得する
SETTINGS_CHARACTER_STATUS	キャラクターのステータスを設定する
SETTINGS_SYBBOLOGIES	バーコードタイプを設定する
SETTING_OCR	OCRを設定する
SETTINGS_SYSTEM_BEEP	システムビープを設定する
SETTINGS_NOTIS_EDITING	NOTIS編集を設定する
FACTORY_DEFAULT	工場出荷時の設定に戻す
FINISH	完了

2.8 BarcodeResult

2.8.1 BarcodeResult.Status

定義	説明
SUCCESS	成功
DAILED	失敗

2.8.2 BarcodeConst. Result

定義	説明
UNKNOWN	未知な結果
SYSTEM_BEEP_TYPE1	短く高い音
SYSTEM_BEEP_TYPE2	短く低い音
SYSTEM_BEEP_TYPE3	長く高い音
SYSTEM_BEEP_TYPE4	長く低い音
SYSTEM_BEEP_TYPE5	速いワープ音
SYSTEM_BEEP_TYPE6	遅いワープ音
SYSTEM_BEEP_TYPE7	高いクリック音
SYSTEM_BEEP_TYPE8	低いクリック音
SYSTEM_BEEP_SOUND_NONE	ビーブなし
FACTORY_DEFAULT	工場出荷時の設定に戻す
CHARACTER_NONE	キャラクターなし
CHARACTER_SYMBOL	Symbol CodeID キャラクター
CHARACTER_AIM	Aim CodeID キャラクター
STOP_CONDITIONS_CONTINUOUS_ON	連続読取 On
STOP_CONDITIONS_CONTINUOUS_OFF	連続読取 Off
SYMBOLGY_STATUS_GET	バーコードのステータスを取得する
SYMBOLGY_STATUS_SET	バーコードのステータスを設定する
OCR_STATUS_GET	OCR のステータスを取得する
OCR_STATUS_SET	OCR のステータスを設定する
FORMAT_DATA_ONLY	生データ
FORMAT_DATA_SUFFIX_TYPE1	BC + SUFFIX1
FORMAT_DATA_SUFFIX_TYPE2	BC + SUFFIX2
FORMAT_DATA_SUFFIX_TYPE3	BC + SUFFIX1 + SUFFIX2

FORMAT_PREFIX_DATA	PREFIX + BC
FORMAT_PREFIX_DATA_SUFFIX_TYPE1	PREFIX + BC + SUFFIX1
FORMAT_PREFIX_DATA_SUFFIX_TYPE2	PREFIX + BC + SUFFIX2
FORMAT_PREFIX_DATA_SUFFIX_TYPE3	PREFIX + BC + SUFFIX1 + SUFFIX2
NOTIS_EDITING_GET_ENABLED	NOTIS 編集 On
NOTIS_EDITING_GET_DISABLED	NOTIS 編集 Off

2.8.3 BarcodeResult

関数名	public BarcodeResult()
<p>■ メソッドの説明 : オブジェクト BarcodeResult を作成します。</p> <p>■ サンプルコード : <pre>BarcodeResult result= new BarcodeResult();</pre></p>	

2.8.4 BarcodeResult

関数名	public BarcodeResult(Status status, Result result, String message, ArrayList<SymbologyInfoModel> array)		
引数名	IN/OUT	型	説明
status	IN	Status	Status (2.8.1 を参照)
result	IN	Result	Result (2.8.2 を参照)
message	IN	String	メッセージ
array	IN	ArrayList<SymbologyInfoModel>	SymbologyInfoModel 配列 (2.13 を参照)
<p>■ メソッドの説明 : BarcodeResult オブジェクトを作成します。</p> <p>■ サンプルコード : <pre>BarcodeResult barcodeResult = new BarcodeResult(BarcodeResult.Status.SUCCESS, BarcodeResult.Result.SYMBOLGY_STATUS_SET, "", null);</pre></p>			

2.8.5 getResult

関数名	public Result getResult()		
引数名	IN/OUT	型	説明
	OUT	Result	Result (2.8.2を参照)
<p>■メソッドの説明： オブジェクト BarcodeResult の結果のメッセージを取得します。</p> <p>■サンプルコード： Result result = barcodeResult.getResult(); (barcodeResult は BarcodeResult オブジェクト)</p>			

2.8.6 setResult

関数名	public void setResult(Result result)		
引数名	IN/OUT	型	説明
result	IN	Result	Result (2.8.2を参照)
<p>■メソッドの説明： オブジェクト BarcodeResult の結果のメッセージを設定します。</p> <p>■サンプルコード： Result result = barcodeResult.setResult(Result.SYSTEM_BEEP_TYPE1); (barcodeResult は BarcodeResult オブジェクト)</p>			

2.8.7 getMessage

関数名	public String getMessage()		
引数名	IN/OUT	型	説明
	OUT	String	メッセージ
<p>■メソッドの説明： オブジェクト BarcodeResult のメッセージを取得します。</p> <p>■サンプルコード： String message = barcodeResult.getMessage(); (barcodeResult は BarcodeResult オブジェクト)</p>			

2.8.8 setMessage

関数名	public void setMessage(String msg)		
引数名	IN/OUT	型	説明
msg	IN	String	メッセージ
<p>■ メソッドの説明： オブジェクト BarcodeResult のメッセージを設定します。</p> <p>■ サンプルコード： <pre>barcodeResult.setMessage("message");</pre> (barcodeResult は BarcodeResult オブジェクト)</p>			

2.8.9 getSymbologyInfoArray

関数名	public ArrayList<SymbologyInfoModel> getSymbologyInfoArray()		
引数名	IN/OUT	型	説明
	OUT	ArrayList<SymbologyInfoModel>	SymbologyInfoModel 配列 (2.13を参照)
<p>■ メソッドの説明： バーコードが読み取り可能かどうかを取得します。</p> <p>■ サンプルコード： <pre>ArrayList<SymbologyInfoModel> array = barcodeResult. getSymbologyInfoArray();</pre> (barcodeResult は BarcodeResult オブジェクト)</p>			

2.8.10 setSymbologyInfoArray

関数名	public void setSymbologyInfoArray(ArrayList<SymbologyInfoModel> array)		
引数名	IN/OUT	型	説明
array	IN	ArrayList<SymbologyInfoModel>	SymbologyInfoModel 配列 (2.13 を参照)
<p>■ メソッドの説明 : バーコードが読み取り可能かどうかを設定します。</p> <p>■ サンプルコード : <pre>ArrayList<SymbologyInfoModel> array = barcodeResult. getSymbologyInfoArray(); SymbologyInfoModel model1 = new SymbologyInfoModel(Symbology.UPC_A, true); SymbologyInfoModel model2 = new SymbologyInfoModel(Symbology.UPC_E, true); ArrayList<SymbologyInfoModel> infoModelArrayList = new ArrayList<>(); infoModelArrayList.add(model1); infoModelArrayList.add(model2); barcodeResult.setSymbologySettings(infoModelArrayList); (barcodeResult は BarcodeResult オブジェクト)</pre></p>			

2.8.11 getOCRInfoArray

関数名	public ArrayList<OCRInfoModel> getOCRInfoArray()		
引数名	IN/OUT	型	説明
	OUT	ArrayList<OCRInfoModel>	OCRInfoModel 配列(2.11 を参照)
<p>■ メソッドの説明 : OCR のモード情報 (OCR-A, OCR-B) と、それらの OCR が有効かどうかを取得します。</p> <p>■ サンプルコード : <pre>ArrayList<OCRInfoModel> array = barcodeResult.getOCRInfoArray(); (barcodeResult は BarcodeResult オブジェクト)</pre></p>			

2.8.12 setOCRInfoArray

関数名	public void setOCRInfoArray(ArrayList<OCRInfoModel> array)		
引数名	IN/OUT	型	説明
array	IN	ArrayList<OCRInfoModel>	OCRInfoModel 配列(2.11 を参照)
<p>■メソッドの説明：</p> <p>OCR のモード情報（OCR-A, OCR-B）と、それらの OCR の有効/無効を設定します。</p> <p>■サンプルコード：</p> <pre>OCRInfoModel model = new OCRInfoModel(OCR.OCR_A,true); ArrayList<OCRInfoModel> infoModelArrayList = new ArrayList<>(); infoModelArrayList.add(model); barcodeResult.setOCRInfoArray(infoModelArrayList); (barcodeResult は BarcodeResult オブジェクト)</pre>			

2.9 DeviceConst

2.9.1 DeviceConst. DeviceTriggerEvent

定義	説明
PUSH_NONE	トリガーを押下してない
PUSH_RIGHT	右側のトリガーキーが押下される
PUSH_LEFT	左側のトリガーキーが押下される

2.9.2 DeviceConst.TriggerMode

定義	説明
MANUAL	手動モード
AUTO	自動モード

2.9.3 DeviceConst. DeviceCommandType

定義	説明
SET_READER_SETTING	AsReader の設定項目を設定する
GET_READER_SETTING	AsReader の設定項目を取得する
SET_CHARGING_CONTROL	チャージを設定する
READER_INFO	AsReader 情報
SET_SLEEP_TIME	スリープタイムを設定する
GET_SLEEP_TIME	スリープタイムを取得する
SET_SLEEP_BEEP	スリープ機能の使用可否を設定する
GET_SLEEP_BEEP	スリープ機能の使用可否を取得する
GET_SLEEP_TIME_AND_BEEP	スリープタイムとスリープ機能の使用可否を設定する
SET_SLEEP_TIME_AND_BEEP	スリープタイムとスリープ機能の使用可否を取得する
SET_HID_MODE	HID モードを設定する
GET_HID_MODE	HID モードを取得する

2.9.4 DeviceConst.HIDType

定義	説明
HID_ON	HID オン
HID_OFF	HID オフ
NOT_APPLICABLE	適用しない

2.10 OCR

2.10.1 OCR

定義	説明
OCR_A	OCR A
OCR_B	OCR B

2.11 OCRInfoModel

2.11.1 OCRInfoModel

関数名	public OCRInfoModel(OCR ocr, boolean isEnabled)		
引数名	IN/OUT	型	説明
ocr	IN	OCR	OCR
isEnabled	IN	boolean	エンコードのステータス
<p>■メソッドの説明：</p> <p>OCRInfoModelオブジェクトを作成します。</p> <p>■サンプルコード：</p> <pre>OCRInfoModel model = new OCRInfoModel(OCR.OCR_A,true)</pre>			

2.12 Symbology

2.12.1 Symbology

定義	説明
UPC_A	UPC-A
UPC_E	UPC-E
UPC_E1	UPC-E1
EAN13_JAN13	EAN-13/JAN-13
EAN8_JAN8	EAN-8/JAN-8
BOOK_LAND_EAN	Bookland EAN
ISSN_EAN	ISSN EAN
INTERLEAVED_2_OF_5	Interleaved 2 of 5(ITF)
DISCRETE_2_OF_5	Discrete 2 of 5 (DTF)
CODA_BAR	Codabar(NW-7)
CODE_128	Code128
CODE_39	Code39
CODE_93	Code93
CODE_11	Code11
MSI	MSI
CHINESE_2_OF_5	Chinese 2 of 5
MATRIX_2_OF_5	Matrix 2 of 5
KOREAN_2_OF_5	Korean 3 of 5
GS1_128	GS1-128
GS1_DATA_BAR	GS1 DataBar
PDF_417	PDF417
MACRO_PDF	Macro PDF
DATA_MATRIX	Data Matrix
GS1_DATA_MATRIX	GS1 Data Matrix
QR_CODE	QR-CODE
JAPAN_POSTAL	Japanese Postal
MICRO_QR_CODE	Micro QR Code
AZTEC	Aztec
COMPOSITE_CC_C	Composite CC-C

COMPOSITE_CC_AB	Composite CC-A/B
COMPOSITE_TLC_39	Composite TLC-39
MAXICODE	Maxicode
US_POSTNET	US Postnet
US_PLANET	US Planet
UK_POSTAL	UK Postal
AUSTRALIA_POST	Australia post
NETHERLAND_KIX_CODE	Netherland KIX code

2.13 SymbologyInfoModel

2.13.1 SymbologyInfoModel

関数名	public SymbologyInfoModel(Symbology symbology, boolean isEnabled)		
引数名	IN/OUT	型	説明
symbology	IN	Symbology	バーコードタイプ
isEnabled	IN	boolean	エンコードの On/Off true:On false:Off
<p>■メソッドの説明 :</p> <p>SymbologyInfoModelオブジェクトを作成します。</p> <p>■サンプルコード :</p> <pre>SymbologyInfoModel model = new SymbologyInfoModel(Symbology.UPC_A,true)</pre>			

2.14 LogLevel

2.14.1 LogLevel

定義	説明
Fatal	致命レベル:1
Error	エラーレベル:2
Warning	警告レベル:3
Information	情報レベル:4
Debug	デバッグレベル:5

2.14.2 getCode

関数名	public int getCode()		
引数名	IN/OUT	型	説明
	OUT	int	ログレベル
<p>■ メソッドの説明 :</p> <p>ログレベルを取得します。</p> <p>■ サンプルコード :</p> <pre>int level= logLevel.getCode()</pre>			

2.14.3 getName

関数名	public String getName()		
引数名	IN/OUT	型	説明
	OUT	String	ログレベル名
<p>■ メソッドの説明 :</p> <p>ログレベル名を取得します。</p> <p>■ サンプルコード :</p> <pre>String name = logLevel.getName()</pre>			

2.15 ConnectionType

2.15.1 ConnectionType

定義	説明
USB	USB 接続

2.16 AsReaderError

2.16.1 AsReaderError.ErrorCode

定義	説明
NOT_SUPPORTED	サポートしない
WITHOUT_AUTHORITY	権限なし
MEMORY_OVERRUN	メモリーオーバー
MEMORY_LOCKED	メモリロック
CRYPTO_SUITE_ERROR	暗号化エラー
SECURITY_TIMEOUT	セキュリティタイムアウト
INSUFFICIENT_POWER	バッテリー不足
ALREADY_POWER_ON	パワーオン
FAILED_TO_SEND_PACKET	パッケージ発送失敗