

**AsReader**<sup>®</sup>

# AsReader ASR-P3xU

**Android SDK 開発マニュアル V1.2**

Asterisk Inc.

2024-01-23

## 修正履歴

バージョン	修正内容	日付
1.0	新規作成	2023/3/29
1.1	スペルミスの修正	2024/1/15
1.2	システム要求の修正	2024/1/23

## 目次

概要 .....	5
1 開発環境の構築 .....	6
1.1 SDK の追加 .....	6
1.2 SDK の使用 .....	10
2 メソッド説明 .....	16
2.1 AsReaderP3xU .....	16
2.1.1 getResultCode .....	16
2.1.2 connectDevice .....	16
2.1.3 disconnectDevice .....	16
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 setEventListener .....	19
2.1.11 removeEventListener .....	19
2.1.12 inventory .....	19
2.1.13 readMemory .....	20
2.1.14 writeMemory .....	20
2.1.15 lock .....	21
2.1.16 unlock .....	22
2.1.17 permaLock .....	22
2.1.18 kill .....	23
2.1.19 stop .....	23
2.1.20 defaultParameter .....	24
2.1.21 getBuzzer .....	24
2.1.22 setBuzzer .....	24
2.1.23 getContinuousMode .....	25
2.1.24 setContinuousMode .....	25
2.1.25 getPowerGain .....	26
2.1.26 setPowerGain .....	26
2.1.27 getPowerGainRange .....	27
2.1.28 getOperationTime .....	27
2.1.29 setOperationTime .....	27
2.1.30 getInventoryTime .....	28
2.1.31 setInventoryTime .....	28

2.1.32	getIdleTime	29
2.1.33	setIdleTime	29
2.1.34	setAccessPassword	29
2.1.35	getQuerySession	30
2.1.36	setQuerySession	30
2.1.37	getSessionFlag	31
2.1.38	setSessionFlag	31
2.1.39	getQValueMin	32
2.1.40	getQValueMax	32
2.1.41	getQValue	32
2.1.42	setQValue	33
2.1.43	getSerialNumber	33
2.1.44	getReportRSSI	34
2.1.45	setReportRSSI	34
2.1.46	clearEpcMask	34
2.1.47	getEpcMaskCount	35
2.1.48	addEpcMask	35
2.1.49	getEpcMask	36
2.1.50	getFrequencyAutomatic	36
2.1.51	setFrequencyAutomatic	37
2.1.52	getLbt	37
2.1.53	setLbt	37
2.1.54	getRegion	38
<b>2.2</b>	<b>AsReaderP3xUEventListener</b>	<b>39</b>
2.2.1	onStateChanged	39
2.2.2	onActionChanged	39
2.2.3	onReadTag	40
2.2.4	onAccessResult	41
2.2.5	onKeyEvent	41
<b>2.3</b>	<b>AsReaderP3xUManager</b>	<b>42</b>
2.3.1	getInstance	42
2.3.2	onDestroy	42
2.3.3	getVersion	42
<b>2.4</b>	<b>AsReaderP3xUDeviceUsbCdc</b>	<b>43</b>
2.4.1	AsReaderP3xUDeviceUsbCdc	43
<b>2.5</b>	<b>AsReaderP3xULbtItem</b>	<b>43</b>
2.5.1	getSlot	43
2.5.2	isUsed	44
2.5.3	setUsed	44
2.5.4	getFrequency	44

<b>2.6 AsReaderP3xULockParam .....</b>	<b>45</b>
2.6.1 AsReaderP3xULockParam .....	45
<b>2.7 AsReaderP3xUPowerRange.....</b>	<b>45</b>
2.7.1 AsReaderP3xUPowerRange .....	45
2.7.2 getMin .....	46
2.7.3 getMax.....	46
<b>2.8 AsReaderP3xUSelectMaskEpcParam .....</b>	<b>46</b>
2.8.1 AsReaderP3xUSelectMaskEpcParam .....	46
2.8.2 getOffset .....	47
2.8.3 setOffset .....	47
2.8.4 getLength.....	47
2.8.5 setLength .....	47
2.8.6 getMask .....	48
2.8.7 setMask.....	48
<b>2.9 Enum .....</b>	<b>48</b>
2.9.1 AsReaderP3xUMaskTargetType.....	48
2.9.2 AsReaderP3xUSessionFlag .....	49
2.9.3 AsReaderP3xUResultCode .....	49
2.9.4 AsReaderP3xUActionState.....	49
2.9.5 AsReaderP3xUConnectionState.....	50
2.9.6 AsReaderP3xUKeyType.....	50
2.9.7 AsReaderP3xUKeyState .....	50
2.9.8 AsReaderP3xUMemoryBank .....	50
2.9.9 AsReaderP3xUBuzzerState.....	50
2.9.10 AsReaderP3xUQuerySession .....	51

## 概要

本文の主な目的：

- 開発者が AsReaderP3xUSDK 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）

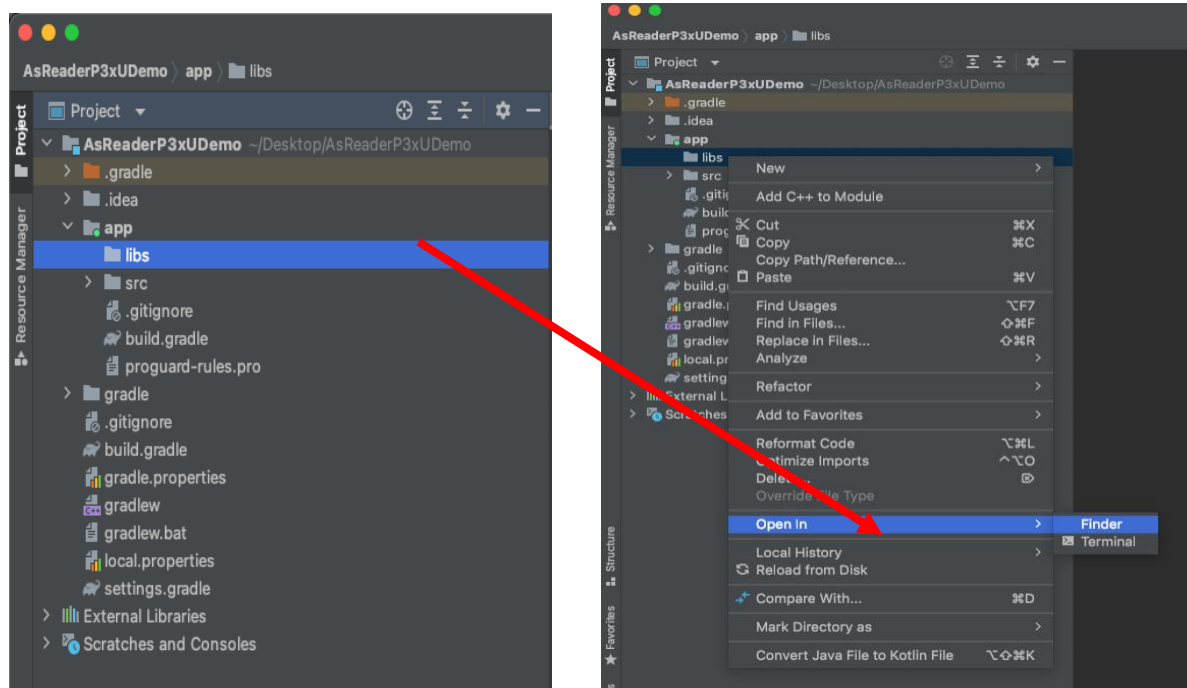


図 1-1-1

2. ポップアップ ウィンドウで「libs」を選択し、「AsReaderP3xUSDk.aar」をこのフォルダに割り当てます。（図 1-1-2）

プロジェクトの「libs」の下に「AsReaderP3xUSDK.aar」が表示されます。(図 1-1-3)

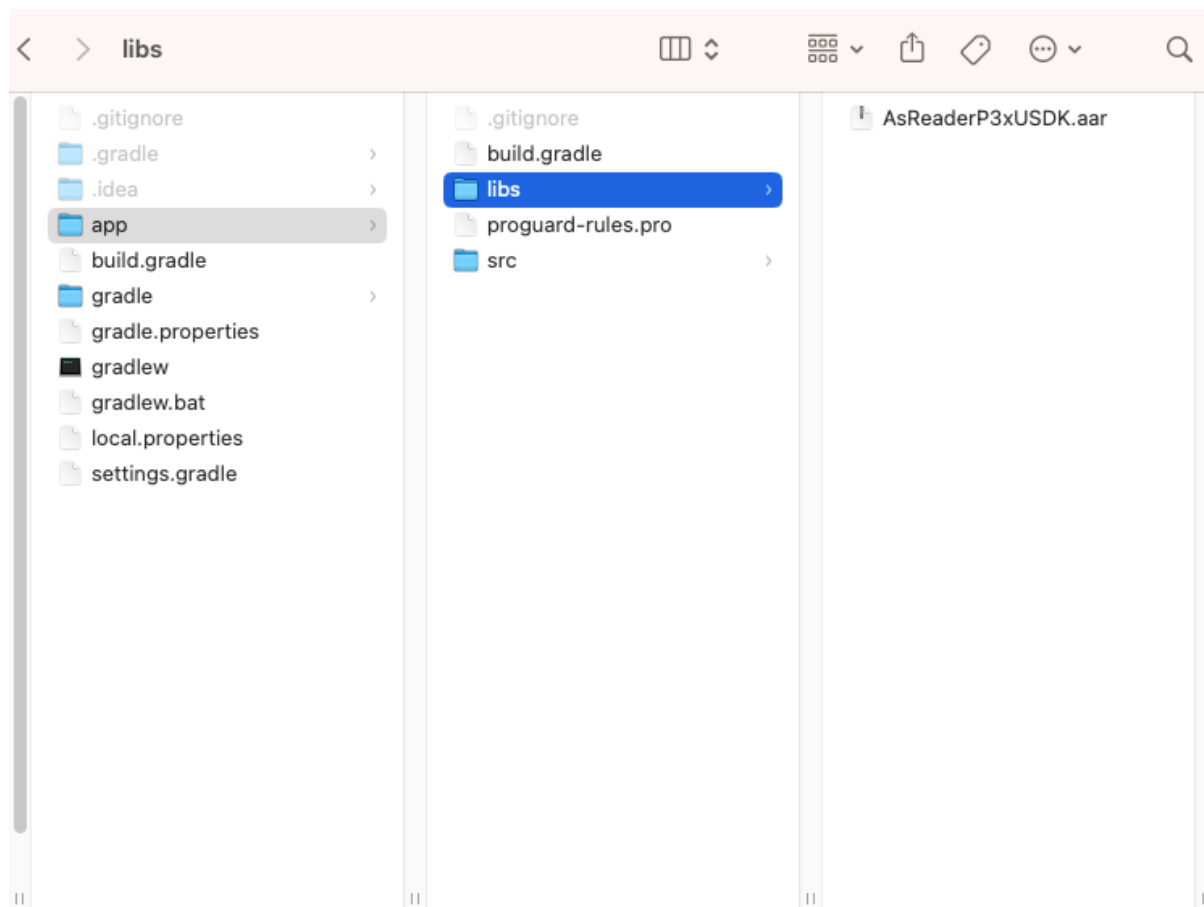


図 1-1-2



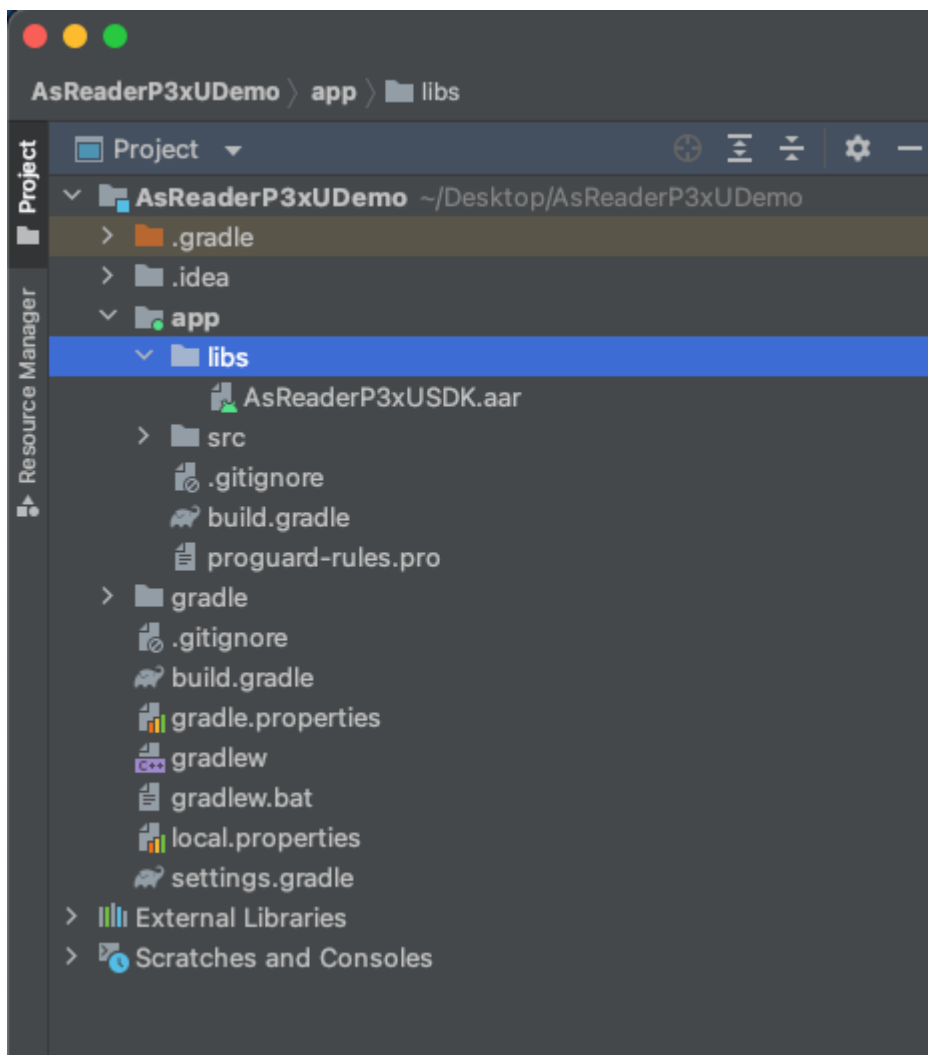


图 1-1-3

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

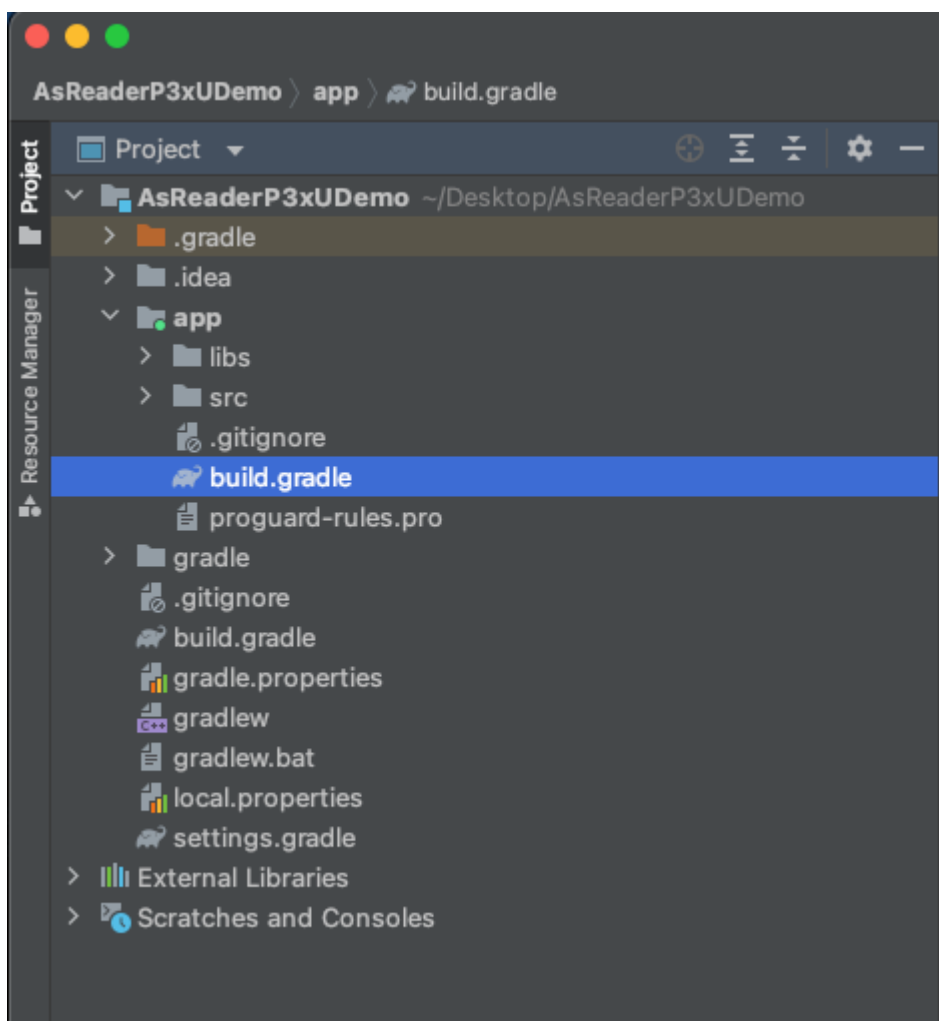


図 1-1-4

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

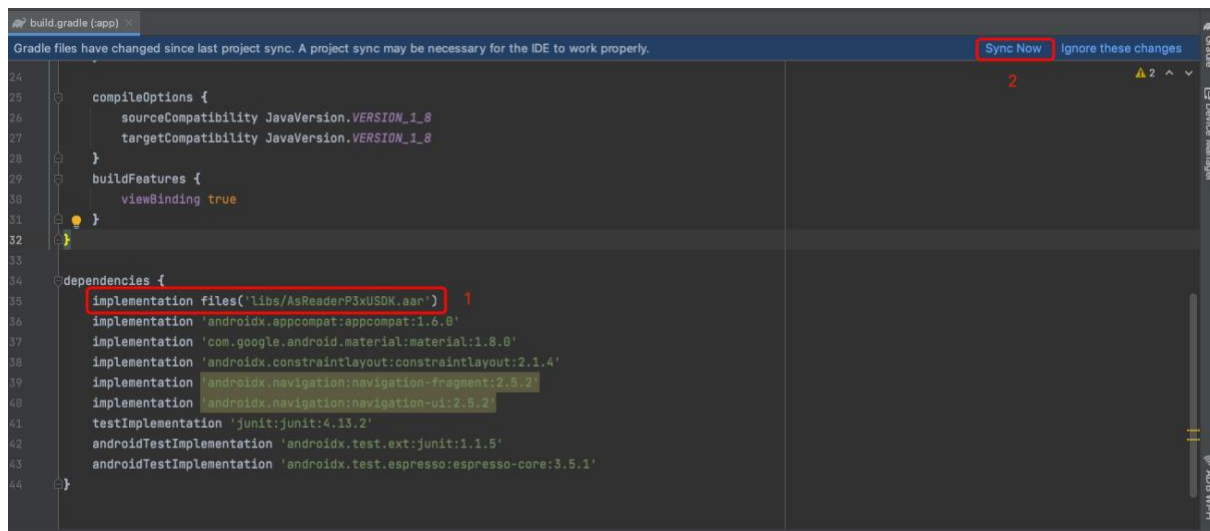


図 1-1-5

5. 同期に成功すると以下の図 1-1-6 のように表示されます。ここまで SDK の追加が完了します。

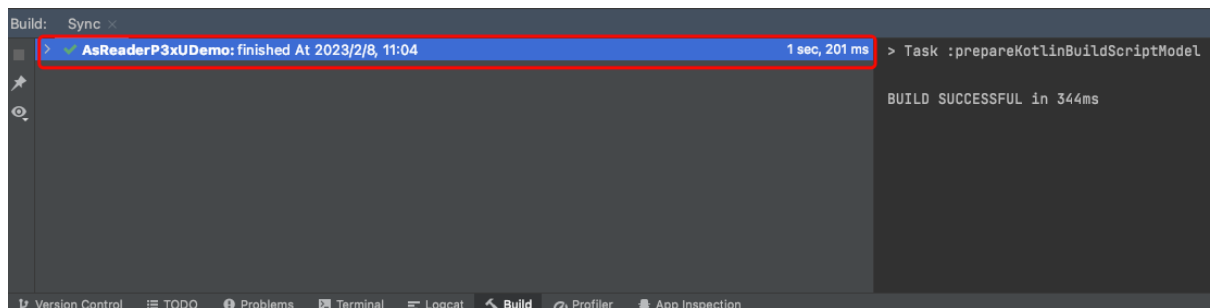


図 1-1-6

## 1.2 SDK の使用

1. 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.p3xu.AsReaderP3xU;  
import com.asreader.p3xu.AsReaderP3xUManager;  
import com.asreader.p3xu.device.AsReaderP3xUDevice;  
import com.asreader.p3xu.device.AsReaderP3xUDeviceUsbCdc;  
import com.asreader.p3xu.rfid.event.AsReaderP3xUEventListener;
```

図 1-2-1

2. 以下の図 1-2-2 の手順で AsReader と USB 接続の処理を実装します。
  - 1) AsReaderP3xUEventListener インターフェースを実装する
  - 2) オブジェクトを宣言する
  - 3) AsReaderP3xU オブジェクトを初期化し、リスナーを設定する
  - 4) USB 接続する

```
public class MainActivity extends AppCompatActivity implements AsReaderP3xUEventListener {  
  
    private AppBarConfiguration appBarConfiguration;  
    private ActivityMainBinding binding;  
  
    private AsReaderP3xU mAsReaderP3xU;  
    private AsReaderP3xUDevice mDevice;  
    private static AsReaderP3xUDeviceUsbCdc usbCdc;  
    private Handler mHandler;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        binding = ActivityMainBinding.inflate(getLayoutInflater());  
        setContentView(binding.getRoot());  
        setSupportActionBar(binding.toolbar);  
        NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment_content_main);  
        appBarConfiguration = new AppBarConfiguration.Builder(navController.getGraph()).build();  
        NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);  
  
        usbCdc = new AsReaderP3xUDeviceUsbCdc(context, this);  
        mAsReaderP3xU = AsReaderP3xUManager.getInstance();  
        if (mAsReaderP3xU == null) {  
            return;  
        }  
        mAsReaderP3xU.setEventListener(this);  
        mHandler = new Handler();  
        mHandler.post(mConnectDevice);  
    }  
  
    private Runnable mConnectDevice = new Runnable() {  
        @Override  
        public void run() {  
            mDevice = mAsReaderP3xU.getCurrentDevice();  
            if (mDevice == null) {  
                mDevice = usbCdc;  
                mAsReaderP3xU.connectDevice(mDevice);  
            }  
        }  
    };  
};
```

図 1-2-2

### 3. 実装した AsReaderP3xUEventListener 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 1-2-3）

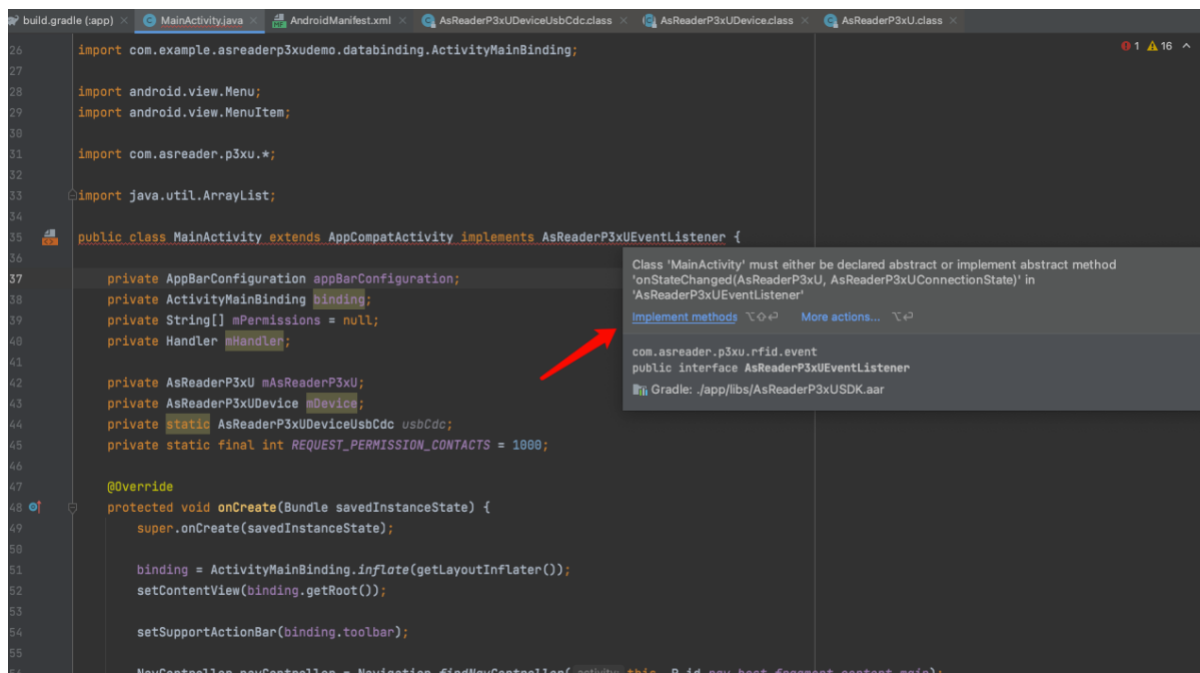


図 1-2-3

4. 青色背景部分を選択し、「OK」をクリックします。(図 1-2-4)

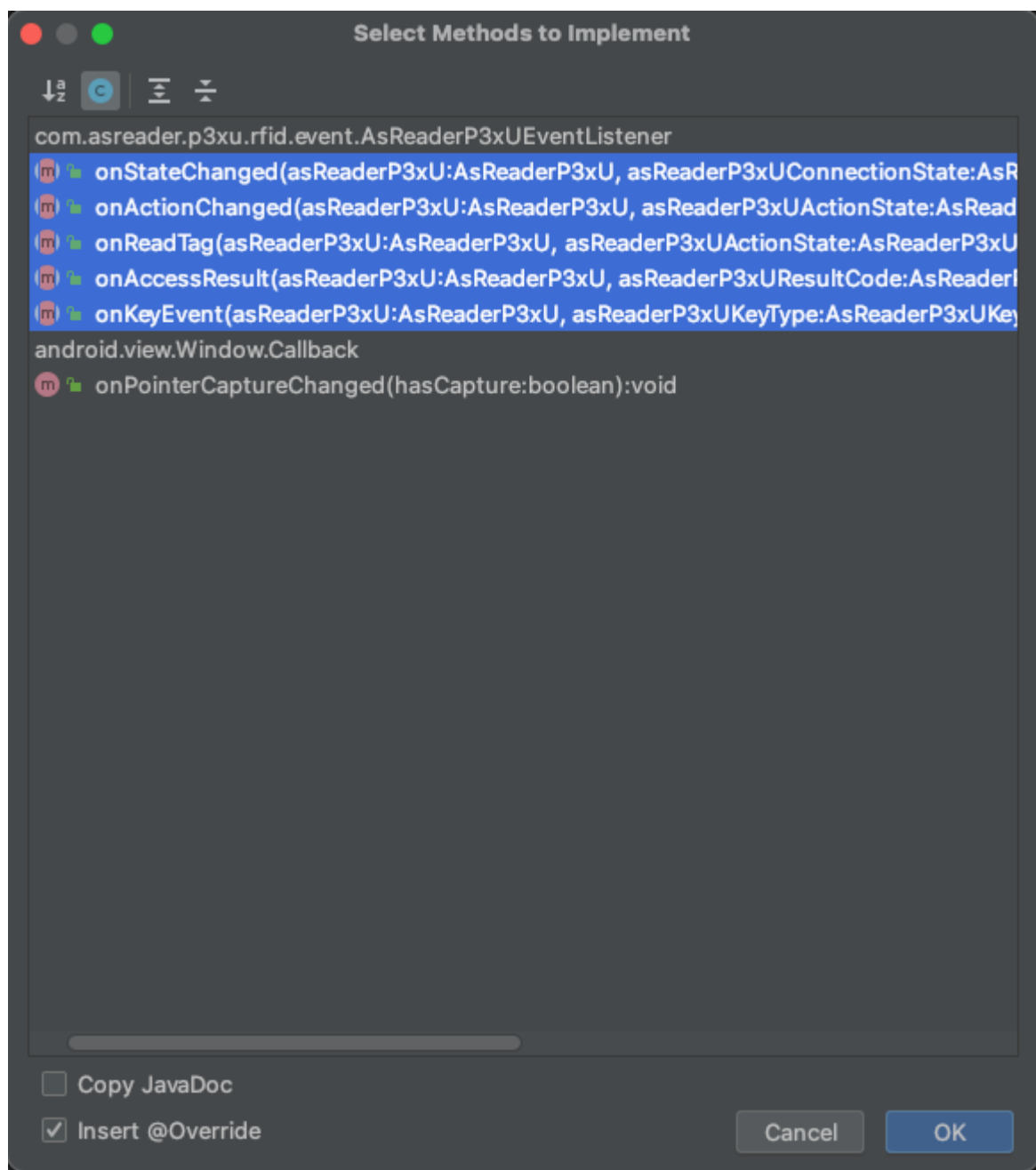


図 1-2-4

5. インターフェイス AsReaderP3xUEventListener のメソッドが自動的に作成されます。  
(図 1-2-5)

```
@Override
public void onStateChanged(AsReaderP3xU asReaderP3xU, AsReaderP3xUConnectionState asReaderP3xUConnectionState) {
}

@Override
public void onActionChanged(AsReaderP3xU asReaderP3xU, AsReaderP3xUActionState asReaderP3xUActionState) {
}

@Override
public void onReadTag(AsReaderP3xU asReaderP3xU, AsReaderP3xUActionState asReaderP3xUActionState, String s, float v) {
}

@Override
public void onAccessResult(AsReaderP3xU asReaderP3xU, AsReaderP3xUResultCode asReaderP3xUResultCode, AsReaderP3xUActionState asReaderP3xUActionState) {
}

@Override
public void onKeyEvent(AsReaderP3xU asReaderP3xU, AsReaderP3xUKeyType asReaderP3xUKeyType, AsReaderP3xUKeyState asReaderP3xUKeyState) {
}
```

図 1-2-5

6. onStateChanged(AsReaderP3xU asReaderP3xU, AsReaderP3xUConnectionState asReaderP3xUConnectionState)メソッドを使って AsReader 接続状態をコールバックします。(図 1-2-6) 戻り値として「Conncted」が返されたら AsReader 接続は成功です。

```
@Override
public void onStateChanged(AsReaderP3xU asReaderP3xU, AsReaderP3xUConnectionState asReaderP3xUConnectionState) {
    switch (asReaderP3xUConnectionState) {
        case Disconnected:
            break;
        case Connecting:
        case Listen:
            break;
        case Connected:
            break;
        case Cancelling:
            break;
    }
}
```

図 1-2-6

7. 接続に成功すると SDK で提供されている API メソッドを呼び出すことができます。以下は inventory ()メソッドを例とします。(図 1-2-7)

```
mAsReaderP3xU.inventory();
```

図 1-2-7

8. inventory ()メソッドの呼び出しに成功すると onReadTag メソッドを使って、読み取ったタグデータをコールバックします。(図 1-2-8)

```
@Override  
public void onReadTag(AsReaderP3xU asReaderP3xU, AsReaderP3xUActionState asReaderP3xUActionState, String s, float v) {  
  
}
```

図 1-2-8



## 2 メソッド説明

### 2.1 AsReaderP3xU

#### 2.1.1 getResultCode

関数	public AsReaderP3xUResultCode getResultCode()		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUResultCode	列挙型 AsReaderP3xUResultCode ( <a href="#">2.9.3</a> 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>AsReaderP3xUResultCode resultcode = asReaderP3xU.getResultCode(); if (resultcode == AsReaderP3xUResultCode.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(AsReaderP3xUDevice device)		
引数	IN/OUT	型	説明
device	IN	AsReaderP3xUDevice	AsReaderP3xUDevice 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 (<a href="#">2.2.1</a>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>asReaderP3xU.connectDevice(mDevice);</pre>			

#### 2.1.3 disconnectDevice

関数	public void disconnectDevice()		
----	--------------------------------	--	--

## ■ 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 ([2.2.1](#) を参照) がコールバックされ、接続状態を取得します。

## ■ サンプルコード :

```
asReaderP3xU.disconnectDevice();
```

### 2.1.4 getCurrentDevice

関数	public AsReaderP3xUDevice getCurrentDevice()		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUDevice	AsReaderP3xUDevice オブジェクト。 (注 : AsReader と接続していない場合、null を返す)
■ 説明 : 現在接続している AsReaderP3xUDevice オブジェクトを取得します。			
■ サンプルコード : <pre>AsReaderP3xUDevice device = asReaderP3xU.getCurrentDevice();</pre>			

### 2.1.5 getState

関数	public AsReaderP3xUConnectionState getState()		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUConnectionState	列挙型 AsReaderP3xUConnectionState ( <a href="#">2.9.5</a> を参照)。
■ 説明 : 現在 AsReader の接続状態を取得します。			
■ サンプルコード : <pre>AsReaderP3xUConnectionState connectState = asReaderP3xU.getState();</pre>			

### 2.1.6 getAction

関数	public AsReaderP3xUActionState getAction()		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUActionState	列挙型 AsReaderP3xUActionState ( <a href="#">2.9.4</a> を参照)。

## ■説明：

AsReaderP3xU 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：

```
AsReaderP3xUActionState actionState = asReaderP3xU.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
<p>■説明：</p> <p>現在 AsReader のファームウェアのバージョン情報を取得します。</p> <p>■サンプルコード：</p> <pre>String firmwareVersion = asReaderP3xU.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
<p>■説明：</p> <p>現在 AsReader のハードウェアのバージョン情報を取得します。</p> <p>■サンプルコード：</p> <pre>String hardwareVersion = asReaderP3xU.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
<p>■説明：</p> <p>現在 AsReader の RF モジュールのバージョン情報を取得します。</p> <p>■サンプルコード：</p> <pre>String rFModuleVersion = asReaderP3xU.getRFModuleVersion();</pre>			

### 2.1.10 setEventListener

関数	public void setEventListener(AsReaderP3xUEventListener listener)		
引数	IN/OUT	型	説明
listener	IN	AsReaderP3xUEventListe ner	AsReaderP3xUEventListener リスナーオブジェクト。 AsReaderP3xUEventListener オブジェクト ( <a href="#">2.2</a> を参照)。
<p>■ 説明 :</p> <p>AsReaderP3xUEventListener リスナーを設定します。</p> <p>■ サンプルコード :</p> <pre>asReaderP3xU.setEventListener(this);</pre>			

### 2.1.11 removeEventListener

関数	public void removeEventListener (AsReaderP3xUEventListener listener)		
引数	IN/OUT	型	説明
listener	IN	AsReaderP3xUEventList ener	AsReaderP3xUEventListener リスナーオブジェクト。 AsReaderP3xUEventListener オブジェクト ( <a href="#">2.2</a> を参照)。
<p>■ 説明 :</p> <p>AsReaderP3xUEventListener リスナーを削除します。</p> <p>■ サンプルコード :</p> <pre>asReaderP3xU.removeEventListener(this);</pre>			

### 2.1.12 inventory

関数	public AsReaderP3xUResultCode inventory()		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUResultC ode	メソッドの実行結果。 列挙型 AsReaderP3xUResultCode ( <a href="#">2.9.3</a> を参 照)。
<p>■ 説明 :</p> <p>タグのインベントリを開始します。 このメソッドを呼び出した後、onActionChanged (<a href="#">2.2.2</a>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 (<a href="#">2.2.3</a>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>AsReaderP3xUResultCode resultCode = asReaderP3xU.inventory();</pre>			

```

if (resultcode == AsReaderP3xUResultCode.NoError) {
    //接続成功時の処理。
} else {
    //接続失敗時の処理。
}

```

### 2.1.13 readMemory

関数	public AsReaderP3xUResultCode readMemory(AsReaderP3xUMemoryBank bank、int offset、 int length)		
引数	IN/OUT	型	説明
bank	IN	AsReaderP3xUMemoryBank	タグのメモリバンク。 列挙型 AsReaderP3xUMemoryBank ( <a href="#">2.9.8</a> 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	AsReaderP3xUResultCode	メソッドの実行結果。 列挙型 AsReaderP3xUResultCode ( <a href="#">2.9.3</a> 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 ([2.2.4](#)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 :**  

```

AsReaderP3xUResultCode resultCode = asReaderP3xU.readMemory
(AsReaderP3xUMemoryBank.EPC,2,4);
if (resultcode == AsReaderP3xUResultCode.NoError) {
    //接続成功時の処理。
} else {
    //接続失敗時の処理。
}

```

### 2.1.14 writeMemory

関数	public AsReaderP3xUResultCode writeMemory(AsReaderP3xUMemoryBank bank、int offset、 String data)		
引数	IN/OUT	型	説明
bank	IN	AsReaderP3xUMemoryBank	タグのメモリバンク。 列挙型 AsReaderP3xUMemoryBank ( <a href="#">2.9.8</a> 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 進数の文字列です）。
	OUT	AsReaderP3xUResultCode	メソッドの実行結果。 列挙型 AsReaderP3xUResultCode（ <a href="#">2.9.3</a> を参照）。
<p>■ 説明：</p> <p>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（<a href="#">2.2.4</a>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>AsReaderP3xUResultCode resultCode = asReaderP3xU. writeMemory(AsReaderP3xUMemoryBank.EPC,2, "1234"); if (resultcode == AsReaderP3xUResultCode.NoError) {     //接続成功時の処理。 } else {     //接続失敗時の処理。 }</pre>			

### 2.1.15 lock

関数	public AsReaderP3xUResultCode lock(AsReaderP3xULockParam param)		
引数	IN/OUT	型	説明
param	IN	AsReaderP3xULockParam	AsReaderP3xULockParam オブジェクト（ <a href="#">2.6</a> を参照）。
	OUT	AsReaderP3xUResultCode	メソッドの実行結果。 列挙型 AsReaderP3xUResultCode（ <a href="#">2.9.3</a> 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（<a href="#">2.2.4</a>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>AsReaderP3xUResultCode resultCode = asReaderP3xU.lock(param); if (resultcode == AsReaderP3xUResultCode.NoError) {     //接続成功時の処理。 } else {     //接続失敗時の処理。 }</pre>			

## 2.1.16 unlock

関数	public AsReaderP3xUResultCode unlock(AsReaderP3xULockParam param)		
引数	IN/OUT	型	説明
param	IN	AsReaderP3xULockParam	AsReaderP3xULockParam オブジェクト。
	OUT	AsReaderP3xUResultCode	メソッドの実行結果。 列挙型 AsReaderP3xUResultCode ( <a href="#">2.9.3</a> 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 (<a href="#">2.2.4</a>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>AsReaderP3xUResultCode resultCode = asReaderP3xU.unlock(param); if (resultCode == AsReaderP3xUResultCode.NoError) {     //接続成功時の処理。 } else {     //接続失敗時の処理。 }</pre>			

## 2.1.17 permaLock

関数	public AsReaderP3xUResultCode permaLock(AsReaderP3xULockParam param)		
引数	IN/OUT	型	説明
param	IN	AsReaderP3xULockParam	AsReaderP3xULockParam オブジェクト ( <a href="#">2.6</a> を参照)。
	OUT	AsReaderP3xUResultCode	メソッドの実行結果。 列挙型 AsReaderP3xUResultCode ( <a href="#">2.9.3</a> 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 (<a href="#">2.2.4</a>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>AsReaderP3xUResultCode resultCode = asReaderP3xU.permaLock(param); if (resultCode == AsReaderP3xUResultCode.NoError) {     //接続成功時の処理。 } else {     //接続失敗時の処理。 }</pre>			

## 2.1.18 kill

関数	public AsReaderP3xUResultCode kill(String killPassword)		
引数	IN/OUT	型	説明
killPassword	IN	String	キルパスワード。
	OUT	AsReaderP3xUResultCode	メソッドの実行結果。 列挙型 AsReaderP3xUResultCode ( <a href="#">2.9.3</a> 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 (<a href="#">2.2.4</a>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>AsReaderP3xUResultCode resultCode = asReaderP3xU.kill("00000000"); if (resultCode == AsReaderP3xUResultCode.NoError) {     //接続成功時の処理。 } else {     //接続失敗時の処理。 }</pre>			

## 2.1.19 stop

関数	public AsReaderP3xUResultCode stop()		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUResultCode	メソッドの実行結果。 列挙型 AsReaderP3xUResultCode ( <a href="#">2.9.3</a> 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 (<a href="#">2.2.2</a>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>AsReaderP3xUResultCode resultCode = asReaderP3xU.stop(); if (resultCode == AsReaderP3xUResultCode.NoError) {     //接続成功時の処理。 } else {     //接続失敗時の処理。 }</pre>			



## 2.1.20 defaultParameter

関数	public AsReaderP3xUResultCode defaultParameter()		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUResultCode	メソッドの実行結果。 列挙型 AsReaderP3xUResultCode ( <a href="#">2.9.3</a> を参照)。
<p>■説明： 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：  <pre>AsReaderP3xUResultCode resultCode = asReaderP3xU.defaultParameter(); if (resultCode == AsReaderP3xUResultCode.NoError) {     //接続成功時の処理。 } else {     //接続失敗時の処理。 }</pre> </p>			

## 2.1.21 getBuzzer

関数	public AsReaderP3xUBuzzerState getBuzzer() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUBuzzerState	列挙型 AsReaderP3xUBuzzerState ( <a href="#">2.9.9</a> を参照)。
	-	AsReaderP3xUException	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 {     AsReaderP3xUBuzzerState mBuzzer = asReaderP3xU.getBuzzer(); } catch (AsReaderP3xUException e) { }</pre> </p>			

## 2.1.22 setBuzzer

関数	public void setBuzzer(AsReaderP3xUBuzzerState state) throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	IN	AsReaderP3xUBuzzerState	列挙型 AsReaderP3xUBuzzerState ( <a href="#">2.9.9</a> を参照)。

	-	AsReaderP3xUException	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： ブザーのパラメータを設定します。</p> <p>■サンプルコード：</p> <pre>try {     asReaderP3xU.setBuzzer(AsReaderP3xUBuzzerState.Off); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.23 getContinuousMode

関数	public boolean getContinuousMode() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	boolean	true: 連続読取有効 false: 連続読取無効
	-	AsReaderP3xUException	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： インベントリを連続実行するかどうかを取得します。</p> <p>■サンプルコード：</p> <pre>try {     boolean continuousMode = asReaderP3xU.getContinuousMode(); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.24 setContinuousMode

関数	public void setContinuousMode(boolean enabled) throws AsReaderP3xUException		
引数	IN/OUT	型	説明
enable	IN	boolean	true: 連続読取有効 false: 連続読取無効
	-	AsReaderP3xUException	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： インベントリを連続実行するかどうかを設定します。</p> <p>■サンプルコード：</p> <pre>try {</pre>			

```

asReaderP3xU.setContinuousMode(true);
} catch (AsReaderP3xUException 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をしてください。

### 2.1.25 getPowerGain

関数	public int getPowerGain() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	int	電波出力(単位： dBm × 10)。
	-	AsReaderP3xUException	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 = asReaderP3xU.getPowerGain(); } catch (AsReaderP3xUException e) { } </pre>			

### 2.1.26 setPowerGain

関数	public void setPowerGain(int power) throws AsReaderP3xUException		
引数	IN/OUT	型	説明
power	IN	int	電波出力(単位： dBm × 10)。
	-	AsReaderP3xUException	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 の電波出力を設定します。</p> <p>■ サンプルコード： (注： powerGain は設定した値です。)</p> <pre> try {     asReaderP3xU.setPowerGain(powerGain); } catch (AsReaderP3xUException e) { } </pre>			

### 2.1.27 getPowerGainRange

関数	public AsReaderP3xUPowerRange getPowerGainRange() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUPowerRange	電波出力の設定可能な範囲。 AsReaderP3xUPowerRange オブジェクト ( <a href="#">2.7</a> を参照)。
	-	AsReaderP3xUException	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 {     AsReaderP3xUPowerRange powerGainRange = asReaderP3xU.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 (AsReaderP3xUException e) { }</pre>			

### 2.1.28 getOperationTime

関数	public int getOperationTime() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	int	インベントリ継続時間 (単位: ms)。
	-	AsReaderP3xUException	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 = asReaderP3xU.getOperationTime(); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.29 setOperationTime

関数	public void setOperationTime(int time) throws AsReaderP3xUException		
引数	IN/OUT	型	説明

time	IN	int	インベントリ継続時間（単位：ms）。
	-	AsReaderP3xUException	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： 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：</p> <pre>try {     asReaderP3xU.setOperationTime(200); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.30 getInventoryTime

関数	public int getInventoryTime() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	int	電波出力時間（単位：ms）。
	-	AsReaderP3xUException	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時間を取得します。</p> <p>■サンプルコード：</p> <pre>try {     int mInventoryTime = asReaderP3xU.getInventoryTime(); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.31 setInventoryTime

関数	public void setInventoryTime(int time) throws AsReaderP3xUException		
引数	IN/OUT	型	説明
time	IN	int	電波出力時間（単位：ms）。
	-	AsReaderP3xUException	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時間を設定します。</p> <p>■サンプルコード：</p> <pre>try {     asReaderP3xU.setInventoryTime(200); }</pre>			

```
} catch (AsReaderP3xUException e) {
}
```

### 2.1.32 getIdleTime

関数	public int getIdleTime() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	int	電波出力停止時間（単位：ms）。
	-	AsReaderP3xUException	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：</p> <pre>try {     int mIdleTime = asReaderP3xU.getIdleTime(); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.33 setIdleTime

関数	public void setIdleTime(int time) throws AsReaderP3xUException		
引数	IN/OUT	型	説明
time	IN	int	電波出力停止時間（単位：ms）。
	-	AsReaderP3xUException	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> <pre>try {     asReaderP3xU.setIdleTime(300); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.34 setAccessPassword

関数	public void setAccessPassword(String password) throws AsReaderP3xUException		
引数	IN/OUT	型	説明
password	IN	String	アクセスパスワード。

	-	AsReaderP3xUException	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： 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>■サンプルコード：</p> <pre>try {     asReaderP3xU.setAccessPassword("00000000"); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.35 getQuerySession

関数	public AsReaderP3xUQuerySession getQuerySession() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUQuerySession	列挙型 AsReaderP3xUQuerySession ( <a href="#">2.9.10</a> を参照)。
	-	AsReaderP3xUException	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： インベントリのセッション値を取得します。</p> <p>■サンプルコード：</p> <pre>try {     AsReaderP3xUQuerySession mSession = asReaderP3xU.getQuerySession(); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.36 setQuerySession

関数	public void setQuerySession(AsReaderP3xUQuerySession session) throws AsReaderP3xUException		
引数	IN/OUT	型	説明
session	IN	AsReaderP3xUQuerySession	列挙型 AsReaderP3xUQuerySession ( <a href="#">2.9.10</a> を参照)。
	-	AsReaderP3xUException	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： インベントリのセッション値を設定します。</p>			

## ■ サンプルコード :

```
try {
    asReaderP3xU.setQuerySession(AsReaderP3xUQuerySession.S0);
} catch (AsReaderP3xUException e) {
}
```

### 2.1.37 getSessionFlag

関数	public AsReaderP3xUSessionFlag getSessionFlag() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUSessionFlag	列挙型 AsReaderP3xUSessionFlag ( <a href="#">2.9.2</a> を参照)。
	-	AsReaderP3xUException	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 :

インベントリのセッションフラグを取得します。

## ■ サンプルコード :

```
try {
    AsReaderP3xUSessionFlag mTarget = asReaderP3xU.getSessionFlag();
} catch (AsReaderP3xUException e) {
}
```

### 2.1.38 setSessionFlag

関数	public void setSessionFlag(AsReaderP3xUSessionFlag target) throws AsReaderP3xUException		
引数	IN/OUT	型	説明
target	IN	AsReaderP3xUSessionFlag	列挙型 AsReaderP3xUSessionFlag ( <a href="#">2.9.2</a> を参照)。
	-	AsReaderP3xUException	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 :

インベントリのセッションフラグを設定します。

## ■ サンプルコード :

```
try {
    asReaderP3xU.setSessionFlag(AsReaderP3xUSessionFlag.AB);
} catch (AsReaderP3xUException e) {
}
```



### 2.1.39 getQValueMin

関数	public int getQValueMin() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	int	現在の最小 Q 値。
	-	AsReaderP3xUException	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>インベントリの現在の最小 Q 値を取得します。</p> <p>■ サンプルコード：</p> <pre>try {     int mQValue = asReaderP3xU.getQValueMin(); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.40 getQValueMax

関数	public int getQValueMax() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	int	現在の最大 Q 値。
	-	AsReaderP3xUException	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>インベントリの現在の最大 Q 値を取得します。</p> <p>■ サンプルコード：</p> <pre>try {     int mQValue = asReaderP3xU.getQValueMax(); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.41 getQValue

関数	public int getQValue() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	int	現在の Q 値 値の範囲：0~15。
	-	AsReaderP3xUException	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：  
インベントリの現在の Q 値を取得します。

■サンプルコード：  

```
try {
    int mQValue = asReaderP3xU.getQValue();
} catch (AsReaderP3xUException e) {
}
```

## 2.1.42 setQValue

関数	public void setQValue(int value) throws AsReaderP3xUException		
引数	IN/OUT	型	説明
value	IN	int	現在の Q 値 値の範囲：0~15。
	-	AsReaderP3xUException	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：  
インベントリの現在の Q 値を設定します。

■サンプルコード：  

```
try {
    asReaderP3xU.setQValue(10);
} catch (AsReaderP3xUException e) {
}
```

## 2.1.43 getSerialNumber

関数	public String getSerialNumber() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	String	AsReader の製造番号。
	-	AsReaderP3xUException	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：  
AsReader の製造番号を取得します。

■サンプルコード：  

```
try {
    String mSerialNumber = asReaderP3xU.getSerialNumber();
} catch (AsReaderP3xUException e) {
}
```

### 2.1.44 getReportRSSI

関数	public boolean getReportRSSI() throws AsReaderP3xUException		
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。
	-	AsReaderP3xUException	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。</p> <p>■ サンプルコード :</p> <pre>try {     boolean mReportRSSI = asReaderP3xU.getReportRSSI(); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.45 setReportRSSI

関数	public void setReportRSSI(boolean enabled) throws AsReaderP3xUException		
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。
	-	AsReaderP3xUException	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。</p> <p>■ サンプルコード :</p> <pre>try {     asReaderP3xU.setReportRSSI(true); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.46 clearEpcMask

関数	public void clearEpcMask() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	-	AsReaderP3xUException	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 {
    asReaderP3xU.clearEpcMask();
} catch (AsReaderP3xUException e) {
}
```

### 2.1.47 getEpcMaskCount

関数	public int getEpcMaskCount() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	int	EPC マスクの数量。
		AsReaderP3xUException	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： 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：  <pre>try {     int mCount = asReaderP3xU.getEpcMaskCount(); } catch (AsReaderP3xUException e) { }</pre></p>			

### 2.1.48 addEpcMask

関数	public void addEpcMask(AsReaderP3xUSelectMaskEpcParam param) throws AsReaderP3xUException		
引数	IN/OUT	型	説明
param	IN	AsReaderP3xUSelectMaskEpcParam	AsReaderP3xUSelectMaskEpcParam オブジェクト ( <a href="#">2.8</a> を参照)。
		AsReaderP3xUException	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： AsReaderP3xUSelectMaskEpcParam 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：  <pre>try {     asReaderP3xU.addEpcMask(mask); }</pre></p>			

```
} catch (AsReaderP3xUException e) {
}
```

### 2.1.49 getEpcMask

関数	public AsReaderP3xUSelectMaskEpcParam getEpcMask(int index) throws AsReaderP3xUException		
引数	IN/OUT	型	説明
index	IN	int	EPC マスクのインデックス。
	OUT	AsReaderP3xUSelectMaskEpcParam	メソッドの実行結果。 AsReaderP3xUSelectMaskEpcParam オブジェクト ( <a href="#">2.8</a> を参照)。
		AsReaderP3xUException	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>指定されたインデックスの EPC マスクを取得します。</p> <p>■ サンプルコード :</p> <pre>try {     AsReaderP3xUSelectMaskEpcParam mask = asReaderP3xU.getEpcMask(0); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.50 getFrequencyAutomatic

関数	public boolean getFrequencyAutomatic() throws AsReaderP3xUException		
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。
		AsReaderP3xUException	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> <p>■ サンプルコード :</p> <pre>try {     boolean mFrequencyAutomatic = asReaderP3xU.getFrequencyAutomatic(); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.51 setFrequencyAutomatic

関数	public void setFrequencyAutomatic(boolean isAutomatic) throws AsReaderP3xUException		
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。
		AsReaderP3xUException	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>イベントリで周波数ホッピング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 :</p> <pre>try {     asReaderP3xU. setFrequencyAutomatic(true); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.52 getLbt

関数	public AsReaderP3xULbtItem[] getLbt() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xULbtItem[]	AsReaderP3xULbtItem オブジェクト ( <a href="#">2.5</a> を参照)。
		AsReaderP3xUException	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>LBT の各パラメータ値を取得します。</p> <p>■サンプルコード :</p> <pre>try {     AsReaderP3xULbtItem[] mItems = asReaderP3xU.getLbt(); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.53 setLbt

関数	public void setLbt(AsReaderP3xULbtItem[] table) throws AsReaderP3xUException		
引数	IN/OUT	型	説明

table	IN	AsReaderP3xULbtItem[]	AsReaderP3xULbtItem オブジェクト ( <a href="#">2.5</a> を参照)。
		AsReaderP3xUException	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>LBT の各パラメータ値を設定します。</p> <p>■ サンプルコード :</p> <pre>try {     asReaderP3xU.setLbt(table); } catch (AsReaderP3xUException e) { }</pre>			

### 2.1.54 getRegion

関数	public String getRegion() throws AsReaderP3xUException		
引数	IN/OUT	型	説明
	OUT	String	地域
		AsReaderP3xUException	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 = asReaderP3xU.getRegion(); } catch (AsReaderP3xUException e) { }</pre>			

## 2.2 AsReaderP3xUEventListener

### 2.2.1 onStateChanged

関数	void onStateChanged(AsReaderP3xU asReaderP3xU、 AsReaderP3xUConnectionState state);		
引数	IN/OUT	型	説明
asReaderP3xU	OUT	AsReaderP3xU	AsReaderP3xU クラスのインスタンスオブジェクト。
state	OUT	AsReaderP3xUConnectionState	接続状態。 列挙型 AsReaderP3xUConnectionState ( <a href="#">2.9.5</a> 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 (<a href="#">2.1.2</a>を参照)、disconnectDevice メソッド (<a href="#">2.1.3</a>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>public void onStateChanged(AsReaderP3xU asReaderP3xU、 AsReaderP3xUConnectionState state) {     //asReaderP3xU : AsReaderP3xU クラスのインスタンスオブジェクト     //state :      AsReader の接続状態 }</pre>			

### 2.2.2 onActionChanged

関数	void onActionChanged(AsReaderP3xU asReaderP3xU、 AsReaderP3xUActionState action);		
引数	IN/OUT	型	説明
asReaderP3xU	OUT	AsReaderP3xU	AsReaderP3xU クラスのインスタンスオブジェクト。
action	OUT	AsReaderP3xUActionState	列挙型 AsReaderP3xUActionState ( <a href="#">2.9.4</a> を参照)。



■説明：

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 ([2.1.12](#)を参照)、stop メソッド ([2.1.19](#)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。

■サンプルコード：

```
public void onActionChanged(AsReaderP3xU asReaderP3xU, AsReaderP3xUActionState
action) {
    //asReaderP3xU : AsReaderP3xU オブジェクト
    //action :      AsReader 現在の操作状態
}
```

### 2.2.3 onReadTag

関数	void onReadTag(AsReaderP3xU asReaderP3xU、 AsReaderP3xUActionState action、String tag、float rssi);		
引数	IN/OUT	型	説明
asReaderP3xU	OUT	AsReaderP3xU	AsReaderP3xU クラスのインスタンスオブジェクト。
action	OUT	AsReaderP3xUActionState	列挙型 AsReaderP3xUActionState ( <a href="#">2.9.4</a> 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。

■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 ([2.1.12](#)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：

```
public void onReadTag(AsReaderP3xU asReaderP3xU, AsReaderP3xUActionState action,
String tag, float rssi) {
    //asReaderP3xU : AsReaderP3xU 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
}
```

## 2.2.4 onAccessResult

関数	void onAccessResult(AsReaderP3xU asReaderP3xU、 AsReaderP3xUResultCode code、 AsReaderP3xUActionState action、 String epc、 String data);		
引数	IN/OUT	型	説明
asReaderP3xU	OUT	AsReaderP3xU	AsReaderP3xU クラスのインスタンスオブジェクト。
code	OUT	AsReaderP3xUResultCode	列挙型 AsReaderP3xUResultCode ( <a href="#">2.9.3</a> を参照)。
action	OUT	AsReaderP3xUActionState	列挙型 AsReaderP3xUActionState ( <a href="#">2.9.4</a> 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)。
<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 (<a href="#">2.1.13</a>を参照)、writeMemory メソッド (<a href="#">2.1.14</a>を参照)、lock メソッド (<a href="#">2.1.15</a>を参照)、unlock メソッド (<a href="#">2.1.16</a>を参照)、permaLock メソッド (<a href="#">2.1.17</a>を参照)、kill メソッド (<a href="#">2.1.18</a>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>public void onAccessResult(AsReaderP3xU asReaderP3xU, AsReaderP3xUResultCode code, AsReaderP3xUActionState action, String epc, String data) {     // asReaderP3xU : AsReaderP3xU オブジェクト     // code: 実行結果     // action: AsReader 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) }</pre>			

## 2.2.5 onKeyEvent

関数	void onKeyEvent(AsReaderP3xU asReaderP3xU、 AsReaderP3xUKeyType key、 AsReaderP3xUKeyState state);		
引数	IN/OUT	型	説明
asReaderP3xU	OUT	AsReaderP3xU	AsReaderP3xU クラスのインスタンスオブジェクト。
key	OUT	AsReaderP3xUKeyType	列挙型 AsReaderP3xUKeyType ( <a href="#">2.9.6</a> を参照)。

state	OUT	AsReaderP3xUKeyState e	列挙型 AsReaderP3xUKeyState ( <a href="#">2.9.7</a> を参照)。
<p>■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。</p> <p>■サンプルコード：  <pre>public void onKeyEvent(AsReaderP3xU asReaderP3xU, AsReaderP3xUKeyType key, AsReaderP3xUKeyState state) {     //asReaderP3xU : AsReaderP3xU オブジェクト     //key : キーのタイプ     //state : キーのステータス }</pre> </p>			

## 2.3 AsReaderP3xUManager

### 2.3.1 getInstance

関数	public static AsReaderP3xU getInstance()		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xU	AsReaderP3xU オブジェクト ( <a href="#">2.1</a> を参照)。
<p>■説明： AsReaderP3xU 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：  <pre>AsReaderP3xU asReaderP3xU = AsReaderP3xUManager.getInstance();</pre> </p>			

### 2.3.2 onDestroy

関数	public static void onDestroy()		
<p>■説明： AsReaderP3xU クラスのインスタンスを破棄し、リソースをリリースします。</p> <p>■サンプルコード：  <pre>AsReaderP3xUManager.onDestroy();</pre> </p>			

### 2.3.3 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 :</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 = AsReaderP3xUManager.getVersion();</pre>			

## 2.4 AsReaderP3xUDeviceUsbCdc

### 2.4.1 AsReaderP3xUDeviceUsbCdc

関数	public AsReaderP3xUDeviceUsbCdc(Context context)		
引数	IN/OUT	型	説明
context	IN	Context	AsReaderP3xUDeviceUsbCdc オブジェクトのアクティビティを使用することができます。
<p>■ 説明 :</p> <p>AsReaderP3xUDeviceUsbCdc 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>AsReaderP3xUDeviceUsbCdc usbCdc = new AsReaderP3xUDeviceUsbCdc(this);</pre>			

## 2.5 AsReaderP3xULbtItem

### 2.5.1 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>			

## 2.5.2 isUsed

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

## 2.5.3 setUsed

関数	public void setUsed(boolean used)		
引数	IN/OUT	型	説明
	IN	boolean	true: 有効 false: 無効。
<p>■説明： 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： item.setUsed(true);</p>			

## 2.5.4 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： LBT 有効時の周波数値を取得します。</p> <p>■サンプルコード： String mFrequency= item.getFrequency();</p>			

## 2.6 AsReaderP3xULockParam

### 2.6.1 AsReaderP3xULockParam

関数	public AsReaderP3xULockParam(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	AsReaderP3xULockParam	AsReaderP3xULockParam オブジェクト。
<p>■ 説明 :</p> <p>AsReaderP3xULockParam 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>AsReaderP3xULockParam param = new AsReaderP3xULockParam(false, false, true, false, false);</pre>			

## 2.7 AsReaderP3xUPowerRange

### 2.7.1 AsReaderP3xUPowerRange

関数	public AsReaderP3xUPowerRange(int min、 int max)		
引数	IN/OUT	型	説明
min	IN	int	電波出力の最小値を設定します。
max	IN	int	電波出力の最大値を設定します。
	OUT	AsReaderP3xUPowerRange	AsReaderP3xUPowerRange オブジェクト。

## ■ 説明 :

AsReaderP3xUPowerRange クラスのインスタンスを作成します。

## ■ サンプルコード :

```
AsReaderP3xUPowerRange mPowerRange = new AsReaderP3xUPowerRange(150, 260);
```

## 2.7.2 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>			

## 2.7.3 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>			

## 2.8 AsReaderP3xUSelectMaskEpcParam

### 2.8.1 AsReaderP3xUSelectMaskEpcParam

関数	public AsReaderP3xUSelectMaskEpcParam()		
引数	IN/OUT	型	説明
	OUT	AsReaderP3xUSelectMaskEpcParam	AsReaderP3xUSelectMaskEpcParam オブジェクト。
■ 説明 : AsReaderP3xUSelectMaskEpcParam クラスのインスタンスを作成します。			
■ サンプルコード :			

```
AsReaderP3xUSelectMaskEpcParam selectMaskEpcParam = new
AsReaderP3xUSelectMaskEpcParam();
```

## 2.8.2 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>			

## 2.8.3 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>			

## 2.8.4 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>			

## 2.8.5 setLength

関数	public void setLength(int length)		
引数	IN/OUT	型	説明
length	IN	int	マスクの長さ。



■説明：  
現在のマスクの長さを設定します。

■サンプルコード：  
`selectMaskEpcParam.setLength(1);`

## 2.8.6 getMask

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

## 2.8.7 setMask

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

## 2.9 Enum

### 2.9.1 AsReaderP3xUMaskTargetType

定義	説明
S0	Inventoried S0
S1	inventoried S1
S2	inventoried S2
S3	inventoried S3
SL	Selection Flags

## 2.9.2 AsReaderP3xUSessionFlag

定義	説明
A	A only
B	B only
AB	A or B

## 2.9.3 AsReaderP3xUResultCode

定義	説明
NoError = 0x0000	成功
OtherError = 0x0001	その他のエラー
MemoryOverrun = 0x0003	アクセスメモリーが範囲外
MemoryLocked = 0x0004	メモリーロック
NonSpecificError = 0x000F	未知なエラー
InOperation = 0xE000	実行中
OutOfRange = 0xE001	範囲外
NotConnected = 0xE100	接続されない
InvalidParameter = 0xE200	渡したパラメータが無効
InvalidResponse = 0xE300	返したパラメータが無効
NotSupportFirmware = 0xEE00	ファームウェアサポートされない
Timeout = 0xEFFF	タイムアウト
OutOfRetries = 0xF009	リトライ回数オーバー
OperationFailed = 0xFFFF	操作失敗

## 2.9.4 AsReaderP3xUActionState

定義	説明
Inventory = 0x66	インベントリー中
ReadMemory = 0x72	メモリー読取中
WriteMemory = 0x77	メモリー書き込み中
Kill = 0x6B	タグキル中
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
Stop = 0x73	操作停止
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

## 2.9.5 AsReaderP3xUConnectionState

定義	説明
Disconnected = 0	接続切断
Listen = 1	リスナー接続
Connecting = 2	接続中
Connected = 3	接続済み
Cancelling = 4	接続取消中

## 2.9.6 AsReaderP3xUKeyType

定義	説明
Trigger = 0	トリガー

## 2.9.7 AsReaderP3xUKeyState

定義	説明
KeyUp= 0	押下解除
KeyDown= 1	押下

## 2.9.8 AsReaderP3xUMemoryBank

定義	説明
Reserved	Reserved エリア
EPC	EPC エリア
TID	TID エリア
User	User エリア

## 2.9.9 AsReaderP3xUBuzzerState

定義	説明
Off	ブザー音がない
Low	ブザー音が低い
High	ブザー音が高い

## 2.9.10 AsReaderP3xUQuerySession

定義	説明
S0	Inventoried S0
S1	Inventoried S1
S2	Inventoried S2
S3	Inventoried S3