

AsReader[®]

AsReaderP252B SDK

Android SDK 開発マニュアル

修正履歴

バージョン	修正内容	担当者	日付
1.0	新規作成	郭秀敏	2023/05/04

目次

概要	6
1 開発環境の構築	7
1.1 SDK の追加	7
1.2 SDK の使用	10
1.2.1 SDK のインポート	10
1.2.2 「AsReaderP252B」オブジェクトの作成・初期化	10
1.2.3 AsReaderP252BEventListener インターフェースの実装	10
1.2.4 接続	12
1.2.5 データのインベントリ	14
2 メソッド説明	16
2.1 AsReaderP252B	16
2.1.1 getResultCode	16
2.1.2 connectDevice	16
2.1.3 disconnectDevice	17
2.1.4 getCurrentDevice	17
2.1.5 getState	17
2.1.6 getAction	17
2.1.7 getFirmwareVersion	18
2.1.8 getHardwareVersion	18
2.1.9 getRFModuleVersion	18
2.1.10 getPowerGainRange	19
2.1.11 setEventListener	19
2.1.12 removeEventListener	19
2.1.13 inventory	20
2.1.14 inventoryHumidityTag	20
2.1.15 inventoryTemperatureTagWithType	21
2.1.16 readMemory	21
2.1.17 writeMemory	22
2.1.18 lock	23
2.1.19 unlock	23
2.1.20 permaLock	24
2.1.21 kill	24
2.1.22 stop	25
2.1.23 defaultParameter	25
2.1.24 saveParameter	26
2.1.25 getBuzzer	26
2.1.26 setBuzzer	26

2.1.27	getContinuousMode.....	27
2.1.28	setContinuousMode.....	27
2.1.29	getPowerGain.....	28
2.1.30	setPowerGain	28
2.1.31	getPowerGainRange	29
2.1.32	getOperationTime	29
2.1.33	setOperationTime	29
2.1.34	getIdleTime	30
2.1.35	setIdleTime.....	30
2.1.36	getAutoOffTime	31
2.1.37	setAutoOffTime	31
2.1.38	getSleepTime	31
2.1.39	setSleepTime	32
2.1.40	getBarcodeTimeOut	32
2.1.41	setBarcodeTimeOut.....	33
2.1.42	getBaudRateList	33
2.1.43	getBaudRate	34
2.1.44	setBaudRate	34
2.1.45	getAccessPassword	34
2.1.46	setAccessPassword	35
2.1.47	getQuerySession.....	35
2.1.48	setQuerySession	36
2.1.49	getSessionFlag	36
2.1.50	setSessionFlag.....	37
2.1.51	getLinkProfile	37
2.1.52	setLinkProfile.....	37
2.1.53	getQValue.....	38
2.1.54	setQValue.....	38
2.1.55	getMaxQ.....	39
2.1.56	getMinQ.....	39
2.1.57	getSerialNumber.....	39
2.1.58	getBatteryStatus.....	40
2.1.59	getReportRSSI.....	40
2.1.60	setReportRSSI.....	41
2.1.61	clearEpcMask	41
2.1.62	getEpcMaskCount	42
2.1.63	addEpcMask	42
2.1.64	getEpcMask.....	42
2.1.65	getFrequencyAutomatic	43
2.1.66	setFrequencyAutomatic	43
2.1.67	getFrequencyList	44
2.1.68	setFrequencyList.....	44

2.1.69 startDecode	45
2.1.70 stopDecode	45
2.1.71 setCharset	46
2.1.72 getRegion	46
2.1.73 setScanMode	47
2.2 AsReaderP252BManager	48
2.2.1 getInstance	48
2.2.2 onDestroy	48
2.2.3 getVersion	48
2.3 AsReaderP252BEventListener	49
2.3.1 onStateChanged	49
2.3.2 onActionChanged	49
2.3.3 onReadTag	50
2.3.4 onAccessResult	51
2.3.5 onReadTemperatureTag	52
2.3.6 onReadHumidityTag	52
2.3.7 onKeyEvent	53
2.3.8 onModeKeyEvent	54
2.3.9 onReadBarcode	54
2.3.10 onReceivedData	55
2.4 AsReaderP252BBluetoothDiscoveryEventListener	55
2.4.1 onReceivedDevice	55
2.4.2 onFoundDeviceFinished	56
2.5 AsReaderP252BDeviceUsbCdc	56
2.5.1 AsReaderP252BDeviceUsbCdc	56
2.6 AsReaderP252BDeviceBluetoothCdc	57
2.6.1 AsReaderP252BDeviceBluetoothCdc	57
2.6.2 startDiscovery	57
2.6.3 stopDiscovery	57
2.7 AsReaderP252BLbtItem	58
2.7.1 getSlot	58
2.7.2 isUsed	58
2.7.3 setUsed	58
2.7.4 getFrequency	59
2.8 AsReaderP252BLockParam	59
2.8.1 AsReaderP252BLockParam	59
2.9 AsReaderP252BPowerRange	60
2.9.1 getMin	60

2.9.2 getMax.....	60
2.10 AsReaderP252BSelectMaskEpcParam	61
2.10.1 AsReaderP252BSelectMaskEpcParam.....	61
2.10.2 getOffset	61
2.10.3 setOffset.....	61
2.10.4 getLength	61
2.10.5 setLength	62
2.10.6 getMask.....	62
2.10.7 setMask	62
2.11 Enum	63
2.11.1 AsReaderP252BConnectionState	63
2.11.2 AsReaderP252BKeyState	63
2.11.3 AsReaderP252BActionState	63
2.11.4 AsReaderP252BBuzzerState	63
2.11.5 AsReaderP252BMemoryBank.....	64
2.11.6 AsReaderP252BQuerySession	64
2.11.7 AsReaderP252BSessionFlag.....	64
2.11.8 AsReaderP252BResultCode	64
2.11.9 AsReaderP252BScanMode.....	65
2.11.10 AsReaderP252BTemperatureTagType	65
2.11.11 AsReaderP252BBarcodeType	65

概要

本文の主な目的：

- 開発者が AsReaderP252BSDK 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 Arctic Fox | 2020.3.1
- Android SDK 24
- Android Gradle 8.1

システム要求：

- Android 10.0+

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)

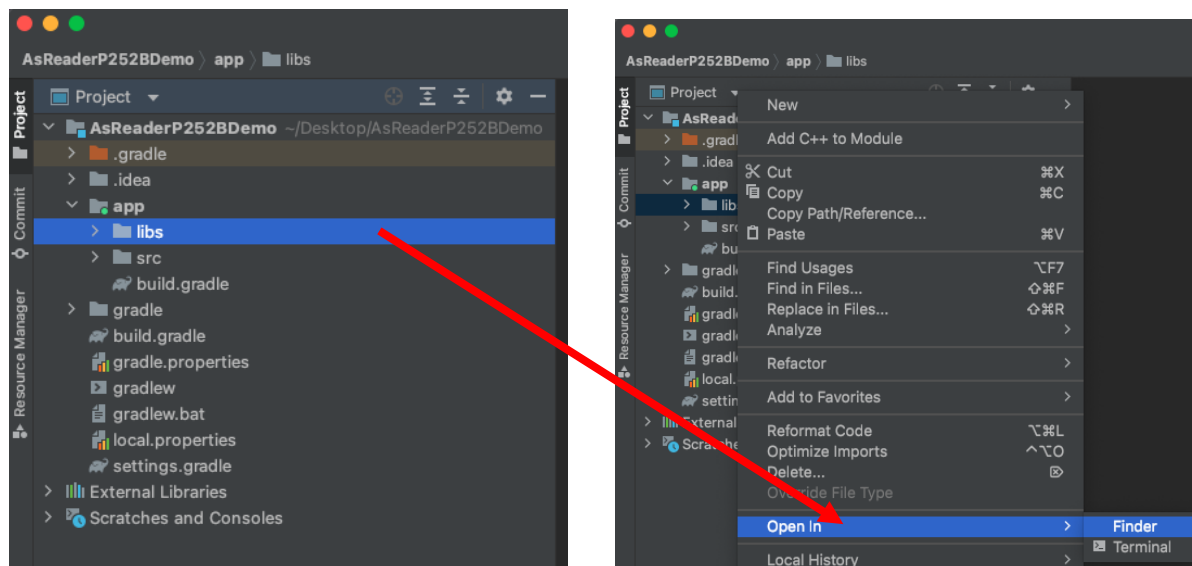


図 1-1-1

2. ダウンロードした「AsReaderP252BSDK.aar」を選択します。(図 1-1-2)
プロジェクトの「libs」の下に「AsReaderP252BSDK.aar」が表示されます。(図 1-1-3)

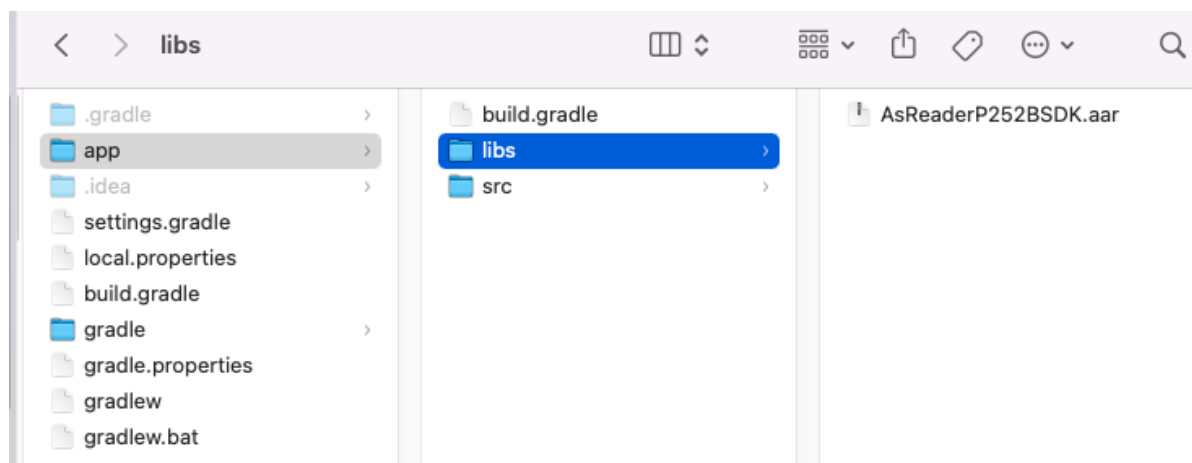


図 1-1-2

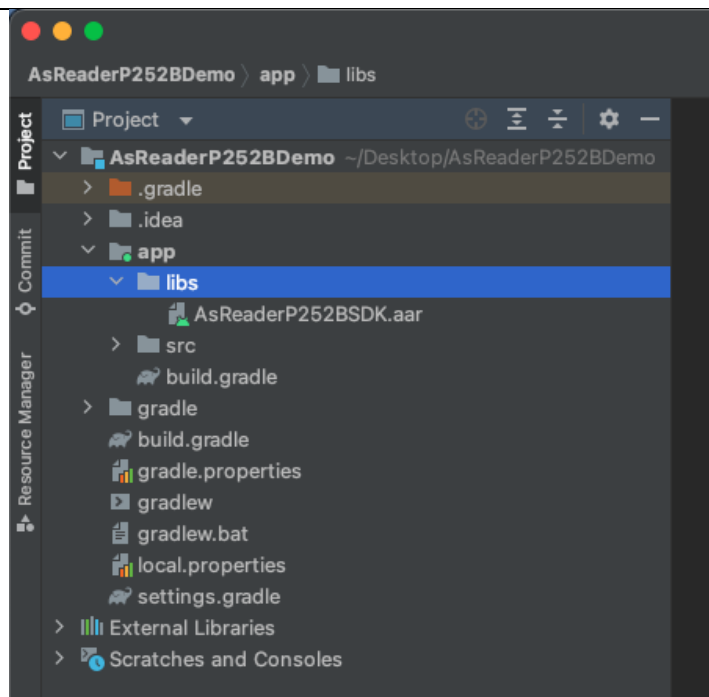


図 1-1-3

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

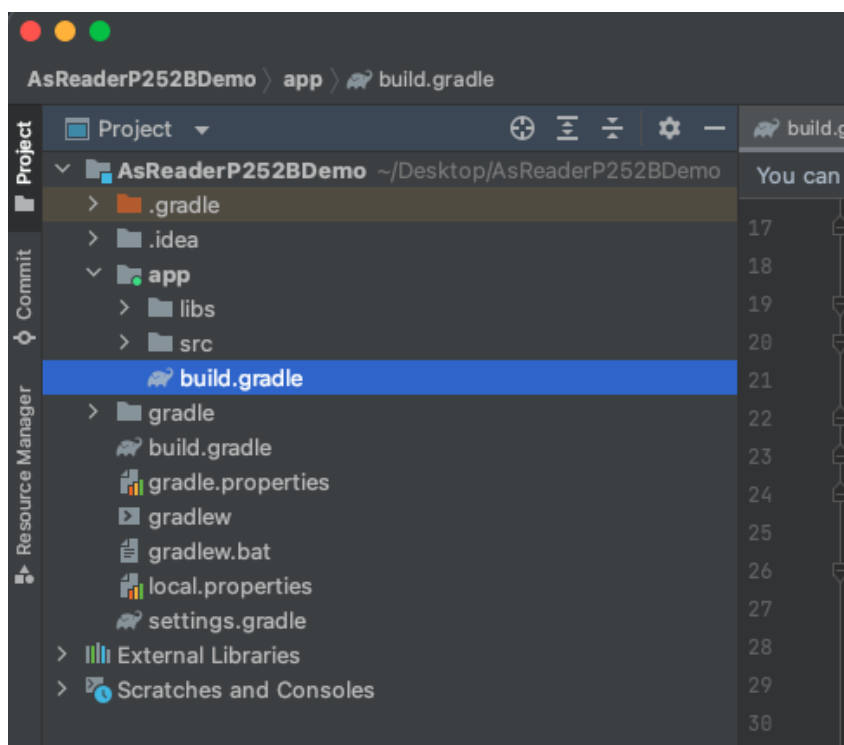


図 1-1-4

4. 依存関係にライブラリをインポートします。(図 1-1-5 1)
次に「Sync Now」をクリックし、同期します。(図 1-1-5 2)

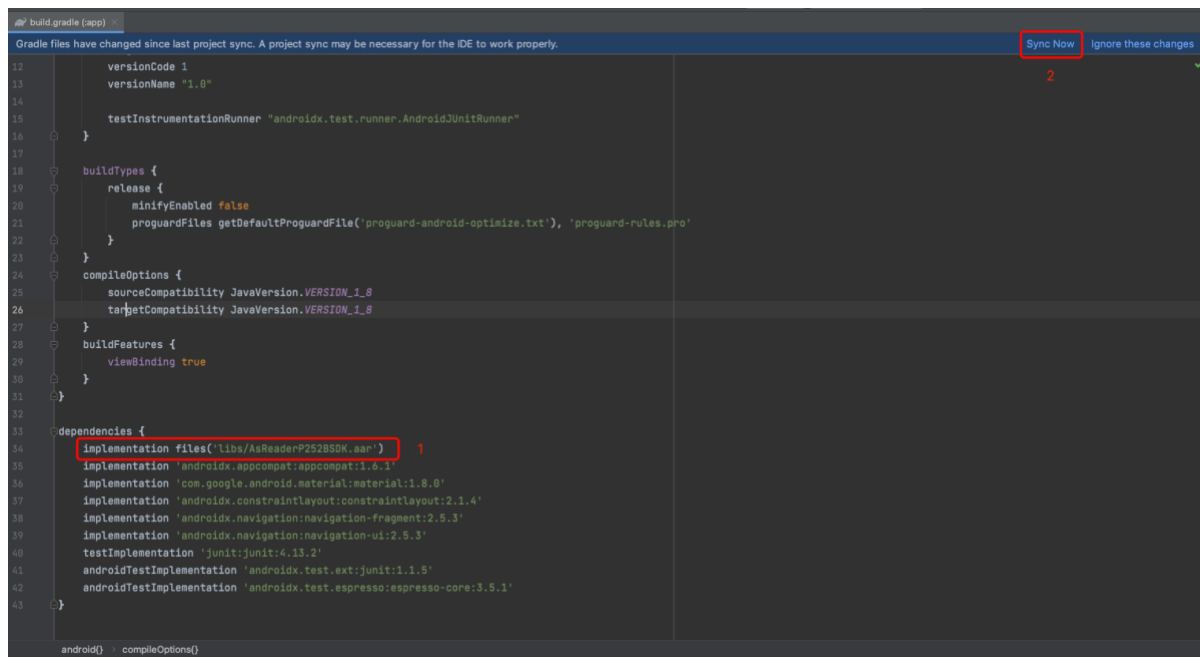


図 1-1-5

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。

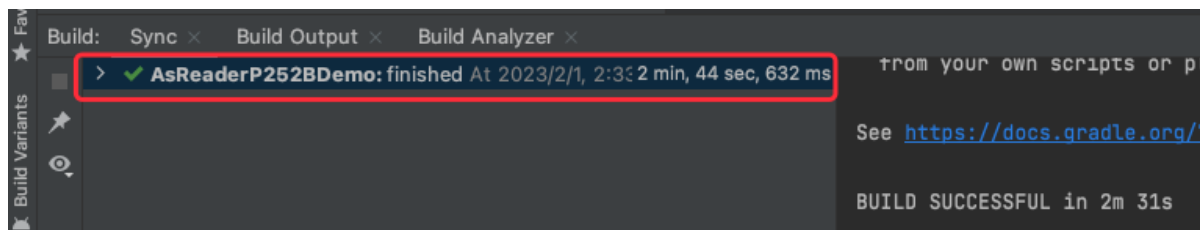


図 1-1-6

1.2 SDK の使用

1.2.1 SDK のインポート

SDK を使用するクラスで SDK をインポートします。(図 1-2-1)

```
import com.asreader.p252b.AsReaderP252B;
```

図 1-2-1

1.2.2 「AsReaderP252B」オブジェクトの作成・初期化

「AsReaderP252B」オブジェクトは AsReaderP252BManager を通じて取得できます。AsReaderP252BManager.getInstance()メソッドを呼び出し、「AsReaderP252B」オブジェクトを作成します。(図 1-2-2)

```
mReader = AsReaderP252BManager.getInstance();
```

図 1-2-2

1.2.3 AsReaderP252BEventListener インターフェースの実装

1. AsReaderP252BEventListener インターフェースを実装します。(図 1-2-3)

```
public class MainActivity extends Activity implements AsReaderP252BEventListener {
```

図 1-2-3

2. リスナーを設定します。(図 1-2-4)

```
mReader.setEventListener(this);
```

図 1-2-4

3. 実装した AsReaderP252BEventListener にカーソルを移動し、「Implement methods」が表示されたらクリックします。(図 1-2-5)

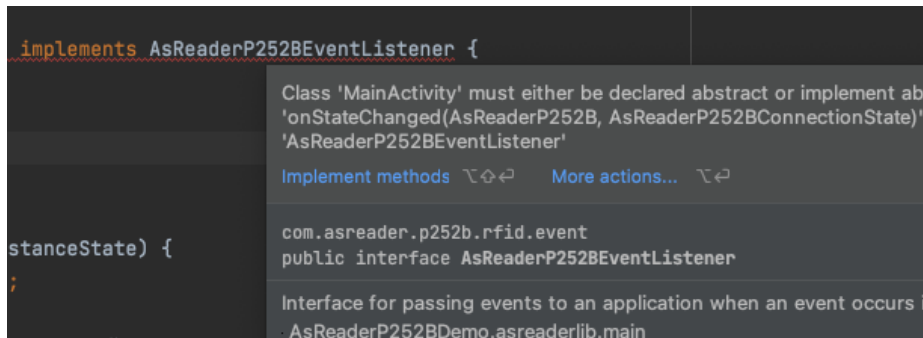


図 1-2-5

4. 青色背景部分を選択し、「OK」をクリックします。(図 1-2-6)

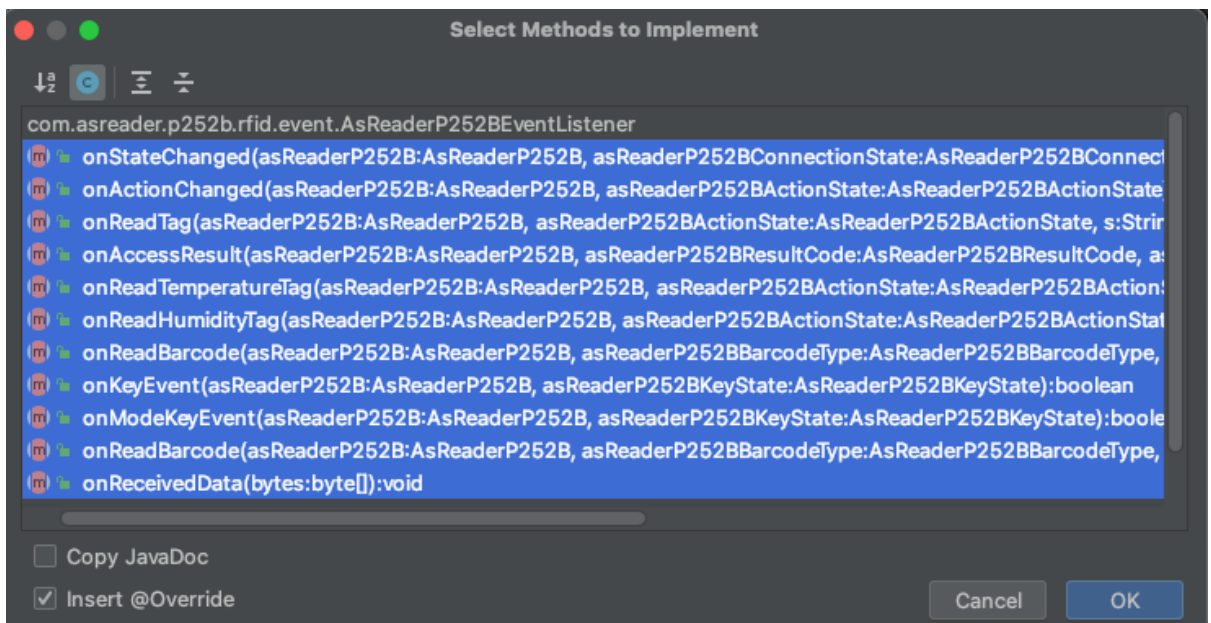


図 1-2-6

5. インターフェイス AsReaderP252BEventListener のメソッドが自動的に作成されます。
(図 1-2-7)

```
@Override
public void onStateChanged(AsReaderP252B reader, AsReaderP252BConnectionState state) {
}

@Override
public void onActionChanged(AsReaderP252B reader, AsReaderP252BActionState action) {
}

@Override
public void onReadTag(AsReaderP252B reader, AsReaderP252BActionState action, String tag, float rssi,
}

@Override
public void onAccessResult(AsReaderP252B reader, AsReaderP252BResultCode code, AsReaderP252BActionSta
}
```

図 1-2-7

1.2.4 接続

1.2.4.1 USB で接続

AsReaderP252BDeviceUsbCdc オブジェクトを作成します。(図 1-2-8)

```
AsReaderP252BDeviceUsbCdc usbCdc = new AsReaderP252BDeviceUsbCdc( context: this);
```

図 1-2-8

AsReaderP252BDeviceUsbCdc オブジェクトをパラメーターとして connectDevice() メソッドに渡します。(図 1-2-9)

```
mReader.connectDevice(usbCdc);
```

図 1-2-9

onStateChanged (2.3.1 を参照) メソッドを使って AsReader 接続状態をコールバックします。戻り値として「Conncted」が返されたら AsReader との接続は成功です。

1.2.4.2 Bluetooth で接続

AsReaderP252BDeviceBluetoothCdc オブジェクトを作成します。(図 1-2-10)

```
AsReaderP252BDeviceBluetoothCdc deviceBluetoothCdc = new AsReaderP252BDeviceBluetoothCdc( context: this);
```

図 1-2-10

AsReaderP252BBluetoothDiscoveryEventListener のリスナーメソッドを設定します。
(図 1-2-11)

```
deviceBluetoothCdc.setBluetoothDiscoveryEventListener(new AsReaderP252BBluetoothDiscoveryEventListener() {  
    @Override  
    public void onReceivedDevice(BluetoothDevice device) {  
        mBluetoothDevice = device;  
    }  
  
    @Override  
    public void onFoundDeviceFinished() {  
  
    }  
});
```

図 1-2-11

startDiscovery メソッド (2.6.2 を参照) を呼び出すことで Bluetooth デバイスの検索を開始します。Bluetooth 権限が許可されている場合、検索できた Bluetooth デバイスを取得します。また、stopDiscovery メソッド (2.6.3 を参照) を呼び出し Bluetooth デバイスの検索を停止できます。(図 1-2-12)

```
deviceBluetoothCdc.startDiscovery();  
deviceBluetoothCdc.stopDiscovery();
```

図 1-2-12

接続する Bluetooth デバイスのアドレスをパラメータとして
AsReaderP252BDeviceBluetoothCdc オブジェクトに渡します。(図 1-2-13)

```
String address = mBluetoothDevice.getAddress();  
deviceBluetoothCdc.setAddress(address);
```

図 1-2-13

AsReaderP252BDeviceBluetoothCdc オブジェクトをパラメータとして connectDevice()メソッドに渡します。(図 1-2-14)

```
mReader.connectDevice(deviceBluetoothCdc);
```

図 1-2-14

onStateChanged (2.3.1 を参照) メソッドを使って AsReader 接続状態をコールバックします。戻り値として「Conncceted」が返されたら AsReader との接続は成功です。

1.2.5 データのインベントリ

(例 : RFID をインベントリする)

AsReader が接続された後、RFID データをインベントリできます。(図 1-2-15)

```
AsReaderP252BResultCode res = getReader().inventory();  
if (res == AsReaderP252BResultCode.NoError) {  
    //Success to start inventory()  
}else{  
    //Failed to start inventory()  
}
```

図 1-2-15

onActionChanged (2.3.2 を参照) メソッドを使って AsReader の操作状態をコールバックします。(図 1-2-16)

```
@Override  
public void onActionChanged(AsReaderP252B reader, AsReaderP252BActionState action) {  
    //reader: AsReaderP252B対象  
    //action: AsReaderP252B当前操作状态  
}
```

図 1-2-16

イベントリしたデータは `onReadTag` (2.3.3 を参照) メソッドを使ってコールバックします。

(図 1-2-17)

```
@Override
public void onReadTag(AsReaderP252B reader, AsReaderP252BActionState action,
    String tag, float rssi, float phase, float frequency) {
    //reader: AsReaderP252B
    //action: AsReaderP252B当前操作状态
    //tag: 标签的十六进制字符串
    // rssi: 标签的RSSI的值
    // phase: 标签的phase值
    // frequency: 标签的frequency值
}
```

図 1-2-17

2 メソッド説明

2.1 AsReaderP252B

2.1.1 getResultCode

関数	public AsReaderP252BResultCode getResultCode()		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BResultCode	列挙型 AsReaderP252BResultCode (2.11.8 を参照)
<p>■ 説明 :</p> <p>コマンドの実行結果を取得します。</p> <p>■ サンプルコード :</p> <pre>AsReaderP252BResultCode resultcode = asReader.getResultCode(); if (resultcode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 } else { //接続失敗時の処理 }</pre>			

2.1.2 connectDevice

関数	public void connectDevice(AsReaderP252BDevice asReaderP252BDevice)		
引数	IN/OUT	型	説明
	IN	AsReaderP252BDevice	AsReaderP252BDevice オブジェクト
<p>■ 説明 :</p> <p>AsReader に接続します。 このメソッドを呼び出した後、onStateChanged (2.3.1 を参照) がコールバックされ、接続状態を取得します。</p> <p>■ サンプルコード :</p> <pre>asReader.connectDevice(mDevice);</pre>			

2.1.3 disconnectDevice

関数	public void disconnectDevice()
<p>■説明： AsReader との接続を切断します。 このメソッドを呼び出した後、onStateChanged（2.3.1 を参照）がコールバックされ、接続状態を取得します。</p> <p>■サンプルコード： asReader.disconnectDevice();</p>	

2.1.4 getCurrentDevice

関数	public AsReaderP252BDevice getCurrentDevice()		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BDevice	AsReaderP252BDevice オブジェクト (注：AsReader と接続していない場合、null を返す)
<p>■説明： 現在接続している AsReaderP252BDevice オブジェクトを取得します。</p> <p>■サンプルコード： AsReaderP252BDevice device = asReader.getCurrentDevice();</p>			

2.1.5 getState

関数	public AsReaderP252BConnectionState getState()		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BConnectionState	列挙型 AsReaderP252BConnectionState (2.11.1 を参照)
<p>■説明： 現在の AsReader の接続状態を取得します。</p> <p>■サンプルコード： AsReaderP252BConnectionState connectStage = asReader.getState();</p>			

2.1.6 getAction

関数	public AsReaderP252BActionState getAction()		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BActionState	列挙型 AsReaderP252BActionState (2.11.3 を参照)

■ 説明 :

AsReaderP252B インスタンスの現在の動作状態を取得します。

■ サンプルコード :

```
AsReaderP252BActionState actionStage = asReader.getAction();
```

2.1.7 getFirmwareVersion

関数	public String getFirmwareVersion()		
引数	IN/OUT	型	説明
	OUT	String	AsReader のファームウェアのバージョン
■ 説明 : AsReader のファームウェアのバージョン情報を取得します。			
■ サンプルコード : <pre>String firmwareVersion = asReader.getFirmwareVersion();</pre>			

2.1.8 getHardwareVersion

関数	public String getHardwareVersion()		
引数	IN/OUT	型	説明
	OUT	String	AsReader のハードウェアのバージョン
■ 説明 : AsReader のハードウェアのバージョン情報を取得します。			
■ サンプルコード : <pre>String hardwareVersion = asReader.getHardwareVersion();</pre>			

2.1.9 getRFModuleVersion

関数	public String getRFModuleVersion()		
引数	IN/OUT	型	説明
	OUT	String	AsReader の RF モジュールのバージョン
■ 説明 : AsReader の RF モジュールのバージョン情報を取得します。			
■ サンプルコード : <pre>String rFModuleVersion = asReader.getRFModuleVersion();</pre>			

2.1.10 getPowerGainRange

関数	public AsReaderP252BPowerRange getPowerGainRange()		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BPowerRange	電波出力の設定可能な範囲 AsReaderP252BPowerRange オブジェクト (2.9を参照)
<p>■ 説明 :</p> <p>AsReader の設定可能な最大出力値と最小出力値を取得します。</p> <p>■ サンプルコード :</p> <pre>try { AsReaderP252BPowerRange powerGainRange = asReader.getPowerGainRange(); int min = powerGainRange.min; //設定可能な最小出力値 int max = powerGainRange.max; //設定可能な最大出力値 } catch (AsReaderP252BException e) { }</pre>			

2.1.11 setEventListener

関数	public void setEventListener(AsReaderP252BEventListener listener)		
引数	IN/OUT	型	説明
listener	IN	AsReaderP252BEventListener	AsReaderP252BEventListener リスナーオブジェクト AsReaderP252BEventListener オブジェクト (2.3を参照)
<p>■ 説明 :</p> <p>AsReaderP252BEventListener リスナーを設定します。</p> <p>■ サンプルコード :</p> <pre>asReader.setEventListener(this);</pre>			

2.1.12 removeEventListener

関数	public void removeEventListener (AsReaderP252BEventListener listener)		
引数	IN/OUT	型	説明
listener	IN	AsReaderP252BEventListener	AsReaderP252BEventListener リスナーオブジェクト AsReaderP252BEventListener オブジェクト (2.3を参照)
<p>■ 説明 :</p> <p>AsReaderP252BEventListener リスナーを削除します。</p>			

```

■ サンプルコード :
asReader.removeEventListener(this) ;

```

2.1.13 inventory

関数	public AsReaderP252BResultCode inventory()		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8を参照)
<p>■ 説明 :</p> <p>タグのインベントリを開始します。</p> <p>このメソッドを呼び出した後、onActionChanged (2.3.2を参照) がコールバックされ、実行結果を取得します。また onReadTag (2.3.3を参照) がコールバックされ、読み取ったデータを受信します。</p> <p>■ サンプルコード :</p> <pre> AsReaderP252BResultCode resultCode = asReader.inventory(); if (resultcode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 } else { //接続失敗時の処理 } </pre>			

2.1.14 inventoryHumidityTag

関数	public AsReaderP252BResultCode inventoryHumidityTag()		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8を参照)
<p>■ 説明 :</p> <p>湿度タグのインベントリを開始します。</p> <p>このメソッドを呼び出した後、onActionChanged (2.3.2を参照) がコールバックされ、実行結果を取得します。また onReadHumidityTag (2.3.6を参照) がコールバックされ、読み取ったデータを受信します。</p> <p>■ サンプルコード :</p> <pre> AsReaderP252BResultCode resultCode = asReader.inventoryHumidityTag(); if (resultcode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 } else { //接続失敗時の処理 } </pre>			

```
}

```

2.1.15 inventoryTemperatureTagWithType

関数	public AsReaderP252BResultCode inventoryTemperatureTagWithType(AsReaderP252BTemperatureTagType type)		
引数	IN/OUT	型	説明
type	IN	AsReaderP252BTemperatureTagType	列挙型 AsReaderP252BTemperatureTagType (2.11.10 を参照)
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8 を参照)
<p>■説明：</p> <p>温度タグのインベントリを開始します。</p> <p>このメソッドを呼び出した後、onActionChanged (2.3.2 を参照) がコールバックされ、実行結果を取得します。また onReadTemperatureTag (2.3.5 を参照) がコールバックされ、読み取ったデータを受信します。</p> <p>■サンプルコード：</p> <pre>AsReaderP252BResultCode resultCode = asReader.inventoryTemperatureTagWithType (AsReaderP252BTemperatureTagType.Type_0); if (resultcode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 } else { //接続失敗時の処理 }</pre>			

2.1.16 readMemory

関数	public AsReaderP252BResultCode readMemory(AsReaderP252BMemoryBank bank, int offset, int length)		
引数	IN/OUT	型	説明
bank	IN	AsReaderP252BMemoryBank	タグのメモリバンク 列挙型 AsReaderP252BMemoryBank (2.11.5 を参照)
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	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8 を参照)

■ 説明 :

タグの指定されたメモリバンクのデータを読み取ります。

このメソッドを呼び出した後、onAccessResult (2.3.4 を参照) がコールバックされ、実行結果を受信します。。

■ サンプルコード :

```
AsReaderP252BResultCode resultCode = asReader.readMemory
(AsReaderP252BMemoryBank.EPC, 16, 4);
if (resultCode == AsReaderP252BResultCode.NoError) {
    //接続成功時の処理
} else {
    //接続失敗時の処理
}
```

2.1.17 writeMemory

関数	public AsReaderP252BResultCode writeMemory(AsReaderP252BMemoryBank bank, int offset, String data)		
引数	IN/OUT	型	説明
bank	IN	AsReaderP252BMemoryBank	タグのメモリバンク 列挙型 AsReaderP252BMemoryBank (2.11.5 を参照)
offset	IN	int	タグのスタートアドレス (単位 : word)
data	IN	String	書き込みデータ (データは 16 進数の文字列)
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8 を参照)

■ 説明 :

タグの指定されたメモリバンクのデータを書き換えます。

このメソッドを呼び出した後、onAccessResult (2.3.4 を参照) がコールバックされ、実行結果を受信します。

■ サンプルコード :

```
AsReaderP252BResultCode resultCode = asReader.
writeMemory(AsReaderP252BMemoryBank.EPC, 16, "1234");
if (resultCode == AsReaderP252BResultCode.NoError) {
    //接続成功時の処理
} else {
    //接続失敗時の処理
}
```

2.1.18 lock

関数	public AsReaderP252BResultCode lock(AsReaderP252BLockParam param)		
引数	IN/OUT	型	説明
param	IN	AsReaderP252BLockParam	AsReaderP252BLockParam オブジェクト (2.8 を参照)
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8 を参照)
<p>■ 説明 :</p> <p>タグの指定されたメモリバンクをロックします。 このメソッドを呼び出した後、onAccessResult (2.3.4 を参照) がコールバックされ、実行結果を受信します。</p> <p>■ サンプルコード :</p> <pre>AsReaderP252BResultCode resultCode = asReader.lock(param); if (resultCode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 } else { //接続失敗時の処理 }</pre>			

2.1.19 unlock

関数	public AsReaderP252BResultCode unlock (AsReaderP252BLockParam param)		
引数	IN/OUT	型	説明
param	IN	AsReaderP252BLockParam	AsReaderP252BLockParam オブジェクト (2.8 を参照)
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8 を参照)
<p>■ 説明 :</p> <p>タグのロックされたメモリ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データを書き換えます。 このメソッドを呼び出した後、onAccessResult (2.3.4 を参照) がコールバックされ、実行結果を受信します。</p> <p>■ サンプルコード :</p> <pre>AsReaderP252BResultCode resultCode = asReader.unlock(param); if (resultCode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 } else { //接続失敗時の処理 }</pre>			

2.1.20 permaLock

関数	public AsReaderP252BResultCode permaLock(AsReaderP252BLockParam param)		
引数	IN/OUT	型	説明
param	IN	AsReaderP252BLockParam	AsReaderP252BLockParam オブジェクト (2.8 を参照)
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8 を参照)
<p>■ 説明 :</p> <p>タグの指定されたメモリバンクを永久ロックします。 永久ロックされたメモリバンクのデータは書き換え、アンロックともできなくなります。 このメソッドを呼び出した後、onAccessResult (2.3.4 を参照) がコールバックされ、実行結果を受信します。</p> <p>■ サンプルコード :</p> <pre>AsReaderP252BResultCode resultCode = asReader.permaLock (param); if (resultCode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 } else { //接続失敗時の処理 }</pre>			

2.1.21 kill

関数	public AsReaderP252BResultCode kill(String killPassword)		
引数	IN/OUT	型	説明
killPassword	IN	String	キルパスワード
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8 を参照)
<p>■ 説明 :</p> <p>対象タグをキルします。 キルした後、タグが使えなくなります。 このメソッドを呼び出した後、onAccessResult (2.3.4 を参照) がコールバックされ、実行結果を受信します。 ※パスワードが「00000000」となっている場合、タグをキルできません。</p> <p>■ サンプルコード :</p> <pre>AsReaderP252BResultCode resultCode = asReader.kill("12345678"); if (resultCode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 }</pre>			

```

} else {
    //接続失敗時の処理
}

```

2.1.22 stop

関数	public AsReaderP252BResultCode stop()		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8を参照)
<p>■ 説明 :</p> <p>タグのインベントリを停止します。 このメソッドを呼び出した後、onActionChanged (2.3.2を参照) がコールバックされ、実行結果を受信します。</p> <p>■ サンプルコード :</p> <pre> AsReaderP252BResultCode resultCode = asReader.stop(); if (resultCode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 } else { //接続失敗時の処理 } </pre>			

2.1.23 defaultParameter

関数	public AsReaderP252BResultCode defaultParameter()		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8を参照)
<p>■ 説明 :</p> <p>全てのパラメータ設定をデフォルト値に戻します。</p> <p>■ サンプルコード :</p> <pre> AsReaderP252BResultCode resultCode = asReader.defaultParameter(); if (resultCode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 } else { //接続失敗時の処理 } </pre>			

2.1.24 saveParameter

関数	public AsReaderP252BResultCod		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BResultCod e	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8 を参照)
<p>■説明： 全てのパラメータ設定を AsReader に保存します。</p> <p>■サンプルコード： <pre>AsReaderP252BResultCode resultCode = asReader.saveParameter(); if (resultcode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 } else { //接続失敗時の処理 }</pre> </p>			

2.1.25 getBuzzer

関数	public AsReaderP252BBuzzerState getBuzzer() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BBuzzerS tate	列挙型 AsReaderP252BBuzzerState (2.11.4 を参照)
	-	AsReaderP252BExceptio n	データの取得に問題がある場合 (例: 他のコマンドが 実行中、クラッシュなど) に例外を返します。
<p>■説明： ブザーのパラメータを取得します。</p> <p>■サンプルコード： <pre>try { AsReaderP252BBuzzerState mBuzzer = asReader.getBuzzer(); } catch (AsReaderP252BException e) { }</pre> </p>			

2.1.26 setBuzzer

関数	public void setBuzzer(AsReaderP252BBuzzerState state) throws AsReaderP252BException		
引数	IN/OUT	型	説明

	IN	AsReaderP252BBuzzerState	列挙型 AsReaderP252BBuzzerState (2.11.4 を参照)
	-	AsReaderP252BException	データの設定に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> ブザーのパラメータを設定します。			
<p>■ サンプルコード :</p> <pre>try { asReader.setBuzzer(AsReaderP252BBuzzerState.Off); } catch (AsReaderP252BException e) { }</pre>			

2.1.27 getContinuousMode

関数	public boolean getContinuousMode() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	boolean	true: 連続読取有効 false: 連続読取無効
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> インベントリを連続実行するかどうかを取得します。			
<p>■ サンプルコード :</p> <pre>try { boolean continuousMode = asReader.getContinuousMode(); } catch (AsReaderP252BException e) { }</pre>			

2.1.28 setContinuousMode

関数	public void setContinuousMode(boolean enabled) throws AsReaderP252BException		
引数	IN/OUT	型	説明
enable	IN	boolean	true: 連続読取有効 false: 連続読取無効
	-	AsReaderP252BException	データの設定に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> インベントリを連続実行するかどうかを設定します。			

■ サンプルコード :

```
try {
    asReader.setContinuousMode(true);
} catch (AsReaderP252BException e) {
}
```

注 : AsReader の電源起動時の初期設定では連続読取モードは無効となります。必要に応じてこの連続読取機能を設定してください。

2.1.29 getPowerGain

関数	public int getPowerGain() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	電波出力(単位: dBm × 10)
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> <p>AsReader の電波出力を取得します。</p> <p>■ サンプルコード :</p> <pre>try { int powerGain = asReader.getPowerGain(); } catch (AsReaderP252BException e) { }</pre>			

2.1.30 setPowerGain

関数	public void setPowerGain(int power) throws AsReaderP252BException		
引数	IN/OUT	型	説明
power	IN	int	電波出力(単位: dBm × 10) 範囲 : 2~30
	-	AsReaderP252BException	データの設定に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> <p>AsReader の電波出力を設定します。</p> <p>■ サンプルコード :</p> <pre>try { asReader.setPowerGain(powerGain); } catch (AsReaderP252BException e) { }</pre>			

2.1.31 getPowerGainRange

関数	public AsReaderP252BPowerRange getPowerGainRange() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BPowerRange	電波出力の設定可能な範囲 AsReaderP252BPowerRange オブジェクト (2.9を参照)
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■説明:</p> <p>AsReader の設定可能な最大出力値と最小出力値を取得します。</p> <p>■サンプルコード:</p> <pre>try { AsReaderP252BPowerRange powerGainRange = asReader.getPowerGainRange(); int min = powerGainRange.min; //設定可能な最小出力値 int max = powerGainRange.max; //設定可能な最大出力値 } catch (AsReaderP252BException e) { }</pre>			

2.1.32 getOperationTime

関数	public int getOperationTime() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	インベントリ継続時間 (単位 : s)
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■説明:</p> <p>一回のインベントリの継続時間を取得します。</p> <p>■サンプルコード:</p> <pre>try { int mOperationTime = asReader.getOperationTime(); } catch (AsReaderP252BException e) { }</pre>			

2.1.33 setOperationTime

関数	public void setOperationTime(int time) throws AsReaderP252BException		
引数	IN/OUT	型	説明

time	IN	int	インベントリ継続時間（単位：s） 範囲：0~1800
	-	AsReaderP252BException	データの設定に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明： 一回のインベントリの継続時間を設定します。</p> <p>■サンプルコード： <pre>try { asReader.setOperationTime(200); } catch (AsReaderP252BException e) { }</pre> </p>			

2.1.34 getIdleTime

関数	public int getIdleTime() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	電波出力停止時間（単位：10ms）
	-	AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明： インベントリ中の電波出力停止時間を取得します。</p> <p>■サンプルコード： <pre>try { int idleTime = asReader.getIdleTime(); } catch (AsReaderP252BException e) { }</pre> </p>			

2.1.35 setIdleTime

関数	public void setIdleTime(int time) throws AsReaderP252BException		
引数	IN/OUT	型	説明
time	IN	int	電波出力停止時間（単位：10ms） 範囲：0~65535
	-	AsReaderP252BException	データの設定に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明： インベントリ中の電波出力停止時間を設定します。</p> <p>■サンプルコード：</p>			

```
try {
    asReader.setIdleTime(300);
} catch (AsReaderP252BException e) {
}
```

2.1.36 getAutoOffTime

関数	public int getAutoOffTime () throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	オートオフ時間（単位：s）
	-	AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■ 説明：</p> <p>AsReader 未接続時に電源が自動でオフになるまでの時間を取得します。</p> <p>■ サンプルコード：</p> <pre>try { int autoOffTime = asReader.getAutoOffTime(); } catch (AsReaderP252BException e) { }</pre>			

2.1.37 setAutoOffTime

関数	public void setAutoOffTime(int time) throws AsReaderP252BException		
引数	IN/OUT	型	説明
time	IN	int	オートオフ時間（単位：s） 範囲：0~1800
	-	AsReaderP252BException	データの設定に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■ 説明：</p> <p>AsReader 未接続時に電源が自動でオフになるまでの時間を設定します。</p> <p>■ サンプルコード：</p> <pre>try { asReader.setAutoOffTime(200); } catch (AsReaderP252BException e) { }</pre>			

2.1.38 getSleepTime

関数	public int getSleepTime() throws AsReaderP252BException		
----	---	--	--

引数	IN/OUT	型	説明
	OUT	int	自動スリープ時間 (単位 : s)
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> <p>AsReader がスリープするまでの時間を取得します。</p> <p>■ サンプルコード :</p> <pre>try { int mSleepTime = asReader.getSleepTime(); } catch (AsReaderP252BException e) { } </pre> <p>注 : AsReader が切断された後からカウントします。</p>			

2.1.39 setSleepTime

関数	public void setSleepTime(int time) throws AsReaderP252BException		
引数	IN/OUT	型	説明
time	IN	int	自動スリープ時間 (単位 : s) 範囲 : 0~1800
	-	AsReaderP252BException	データの設定に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> <p>AsReader がスリープするまでの時間を設定します。</p> <p>■ サンプルコード :</p> <pre>try { asReader.setSleepTime(200); } catch (AsReaderP252BException e) { } </pre>			

2.1.40 getBarcodeTimeOut

関数	public int getBarcodeTimeOut() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	スキャンのタイムアウト時間 (単位 : s)
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> <p>AsReader の Barcode スキャンのタイムアウト時間を取得します。</p>			

■ サンプルコード :

```
try {
    int mBarcodeTimeOut= asReader.getBarcodeTimeOut();
} catch (AsReaderP252BException e) {
}
```

2.1.41 setBarcodeTimeOut

関数	public void setBarcodeTimeOut(int time) throws AsReaderP252BException		
引数	IN/OUT	型	説明
time	IN	int	スキャンのタイムアウト時間 (単位 : s) 範囲 : 4~300
	-	AsReaderP252BException	データの設定に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> <p>AsReader の Barcode スキャンのタイムアウト時間を設定します。</p> <p>■ サンプルコード :</p> <pre>try { asReader.setBarcodeTimeOut(10); } catch (AsReaderP252BException e) { }</pre>			

2.1.42 getBaudRateList

関数	public ArrayList<String> getBaudRateList() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	ArrayList	ボーレートリスト
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> <p>AsReader のボーレートリストを取得します。</p> <p>■ サンプルコード :</p> <pre>try { ArrayList mBaudRateList = asReader.getBaudRateList(); } catch (AsReaderP252BException e) { }</pre>			

2.1.43 getBaudRate

関数	public int getBaudRate() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	ボーレートリストでのインデックス
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■説明:</p> <p>AsReader が利用中のボーレートのインデックスを取得します。</p> <p>■サンプルコード:</p> <pre>try { int mBaudRate = asReader.getBaudRate(); } catch (AsReaderP252BException e) { }</pre>			

2.1.44 setBaudRate

関数	public void setBaudRate(int rate) throws AsReaderP252BException		
引数	IN/OUT	型	説明
rate	IN	int	ボーレートリストでのインデックス
	-	AsReaderP252BException	データの設定に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■説明:</p> <p>AsReader が利用するボーレートのインデックスを設定します。</p> <p>■サンプルコード:</p> <pre>try { asReader.setBaudRate(1); } catch (AsReaderP252BException e) { }</pre>			

2.1.45 getAccessPassword

関数	public String getAccessPassword() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	String	アクセスパスワード
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■説明:</p> <p>ロックされたタグにアクセスするために必要なアクセスパスワードを取得します。</p>			

■ サンプルコード :

```
try {
    String mAccessPassword = asReader.getAccessPassword();
} catch (AsReaderP252BException e) {
}
```

2.1.46 setAccessPassword

関数	public void setAccessPassword(String password) throws AsReaderP252BException		
引数	IN/OUT	型	説明
password	IN	String	アクセスパスワード
	-	AsReaderP252BException	データの設定に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> <p>ロックされたタグにアクセスするために必要なアクセスパスワードを設定します。</p> <p>※RFID の仕様としてアクセスパスワードが"00000000"の場合は、パスワード設定が初期状態となります。</p> <p>■ サンプルコード :</p> <pre>try { asReader.setAccessPassword("12345678"); } catch (AsReaderP252BException e) { }</pre>			

2.1.47 getQuerySession

関数	public AsReaderP252BQuerySession getQuerySession() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BQuerySession	列挙型 AsReaderP252BQuerySession (2.11.6 を参照)
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> <p>インベントリのセッション値を取得します。</p> <p>■ サンプルコード :</p> <pre>try { AsReaderP252BQuerySession mSession = asReader.getQuerySession();</pre>			

```
} catch (AsReaderP252BException e) {
}
```

2.1.48 setQuerySession

関数	public void setQuerySession(AsReaderP252BQuerySession session) throws AsReaderP252BException		
引数	IN/OUT	型	説明
session	IN	AsReaderP252BQuerySession	列挙型 AsReaderP252BQuerySession (2.11.6 を参照)
	-	AsReaderP252BException	データの設定に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■説明:</p> <p>インベントリのセッション値を設定します。</p> <p>■サンプルコード:</p> <pre>try { asReader.setQuerySession(AsReaderP252BQuerySession.S0); } catch (AsReaderP252BException e) { }</pre>			

2.1.49 getSessionFlag

関数	public AsReaderP252BSessionFlag getSessionFlag() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BSessionFlag	列挙型 AsReaderP252BSessionFlag (2.11.7 を参照)
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■説明:</p> <p>インベントリのセッションフラグを取得します。</p> <p>■サンプルコード:</p> <pre>try { AsReaderP252BSessionFlag mTarget = asReader.getSessionFlag(); } catch (AsReaderP252BException e) { }</pre>			

2.1.50 setSessionFlag

関数	public void setSessionFlag(AsReaderP252BSessionFlag target) throws AsReaderP252BException		
引数	IN/OUT	型	説明
target	IN	AsReaderP252BSessionFlag	列挙型 AsReaderP252BSessionFlag (2.11.7を参照)
	-	AsReaderP252BException	データの設定に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■説明： インベントリのセッションフラグを設定します。</p> <p>■サンプルコード： <pre>try { asReader.setSessionFlag(AsReaderP252BSessionFlag.AB); } catch (AsReaderP252BException e) { }</pre> </p>			

2.1.51 getLinkProfile

関数	public int getLinkProfile() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	RFID モジュールの Link Profile 値
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■説明： AsReader の RFID モジュールの Link Profile 値を取得します。</p> <p>■サンプルコード： <pre>try { int mLinkProfile = asReader.getLinkProfile(); } catch (AsReaderP252BException e) { }</pre> </p>			

2.1.52 setLinkProfile

関数	public void setLinkProfile(int value) throws AsReaderP252BException		
引数	IN/OUT	型	説明
value	IN	int	RFID モジュールの Link Profile 値 範囲：0~3

	-	AsReaderP252BException	データの設定に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明： AsReader の RFID モジュールの Link Profile 値を設定します。</p> <p>■サンプルコード： <pre>try { asReader.setLinkProfile(1); } catch (AsReaderP252BException e) { }</pre> </p>			

2.1.53 getQValue

関数	public int getQValue() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	現在の Q 値
	-	AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明： インベントリの現在の Q 値を取得します。</p> <p>■サンプルコード： <pre>try { int mQValue = asReader.getQValue(); } catch (AsReaderP252BException e) { }</pre> </p>			

2.1.54 setQValue

関数	public void setQValue(int value) throws AsReaderP252BException		
引数	IN/OUT	型	説明
value	IN	int	現在の Q 値 範囲：0~15
	-	AsReaderP252BException	データの設定に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明： インベントリの現在の Q 値を設定します。</p> <p>■サンプルコード： <pre>try { asReader.setQValue(10); }</pre> </p>			

```
} catch (AsReaderP252BException e) {
}
```

2.1.55 getMaxQ

関数	public int getMaxQ() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	インベントリの最大 Q 値
	-	AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明：</p> <p>インベントリの最大 Q 値を取得します。</p> <p>■サンプルコード：</p> <pre>try { int maxQ = asReader.getMaxQ(); } catch (AsReaderP252BException e) { }</pre>			

2.1.56 getMinQ

関数	public int getMinQ() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	インベントリの最小 Q 値
	-	AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明：</p> <p>インベントリの最小 Q 値を取得します。</p> <p>■サンプルコード：</p> <pre>try { int minQ = asReader.getMinQ(); } catch (AsReaderP252BException e) { }</pre>			

2.1.57 getSerialNumber

関数	public String getSerialNumber() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	String	AsReader のシリアル番号。

	-	AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明： AsReader のシリアル番号を取得します。</p> <p>■サンプルコード：</p> <pre>try { String mSerialNumber = asReader.getSerialNumber(); } catch (AsReaderP252BException e) { }</pre>			

2.1.58 getBatteryStatus

関数	public int getBatteryStatus() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	AsReader の電池残量 範囲：0、1、2、3、4
	-	AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明： AsReader の電池残量を取得します。</p> <p>0: 0 1: 25% 2: 50% 3: 75% 4: 100%</p> <p>■サンプルコード：</p> <pre>try { int mBattery = asReader.getBatteryStatus(); } catch (AsReaderP252BException e) { }</pre>			

2.1.59 getReportRSSI

関数	public boolean getReportRSSI() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	boolean	true : RSSI データを取得する false : RSSI データを取得しない
	-	AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。

■説明：

イベントリでタグを読み取る時に RSSI データを受信するかどうかを取得します。

■サンプルコード：

```
try {
    boolean mReportRSSI = asReader.getReportRSSI();
} catch (AsReaderP252BException e) {
}
```

2.1.60 setReportRSSI

関数	public void setReportRSSI(boolean enabled) throws AsReaderP252BException		
引数	IN/OUT	型	説明
Enabled	IN	boolean	true : RSSI データを取得する false : RSSI データを取得しない
	-	AsReaderP252BException	データの設定に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。

■説明：

イベントリでタグを読み取る時に RSSI データを受信するかどうかを設定します。

■サンプルコード：

```
try {
    asReader.setReportRSSI(true);
} catch (AsReaderP252BException e) {
}
```

2.1.61 clearEpcMask

関数	public void clearEpcMask() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	-	AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。

■説明：

EPC マスクデータを削除します。

■サンプルコード：

```
try {
    asReader.clearEpcMask();
} catch (AsReaderP252BException e) {
}
```

2.1.62 getEpcMaskCount

関数	public int getEpcMaskCount() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	int	EPC マスクの数量
		AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明：</p> <p>保存されている EPC マスクの数量を取得します。</p> <p>■サンプルコード：</p> <pre>try { int mCount = asReader.getEpcMaskCount(); } catch (AsReaderP252BException e) { }</pre>			

2.1.63 addEpcMask

関数	public void addEpcMask(AsReaderP252BSelectMaskEpcParam param) throws AsReaderP252BException		
引数	IN/OUT	型	説明
param	IN	AsReaderP252BSelectMaskEpcParam	AsReaderP252BSelectMaskEpcParam オブジェクト（2.10を参照）
		AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明：</p> <p>AsReaderP252BSelectMaskEpcParam オブジェクトの形式で EPC マスクを追加します。</p> <p>■サンプルコード：</p> <pre>try { asReader.addEpcMask(mask); } catch (AsReaderP252BException e) { }</pre>			

2.1.64 getEpcMask

関数	public AsReaderP252BSelectMaskEpcParam getEpcMask(int index) throws AsReaderP252BException		
引数	IN/OUT	型	説明
index	IN	Int	EPC マスクのインデックス

	OUT	AsReaderP252BSelectMaskEpcParam	メソッドの実行結果 AsReaderP252BSelectMaskEpcParam オブジェクト (2.10 を参照)
		AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> 指定されたインデックスの EPC マスクを取得します。			
<p>■ サンプルコード :</p> <pre>try { AsReaderP252BSelectMaskEpcParam mask = asReader.getEpcMask(0); } catch (AsReaderP252BException e) { }</pre>			

2.1.65 getFrequencyAutomatic

関数	public boolean getFrequencyAutomatic() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	Boolean	周波数ホッピングを使用するかどうか YES : 周波数ホッピング有効 NO : 周波数ホッピング無効
		AsReaderP252BException	データの取得に問題がある場合 (例: 他のコマンドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> インベントリで周波数ホッピングを使用するかどうかを取得します。			
<p>■ サンプルコード :</p> <pre>try { boolean mFrequencyAutomatic = asReader.getFrequencyAutomatic(); } catch (AsReaderP252BException e) { }</pre>			

2.1.66 setFrequencyAutomatic

関数	public void setFrequencyAutomatic(boolean isAutomatic) throws AsReaderP252BException		
引数	IN/OUT	型	説明
isAutomatic	IN	Boolean	周波数ホッピングを使用するかどうか YES : 周波数ホッピング有効 NO : 周波数ホッピング無効

		AsReaderP252BException	データの設定に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■ 説明： インベントリで周波数ホッピングを使用するかどうかを設定します。</p> <p>■ サンプルコード： <pre>try { asReader.setFrequencyAutomatic(true); } catch (AsReaderP252BException e) { }</pre> </p>			

2.1.67 getFrequencyList

関数	public AsReaderP252BLbtItem[] getFrequencyList() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BLbtItem[]	周波数リスト情報
		AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■ 説明： RFID タグをインベントリした時の周波数リスト情報を取得します。</p> <p>■ サンプルコード： <pre>try { AsReaderP252BLbtItem[] mItems = asReader.getFrequencyList(); } catch (AsReaderP252BException e) { }</pre> </p>			

2.1.68 setFrequencyList

関数	public void setFrequencyList(AsReaderP252BLbtItem[] table) throws AsReaderP252BException		
引数	IN/OUT	型	説明
table	IN	AsReaderP252BLbtItem[]	周波数リスト情報
		AsReaderP252BException	データの設定に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■ 説明： RFID タグをインベントリした時の周波数リストを設定します。</p> <p>■ サンプルコード： <pre>try {</pre> </p>			

```
asReader.setFrequencyList(table);
} catch (AsReaderP252BException e) {
}
```

2.1.69 startDecode

関数	public AsReaderP252BResultCod startDecode();		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8を参照)
<p>■ 説明 :</p> <p>バーコードの読み取りを開始します。 このメソッドを呼び出した後、onReadBarcode (2.3.9を参照) がコールバックされ、読み取ったデータを受信します。</p> <p>■ サンプルコード :</p> <pre>AsReaderP252BResultCode resultCode = asReader.startDecode(); if (resultCode == AsReaderP252BResultCode.NoError) { //接続成功時の処理 }else{ //接続失敗時の処理 }</pre>			

2.1.70 stopDecode

関数	public AsReaderP252BResultCod stopDecode();		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8を参照)
<p>■ 説明 :</p> <p>バーコードの読み取りを停止します。</p> <p>■ サンプルコード :</p> <pre>AsReaderP252BResultCode resultCode = asReader.stopDecode(); if (resultCode == AsReaderP252BResultCode.NoError) { //接続成功時の処理</pre>			

```

}else{
    //接続失敗時の処理
}

```

2.1.71 setCharset

関数	public void setCharset(Charset charset) throws AsReaderP252BException		
引数	IN/OUT	型	説明
charset	IN	Charset	キャラクターセット
		AsReaderP252BException	データの設定に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明：</p> <p>バーコードスキャンのエンコード方法を設定します。</p> <p>■サンプルコード：</p> <pre> try { asReader.setCharset(Charset.forName("ASCII")); } catch (AsReaderP252BException e) { } </pre>			

2.1.72 getRegion

関数	public String getRegion() throws AsReaderP252BException		
引数	IN/OUT	型	説明
	OUT	String	地域
		AsReaderP252BException	データの取得に問題がある場合（例：他のコマンドが実行中、クラッシュなど）に例外を返します。
<p>■説明：</p> <p>AsReader を使用する地域を取得します。</p> <p>■サンプルコード：</p> <pre> try { String mGlobalBand = asReader.getRegion(); } catch (AsReaderP252BException e) { } </pre>			

2.1.73 setScanMode

関数	public void setScanMode(AsReaderP252BScanMode scanMode)throws AsReaderP252BException		
引数	IN/OUT	型	説明
scanMode	IN	AsReaderP252BScanMod e	スキャンモード 列挙型 AsReaderP252BScanMode (2.11.9 を参照)
		AsReaderP252BException	データの設定に問題がある場合 (例: 他のコマン ドが実行中、クラッシュなど) に例外を返します。
<p>■ 説明 :</p> <p>AsReader のスキャンモードを設定します。</p> <p>■ サンプルコード :</p> <pre>try { asReader.setScanMode(AsReaderP252BScanMode.RFIDScanMode); } catch (AsReaderP252BException e) { }</pre>			

2.2 AsReaderP252BManager

2.2.1 getInstance

関数	public static AsReaderP252B getInstance()		
引数	IN/OUT	型	説明
	OUT	AsReaderP252B	AsReaderP252B クラスのインスタンスオブジェクト（2.1 を参照）
<p>■ 説明： AsReaderP252B クラスのインスタンスオブジェクトを取得します。</p> <p>■ サンプルコード： AsReaderP252B asReader = AsReaderP252BManager.getInstance();</p>			

2.2.2 onDestroy

関数	public static void onDestroy()		
<p>■ 説明： AsReaderP252B クラスのインスタンスオブジェクトを廃棄し、リソースを解放します。</p> <p>■ サンプルコード： AsReaderP252BManager.onDestroy();</p>			

2.2.3 getVersion

関数	public static String getVersion()		
引数	IN/OUT	型	説明
	OUT	String	SDK バージョン
<p>■ 説明： SDK バージョンを取得します。</p> <p>■ サンプルコード： String version = AsReaderP252BManager.getVersion();</p>			

2.3 AsReaderP252BEventListener

2.3.1 onStateChanged

関数	void onStateChanged(AsReaderP252B reader, AsReaderP252BConnectionState state)		
引数	IN/OUT	型	説明
reader	OUT	AsReaderP252B	AsReaderP252B クラスのインスタンスオブジェクト
state	OUT	AsReaderP252BConnectionState	接続状態 列挙型 AsReaderP252BConnectionState (2.11.1 を参照)
<p>■ 説明 :</p> <p>このメソッドは AsReader 接続状態が変化した時にコールバックされます。 connectDevice メソッド (2.1.2 を参照)、disconnectDevice メソッド (2.1.3 を参照) を呼び出した後、このメソッドがコールバックされ、接続状態を受信します。</p> <p>■ サンプルコード :</p> <pre>@Override public void onStateChanged(AsReaderP252B reader, AsReaderP252BConnectionState state) { //reader : AsReaderP252B オブジェクト // state : AsReader の接続状態 }</pre>			

2.3.2 onActionChanged

関数	void onActionChanged(AsReaderP252B reader, AsReaderP252BActionState action)		
引数	IN/OUT	型	説明
reader	OUT	AsReaderP252B	AsReaderP252B クラスのインスタンスオブジェクト
action	OUT	AsReaderP252BActionState	列挙型 AsReaderP252BActionState (2.11.3 を参照)
<p>■ 説明 :</p> <p>このメソッドは AsReader 操作状態が変化した時にコールバックされます。 inventory メソッド (2.1.13 を参照)、inventoryHumidityTag メソッド (2.1.14 を参照)、 inventoryTemperatureTagWithType メソッド (2.1.15 を参照)、stop メソッド (2.1.22 を参照) を呼び出した後、このメソッドがコールバックされ、実行結果を受信します。</p> <p>■ サンプルコード :</p>			

```
@Override
public onActionChanged(AsReaderP252B reader, AsReaderP252BActionState action) {
    //reader : AsReaderP252B オブジェクト
    //action : AsReaderP252B 現在の操作状態
}
```

2.3.3 onReadTag

関数	void onReadTag(AsReaderP252B reader, AsReaderP252BActionState action, String tag, float rssi, float phase, float frequency)		
引数	IN/OUT	型	説明
reader	OUT	AsReaderP252B	AsReaderP252B クラスのインスタンスオブジェクト
action	OUT	AsReaderP252BActionState	列挙型 AsReaderP252BActionState (2.11.3 を参照)
Tag	OUT	String	タグの PCEPC 値 (16 進数の文字列)
Rssi	OUT	float	タグの RSSI 値
Phase	OUT	float	タグの phase 値
frequency	OUT	float	タグの周波数値

■ 説明 :

このメソッドは RFID タグを読み取った時に、コールバックされます。

inventory メソッド (2.1.13 を参照) を呼び出す、または Trigger キーを押下し、タグを読み取った時に、このメソッドがコールバックされ、読み取ったデータを受信します。

■ サンプルコード :

```
@Override
public void onReadTag(AsReaderP252B reader, AsReaderP252BActionState action, String tag, float rssi, float phase, float frequency) {
    //reader : AsReaderP252B オブジェクト
    //action : AsReaderP252B 現在の操作状態
    //tag: タグの 16 進数の文字列
    //rssi: タグの RSSI 値
    //phase: タグの phase 値
    //frequency: タグの frequency 値
}
```

2.3.4 onAccessResult

関数	void onAccessResult(AsReaderP252B reader, AsReaderP252BResultCode code, AsReaderP252BActionState action, String epc, String data, float rssi, float phase, float frequency)		
引数	IN/OUT	型	説明
Reader	OUT	AsReaderP252B	AsReaderP252B クラスのインスタンスオブジェクト
Code	OUT	AsReaderP252BResultCode	メソッドの実行結果 列挙型 AsReaderP252BResultCode (2.11.8 を参照)
action	OUT	AsReaderP252BActionState	列挙型 AsReaderP252BActionState (2.11.3 を参照)
Epc	OUT	String	タグの EPC 値 (16 進数)
Data	OUT	String	読み取ったタグデータ (16 進数)
Rssi	OUT	float	タグの RSSI 値
phase	OUT	float	タグの phase 値
frequency	OUT	float	タグの周波数値
<p>■ 説明 :</p> <p>このメソッドはタグに対する読み取り、書き込み、ロック、キルなどの操作をした後、コールバックされます。 readMemory メソッド (2.1.16 を参照)、writeMemory メソッド (2.1.17 を参照)、lock メソッド (2.1.18 を参照)、unlock メソッド (2.1.19 を参照)、permaLock メソッド (2.1.20 を参照)、kill メソッド (2.1.21 を参照) を呼び出した後、このメソッドがコールバックされ、実行結果を受信します。</p> <p>■ サンプルコード :</p> <pre>@Override public void onAccessResult(AsReaderP252B reader, AsReaderP252BResultCode code, AsReaderP252BActionState action, String epc, String data, float rssi, float phase, float frequency) { //reader : AsReader オブジェクト //code: 実行の結果 //action: AsReaderP252B 現在の操作状態 //epc: タグの EPC 値 (16 進数) //data: 読み取ったタグデータ (16 進数) //rssi: タグの RSSI 値 //phase: タグの phase 値 //frequency: タグの周波数値 }</pre>			

2.3.5 onReadTemperatureTag

関数	void onReadTemperatureTag(AsReaderP252B reader, AsReaderP252BActionState action, String tag, float rssi, float phase, float frequency, float temperature)		
引数	IN/OUT	型	説明
reader	OUT	AsReaderP252B	AsReaderP252B クラスのインスタンスオブジェクト
action	OUT	AsReaderP252BActionState	列挙型 AsReaderP252BActionState (2.11.3 を参照)
tag	OUT	String	タグの PCEPC 値 (16 進数の文字列)
rssi	OUT	float	タグの RSSI 値
phase	OUT	float	タグの phase 値
frequency	OUT	float	タグの周波数値
temperature	OUT	float	温度
<p>■ 説明 :</p> <p>温度タグデータを受信します。 inventoryTemperatureTagWithType メソッド (2.1.15 を参照) を呼び出す。または Trigger キーを押下し、タグを読み取った時に、このメソッドがコールバックされ、読み取ったデータを受信します。</p> <p>■ サンプルコード :</p> <pre>@Override public onReadTemperatureTag(AsReaderP252B reader, AsReaderP252BActionState action, String tag, float rssi, float phase, float frequency, float temperature) { //reader : AsReader オブジェクト //action : AsReader 現在の操作状態 //tag: タグの 16 進数の文字列 //rssi: タグの RSSI 値 //phase: タグの phase 値 //frequency: タグの frequency 値 //temperature 温度 }</pre>			

2.3.6 onReadHumidityTag

関数	void onReadHumidityTag(AsReaderP252B reader, AsReaderP252BActionState action, String tag, float rssi, float phase, float frequency, float humidity)		
引数	IN/OUT	型	説明
reader	OUT	AsReaderP252B	AsReaderP252B クラスのインスタンスオブジェクト

action	OUT	AsReaderP252BActionState	列挙型 AsReaderP252BActionState (2.11.3 を参照)
tag	OUT	String	タグの PCEPC 値 (16 進数の文字列)
rsi	OUT	float	タグの RSSI 値
phase	OUT	float	タグの phase 値
frequency	OUT	float	タグの周波数値
humidity	OUT	float	湿度

■説明：
 湿度タグデータを受信します。
 inventoryHumidityTag メソッド (2.1.14 を参照) を呼び出す。または Trigger キーを押下し、タグを読み取った時に、このメソッドがコールバックされ、読み取ったデータを受信します。

■サンプルコード：

```

@Override
public onReadHumidityTag(AsReaderP252B reader, AsReaderP252BActionState action,
String tag, float rssi, float phase, float frequency, float humidity) {
    //reader : AsReader オブジェクト
    //action : AsReader 現在の操作状態
    //tag: タグの 16 進数の文字列
    //rssi: タグの RSSI 値
    //phase: タグの phase 値
    //frequency: タグの frequency 値
    //humidity: 湿度
}
        
```

2.3.7 onKeyEvent

関数	boolean onKeyEvent(AsReaderP252B reader, AsReaderP252BKeyState state)		
引数	IN/OUT	型	説明
reader	OUT	AsReader	AsReader クラスのインスタンスオブジェクト
state	OUT	AsReaderP252BKeyState	AsReaderP252B の Trigger キーのステータス 列挙型 AsReaderP252BKeyState (2.11.2 を参照)
	IN	boolean	YES : SDK のデフォルト操作を実行します。 Trigger キーを押下し、スキャンを開始する Trigger キーを離し、スキャンを停止する NO : SDK のデフォルト操作をしません。

■ 説明 :

このメソッドは AsReader の Trigger キーが押下または離れた時に、コールバックされます。

■ サンプルコード :

```
@Override
public boolean onKeyEvent(AsReaderP252B reader, AsReaderP252BKeyState state) {
    //reader : AsReaderP252B オブジェクト
    //state : キーのステータス
}
```

2.3.8 onModeKeyEvent

関数	boolean onModeKeyEvent(AsReaderP252B reader, AsReaderP252BKeyState state)		
引数	IN/OUT	型	説明
reader	OUT	AsReaderP252B	AsReaderP252B クラスのインスタンスオブジェクト
state	OUT	AsReaderP252BKeyState	AsReaderP252B の Mode キーのステータス 列挙型 AsReaderP252BKeyState (2.11.2 を参照)
	IN	boolean	YES : Barcode/RFID モードを切り替える NO : 現在モードを維持する

■ 説明 :

このメソッドは AsReader の Mode キーが押下または離れた時に、コールバックされます。

■ サンプルコード :

```
@Override
public boolean onModeKeyEvent(AsReaderP252B reader, AsReaderP252BKeyState state){
    //reader : AsReaderP252B オブジェクト
    //state : キーのステータス
}
```

2.3.9 onReadBarcode

関数	void onReadBarcode(AsReaderP252B reader, AsReaderP252BBarcodeType type, byte[] barcodeData)		
引数	IN/OUT	型	説明
Reader	OUT	AsReaderP252B	AsReader クラスのインスタンスオブジェクト

Type	OUT	AsReaderP252BBarcodeType	バーコードのタイプ 列挙型 AsReaderP252BBarcodeType (2.11.11 を参照)
barcodeData	OUT	byte[]	バーコードデータ

■説明：
 スキャンされたバーコードデータを受信します。
 startDecode メソッド (2.1.69 を参照) を呼び出す、または Trigger キーを押下し、タグを読み取った時に、このメソッドがコールバックされ、読み取ったデータを受信します。

■サンプルコード：

```

@Override
void onReadBarcode(AsReaderP252B reader, AsReaderP252BBarcodeType type, byte[] barcodeData) {
    //reader : AsReaderP252B オブジェクト
    // type: バーコードのタイプ
    // barcodeData: バーコードデータ
    //String string = new String(barcodeData, Charset.forName("ASCII"));
}
        
```

2.3.10 onReceivedData

関数	void onReceivedData(byte[] data)		
引数	IN/OUT	型	説明
data	OUT	byte[]	コマンド

■説明：
 AsReader から返されたコマンドを受信します。

■サンプルコード：

```

@Override
void onReceivedData(byte[] data){
    // data : コマンド
}
        
```

2.4 AsReaderP252BBluetoothDiscoveryEventListener

2.4.1 onReceivedDevice

関数	void onReceivedDevice(BluetoothDevice device);		
引数	IN/OUT	型	説明

device	OUT	BluetoothDevice	Bluetooth デバイス
<p>■ 説明 :</p> <p>検索された Bluetooth デバイスを受信します。 startDiscovery メソッド (2.6.2 を参照) を呼び出した後、このメソッドがコールバックされ、検索された Bluetooth デバイスを受信します。</p> <p>■ サンプルコード :</p> <pre>@Override public void onReceivedDevice(final BluetoothDevice device) { //device : Bluetooth オブジェクト }</pre>			

2.4.2 onFoundDeviceFinished

関数	void onFoundDeviceFinished();
<p>■ 説明 :</p> <p>Bluetooth デバイスの検索を停止します。 stopDiscovery メソッド (2.6.3 を参照) を呼び出す、または 15 秒以内に Bluetooth デバイスを検索できない時、このメソッドがコールバックされ、検索停止のステータスを受信します。</p> <p>■ サンプルコード :</p> <pre>@Override public void onFoundDeviceFinished() { }</pre>	

2.5 AsReaderP252BDeviceUsbCdc

2.5.1 AsReaderP252BDeviceUsbCdc

関数	public AsReaderP252BDeviceUsbCdc(Context context)		
引数	IN/OUT	型	説明
context	IN	Context	AsReaderP252BDeviceUsbCdc オブジェクト
<p>■ 説明 :</p> <p>AsReaderP252BDeviceUsbCdc クラスのインスタンスを作成します。</p> <p>■ サンプルコード :</p> <pre>AsReaderP252BDeviceUsbCdc usbCdc = new AsReaderP252BDeviceUsbCdc(this);</pre>			

2.6 AsReaderP252BDeviceBluetoothCdc

2.6.1 AsReaderP252BDeviceBluetoothCdc

関数	public AsReaderP252BDeviceBluetoothCdc(Context context)		
引数	IN/OUT	型	説明
context	IN	Context	AsReaderP252BDeviceBluetoothCdc オブジェクト
<p>■ 説明 :</p> <p>AsReaderP252BDeviceBluetoothCdc クラスのインスタンスを作成します。</p> <p>■ サンプルコード :</p> <pre>AsReaderP252BDeviceBluetoothCdc bluetoothCdc = new AsReaderP252BDeviceBluetoothCdc(this);</pre>			

2.6.2 startDiscovery

関数	public void startDiscovery()
<p>■ 説明 :</p> <p>Bluetooth デバイスの検索を開始します。 このメソッドを呼び出した後、onReceivedDevice (2.4.1 を参照) がコールバックされ、検索された Bluetooth デバイスを受信します。</p> <p>■ サンプルコード :</p> <pre>bluetoothCdc.startDiscovery();</pre>	

2.6.3 stopDiscovery

関数	public void stopDiscovery()
<p>■ 説明 :</p> <p>Bluetooth デバイスの検索を停止します。 このメソッドを呼び出した後、onFoundDeviceFinished (2.4.2 を参照) がコールバックされ、検索停止のステータスを受信します。</p> <p>■ サンプルコード :</p> <pre>bluetoothCdc.stopDiscovery();</pre>	

2.7 AsReaderP252BLbtItem

2.7.1 getSlot

関数	public int getSlot()		
引数	IN/OUT	型	説明
	OUT	int	LBT 周波数リストの周波数位置
<p>■ 説明 :</p> <p>LBT 周波数リストの周波数のインデックスを取得します。</p> <p>■ サンプルコード :</p> <pre>int mSlot = item.getSlot();</pre>			

2.7.2 isUsed

関数	public boolean isUsed()		
引数	IN/OUT	型	説明
	OUT	boolean	true: 有効 false: 無効
<p>■ 説明 :</p> <p>LBT が有効かどうかを取得します。</p> <p>■ サンプルコード :</p> <pre>boolean mIsUsed = item.isUsed();</pre>			

2.7.3 setUsed

関数	public void setUsed(boolean used)		
引数	IN/OUT	型	説明
used	IN	boolean	true: 有効 false: 無効
<p>■ 説明 :</p> <p>LBT を有効にするかどうかを設定します。</p> <p>■ サンプルコード :</p> <pre>item.setUsed(true);</pre>			

2.7.4 getFrequency

関数	public String getFrequency()		
引数	IN/OUT	型	説明
	OUT	String	LBT 有効時の周波数値
<p>■ 説明 :</p> <p>LBT 有効時の周波数値を取得します。</p> <p>■ サンプルコード :</p> <pre>String mFrequency= item.getFrequency();</pre>			

2.8 AsReaderP252BLockParam

2.8.1 AsReaderP252BLockParam

関数	public AsReaderP252BLockParam(boolean killPassword, boolean accessPassword, boolean epc, boolean tid, boolean user)		
引数	IN/OUT	型	説明
killPassword	IN	boolean	true: キルパスワードをロックする false: キルパスワードをロックしない
accessPassword	IN	boolean	true: アクセスパスワードをロックする false: アクセスパスワードをロックしない
epc	IN	boolean	true: epc をロックする false: epc をアンロックしない
tid	IN	boolean	true: tid をロックする false: tid をアンロックする
user	IN	boolean	true: ユーザーをロックする false: ユーザーをロックしない
	OUT	AsReaderP252BLockParam	AsReaderP252BLockParam オブジェクト
<p>■ 説明 :</p> <p>AsReaderP252BLockParam クラスのインスタンスを作成します。</p> <p>■ サンプルコード :</p> <pre>AsReaderP252BLockParam param = new AsReaderP252BLockParam(true, true, true, true, true);</pre>			

2.9 AsReaderP252BPowerRange

2.9.1 getMin

関数	public int getMin()		
引数	IN/OUT	型	説明
	OUT	int	電波出力の最小値
■ 説明： 電波出力の最小値を取得します。			
■ サンプルコード： <pre>int min = powerGainRange.min;</pre>			

2.9.2 getMax

関数	public int getMax()		
引数	IN/OUT	型	説明
	OUT	int	電波出力の最大値
■ 説明： 電波出力の最大値を取得します。			
■ サンプルコード： <pre>int max = powerGainRange.max;</pre>			

2.10 AsReaderP252BSelectMaskEpcParam

2.10.1 AsReaderP252BSelectMaskEpcParam

関数	public AsReaderP252BSelectMaskEpcParam()		
引数	IN/OUT	型	説明
	OUT	AsReaderP252BSelectMaskEpcParam	AsReaderP252BSelectMaskEpcParam オブジェクト
<p>■ 説明 :</p> <p>AsReaderP252BSelectMaskEpcParam クラスのインスタンスを作成します。</p> <p>■ サンプルコード :</p> <pre>AsReaderP252BSelectMaskEpcParam selectMaskEpcParam = new AsReaderP252BSelectMaskEpcParam();</pre>			

2.10.2 getOffset

関数	public int getOffset()		
引数	IN/OUT	型	説明
	OUT	int	マスクのオフセット
<p>■ 説明 :</p> <p>現在のマスクのオフセットを取得します。</p> <p>■ サンプルコード :</p> <pre>int mOffset = selectMaskEpcParam.getOffset();</pre>			

2.10.3 setOffset

関数	public void setOffset(int offset)		
引数	IN/OUT	型	説明
offset	IN	int	マスクのオフセット
<p>■ 説明 :</p> <p>現在のマスクのオフセットを設定します。</p> <p>■ サンプルコード :</p> <pre>selectMaskEpcParam.setOffset(1);</pre>			

2.10.4 getLength

関数	public int getLength()		
----	------------------------	--	--

引数	IN/OUT	型	説明
	OUT	int	マスクの長さ
<p>■ 説明 :</p> <p>現在のマスクの長さを取得します。</p> <p>■ サンプルコード :</p> <pre>int mLength = selectMaskEpcParam.getLength();</pre>			

2.10.5 setLength

関数	public void setLength(int length)		
引数	IN/OUT	型	説明
length	IN	int	マスクの長さ
<p>■ 説明 :</p> <p>現在のマスクの長さを設定します。</p> <p>■ サンプルコード :</p> <pre>selectMaskEpcParam.setLength(1);</pre>			

2.10.6 getMask

関数	public String getMask()		
引数	IN/OUT	型	説明
	OUT	String	マスク文字列
<p>■ 説明 :</p> <p>現在のマスクを取得します。(16 進数)</p> <p>■ サンプルコード :</p> <pre>String mMask = selectMaskEpcParam.getMask();</pre>			

2.10.7 setMask

関数	public void setMask(String mask)		
引数	IN/OUT	型	説明
mask	IN	String	マスク文字列
<p>■ 説明 :</p> <p>現在のマスクを設定します。(16 進数)</p> <p>■ サンプルコード :</p> <pre>selectMaskEpcParam.setMask("1111");</pre>			

2.11 Enum

2.11.1 AsReaderP252BConnectionState

定義	説明
Disconnected = 0	接続切断
Connecting = 2	接続中
Connected = 3	接続済み
Cancelling = 4	接続取消中

2.11.2 AsReaderP252BKeyState

定義	説明
KeyUp = 0	Trigger キー押下解除
KeyDown = 1	Trigger キー押下

2.11.3 AsReaderP252BActionState

定義	説明
Stop = 0x73	操作停止
Inventory = 0x66	インベントリ中
ReadMemory = 0x72	メモリ読取中
WriteMemory = 0x77	メモリ書き込み中
Lock = 0x6C	ロック操作中
Unlock = 0x6D	アンロック操作中
PermaLock = 0x70	永久ロック操作中
Kill = 0x6B	タグキル中
StartDecode = 0x64	バーコード読取中
StartBuzzer = 0x75	ブザー開始中
StartVibrator = 0x76	振動開始中
WaitForResponse = 0xF0	レスポンス待ち

2.11.4 AsReaderP252BBuzzerState

定義	説明
Off = 0	ブザー音 なし
Low = 1	ブザー音 低
High = 2	ブザー音 高

2.11.5 AsReaderP252BMemoryBank

定義	説明
Reserved=0	Reserved エリア
EPC = 1	EPC エリア
TID = 2	TID エリア
User = 3	User エリア

2.11.6 AsReaderP252BQuerySession

定義	説明
S0=0	Inventoried S0
S1=1	inventoried S1
S2=2	inventoried S2
S3=3	inventoried S3

2.11.7 AsReaderP252BSessionFlag

定義	説明
A=0	A only
B=1	B only
AB=2	A or B

2.11.8 AsReaderP252BResultCode

定義	説明
NoError = 0x0000	成功
OtherError = 0x0001	未知なエラー
Undefined = 0x0002	未定義なエラー
MemoryOverrun = 0x0003	アクセスするメモリーは範囲を超えた
MemoryLocked = 0x0004	タグがロックされた
InsufficientPower = 0x000B	バッテリー不足
NonSpecificError = 0x000F	未定義なエラー
InOperation = 0xE000	動作中
NotConnected = 0xE100	AsReader 未接続
InvalidParameter = 0xE200	無効なパラメータが送信された
NotSupportFirmware = 0xEE00	サポートしていないファームウェア
Timeout = 0xEFFF	タイムアウト
HandleMismatch = 0xF001	ハンドラーが不一致

CRCError = 0xF002	CRC エラー
CommandFormatError = 0xF007	コマンドフォーマットエラー
OutOfRetries = 0xF009	リトライ失敗
OperationFailed = 0xFFFF	オペレーション失敗

2.11.9 AsReaderP252BScanMode

定義	説明
RFIDScanMode = 0	RFID モード
BarcodeScanMode = 1	Barcode モード

2.11.10 AsReaderP252BTemperatureTagType

定義	説明
Type_0= 0	温度タグタイプ Magnus-S3

2.11.11 AsReaderP252BBarcodeType

定義	説明
BarcodeTypeNoRead	UNKNOWNTYPE
BarcodeTypeCode39	Code39
BarcodeTypeCode11	Code11
BarcodeTypeCodabar	Codabar
BarcodeTypeEAN13	EAN-13
BarcodeTypeCode128	Code128
BarcodeTypeEAN13With2Supps	EAN-13 with 2 Supps.
BarcodeTypeIndustrial2Of5	Industrial 2 of 5
BarcodeTypeEAN13With5Supps	EAN-13 with 5 Supps.
BarcodeTypeIATA2Of5	IATA 2 of 5
BarcodeTypeMSI	MSI
BarcodeTypeInterleaved2Of5	Interleaved 2 of 5
BarcodeTypeEAN128	EAN-128
BarcodeTypeCode93	Code93
BarcodeTypeUPCE1	UPC-E1
BarcodeTypeUPCA	UPC-A
BarcodeTypeUPCE1With2Supps	UPC-E1 with 2 Supps.
BarcodeTypeUPCE1With5Supps	UPC-E1 with 5 Supps.
BarcodeTypeUPCAWith2Supps	UPC-A with 2 Supps.

BarcodeTypeTriopticCode39	Trioptic Code39
BarcodeTypeUPCE0	UPC-E
BarcodeTypeBooklandEAN	Bookland EAN
BarcodeTypeUPCE0With2Supps	UPC-E with 2 Supps.
BarcodeTypeCouponCode	Coupon Code
BarcodeTypeUPCE0With5Supps	UPC-E with 5 Supps.
BarcodeTypeGS1DataBarLimitedRSS Limited	GS1 DataBar Limited (RSS-Limited)
BarcodeTypeEAN8	EAN-8
BarcodeTypeGS1DataBarRSS14	GS1 DataBar (RSS-14)
BarcodeTypeEAN8With2Supps	EAN-8 with 2 Supps.
BarcodeTypeGS1DataBarExpandedRSSExpanded	GS1 DataBar Expanded (RSS-Expanded)
BarcodeTypeEAN8With5Supps	EAN-8 with 5 Supps.
BarcodeTypeMatrix2Of5	Matrix 2 of 5
BarcodeTypeChinaPostChinese2Of5	China Post (Chinese 2 of 5)
BarcodeTypeCode32	Code32
BarcodeTypeUKPlessey	UK Plessey
BarcodeTypeISBT128	ISBT128
ParameterFNC3	Parameter (FNC3)
BarcodeTypePDF417	PDF417
BarcodeTypeAztec	Aztec
BarcodeTypeMicroPDF417	MicroPDF417
BarcodeTypeQR	QR
BarcodeTypeDataMatrix	DataMatrix
BarcodeTypeMicroQR	Micro QR
BarcodeTypeHanXinCode	HanXin Code
BarcodeTypeMaxicode	Maxicode
BarcodeTypeITF14	ITF-14
BarcodeTypeITF6	ITF-6
BarcodeTypeAIM128	AIM 128
BarcodeTypeISSN	ISSN
BarcodeTypeISBN	ISBN
BarcodeTypeGS1Databar	GS1-Databar