

Star mPOP Android SDK ユーザーズマニュアル

2015 年 08 月 24 日

要約

このマニュアルは Star mPOP Android SDK に関する情報を提供するものです。
ユーザー向けに mPOP アプリケーションの構築に関するガイドラインを示します。

ドキュメントの履歴

日付	変更内容
2015 年 08 月 24 日	第 1 版発行

本書に関して

本マニュアルは StarIO ライブラリ / StarIO_Extension ライブラリと mPOP が通信を行う Android アプリケーションの作成方法を解説しています。

StarIO ライブラリの詳細を知りたい場合、StarIO Android SDK をダウンロードし、ドキュメントを参照してください。

またこのマニュアルはアプリケーション開発者を対象に作成しており、利用者は Java 言語の基礎を理解していることを前提としています。

[スター精密グローバルサポートサイト](#)の Developers セクションには、その他のオペレーティングシステムとプログラミング言語に利用可能な SDK が用意されています。最新の SDK、テクニカルドキュメント、FAQ 及びその他の追加情報については Developers セクションをご確認ください。

注意事項

- 本マニュアルの内容は予告無く変更する場合があります。
- スター精密株式会社は正確な情報を提供するためにあらゆる措置を取っていますが、誤りや不作為について責任を負うものではありません。
- スター精密株式会社はこのマニュアルに記載されている情報の使用に起因するいかなる損害に対しても責任を負うものではありません。
- 本マニュアルの一部あるいは全部を無断で複写・複製・転載することは固くお断りします。

目次

1	はじめに	6
2	mPOP アプリケーション構成方法	9
2.1	StarIO_Extension ライブラリを追加する	9
2.2	AndroidManifest.xml へ項目を設定する	11
2.3	device_filter.xml へ項目を設定する	13
3	StarIoExtManager	14
3.1	StarIoExtManager コンストラクタ	15
3.2	connect メソッド	16
3.3	disconnect メソッド	17
3.4	getPort メソッド	17
3.5	getPrinterOnlineStatus メソッド	18
3.6	getPrinterPaperReadyStatus メソッド	18
3.7	getPrinterCoverOpenStatus メソッド	19
3.8	getCashDrawerOpenStatus メソッド	19
3.9	getBarcodeReaderConnectStatus メソッド	20
3.10	getCashDrawerOpenActiveHigh メソッド	20
3.11	setCashDrawerOpenActiveHigh メソッド	21
3.12	setListener メソッド	22
3.13	Type コンスタント	22
3.14	Status コンスタント	23
4	StarIoExtManagerListener	24
4.1	didPrinterImpossible メソッド	24
4.2	didPrinterOnline メソッド	25
4.3	didPrinterOffline メソッド	25
4.4	didPrinterPaperReady メソッド	26
4.5	didPrinterPaperNearEmpty メソッド	26
4.6	didPrinterPaperEmpty メソッド	27
4.7	didPrinterCoverOpen メソッド	27
4.8	didPrinterCoverClose メソッド	28
4.9	didCashDrawerOpen メソッド	28
4.10	didCashDrawerClose メソッド	29
4.11	didBarcodeReaderImpossible メソッド	29
4.12	didBarcodeReaderConnect メソッド	30
4.13	didBarcodeReaderDisconnect メソッド	30
4.14	didBarcodeDataReceive メソッド	31
4.15	didAccessoryConnectSuccess メソッド	31
4.16	didAccessoryConnectFailure メソッド	32
4.17	didAccessoryDisconnect メソッド	32
5	SCBFactory	33
5.1	createBuilder メソッド	33
5.2	Emulation コンスタント	34
6	ISCBBuilder	35
6.1	appendBitmap メソッド	35
6.2	appendAbsolutePositionBitmap メソッド	37
6.3	appendAlignmentBitmap メソッド	39
6.4	getBuffer メソッド	40
7	SCBAignment	41

7.1	Position コンスタント	41
8	SCBBitmapConverter	41
8.1	Rotation コンスタント	41
9	Star mPOP Android SDK 機能	42
9.1	PrinterFragment.java / PrintExtFragment.java	42
9.2	CashDrawerFragment.java / CashDrawerExtFragment.java	45
9.3	BarcodeExtFragment.java	48
9.4	CombinationFragment.java / CombinationExtFragment.java	51
9.5	Communication	54

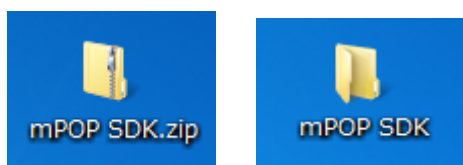
1 はじめに

Android のプロジェクトをビルドするには Android Studio が必要です。これらのツールは [Android Studio – Official Site](#) から入手可能です。

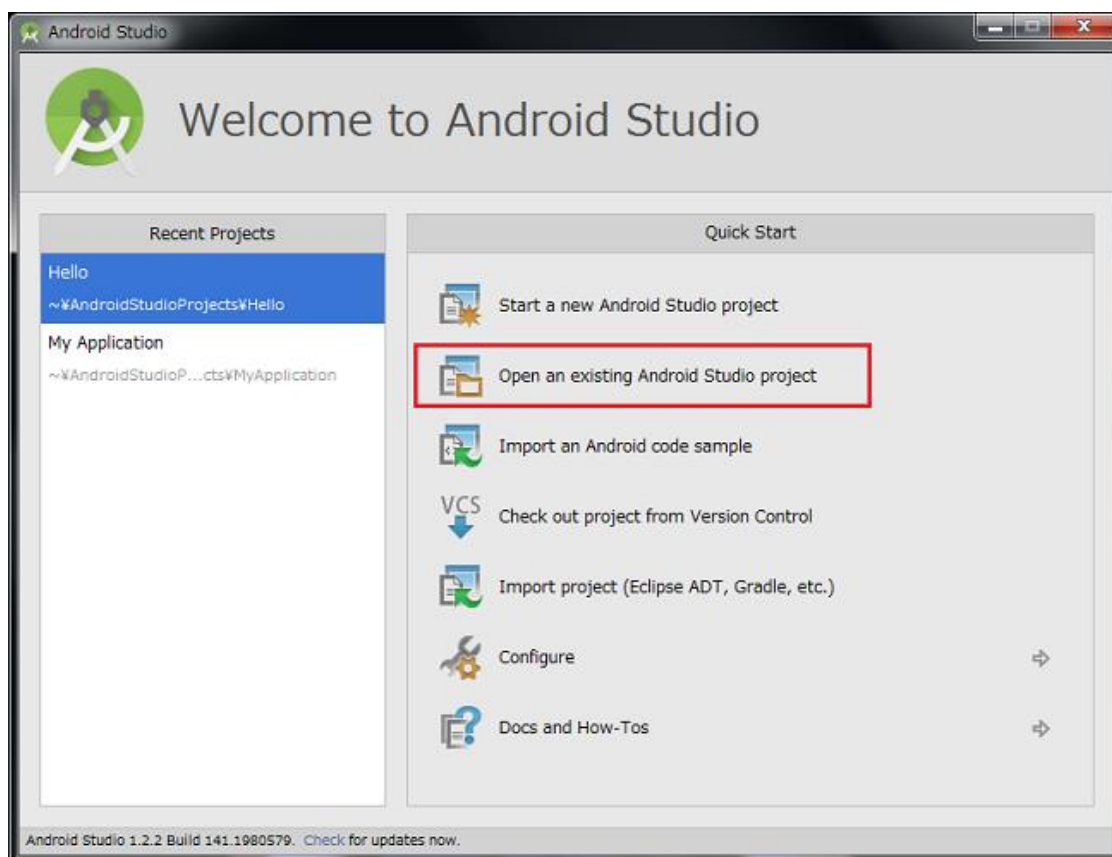
事前に、開発を行う PC に Android Studio のインストールを行ってください。サポートまたは追加情報が必要な場合は [Android Studio – Official Site](#) を参照してください。

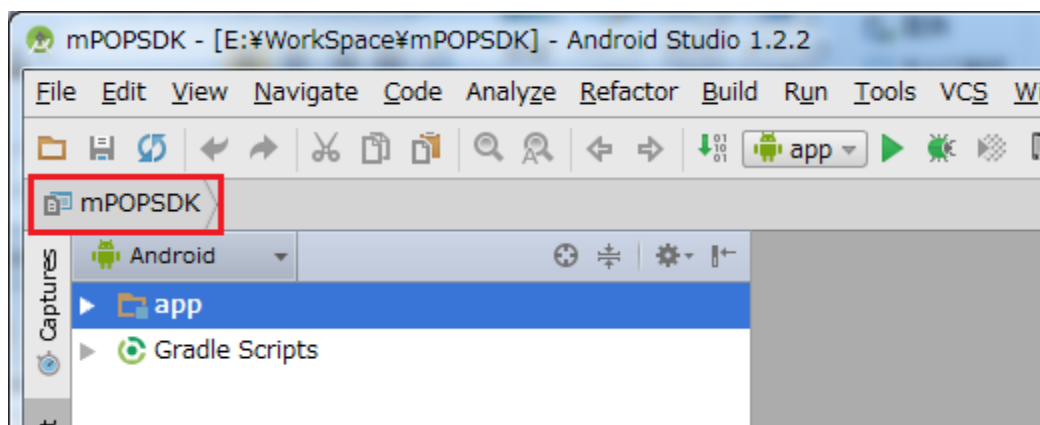
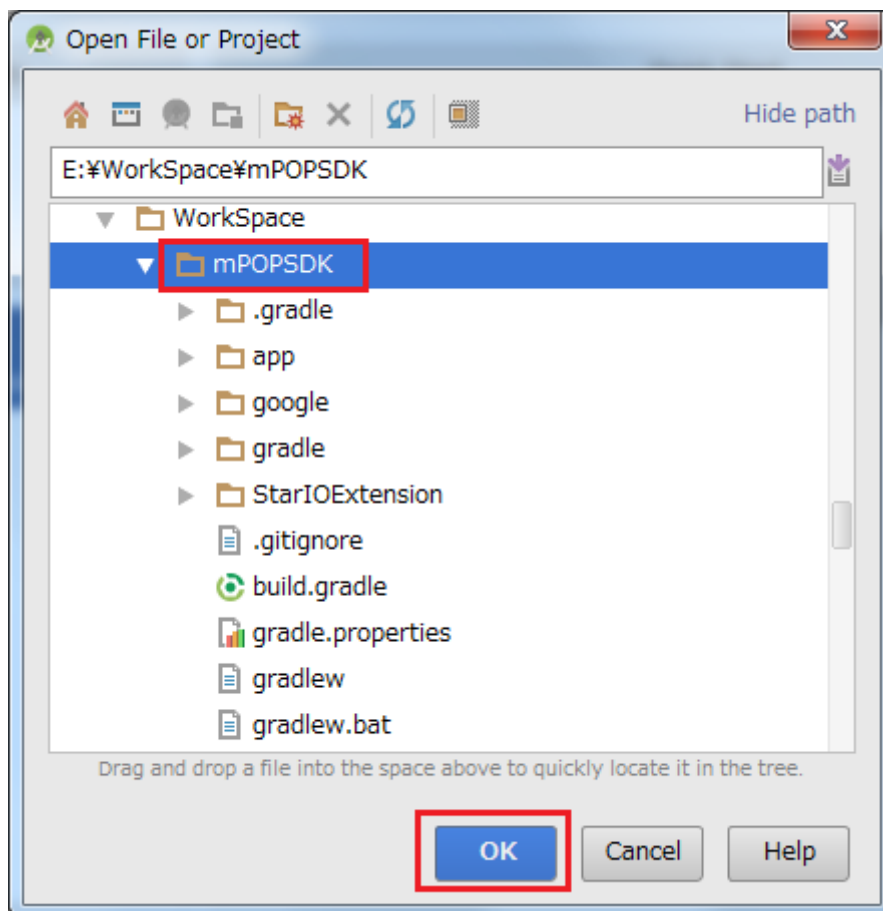
Android Studio で Star mPOP Android SDK プロジェクトを開く方法

1.Star mPOP Android SDK フォルダを解凍します。



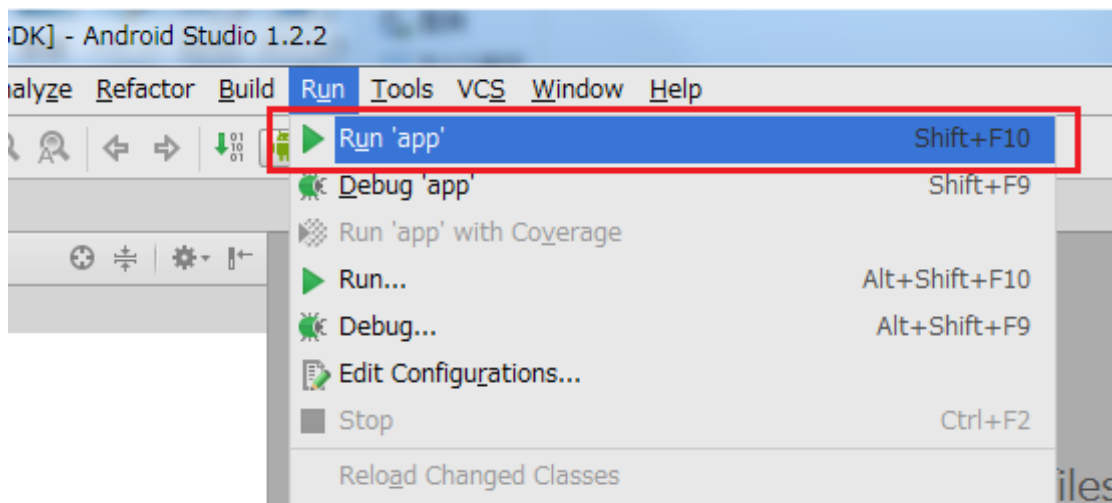
2.Android Studio を実行し、mPOP SDK を開きます。



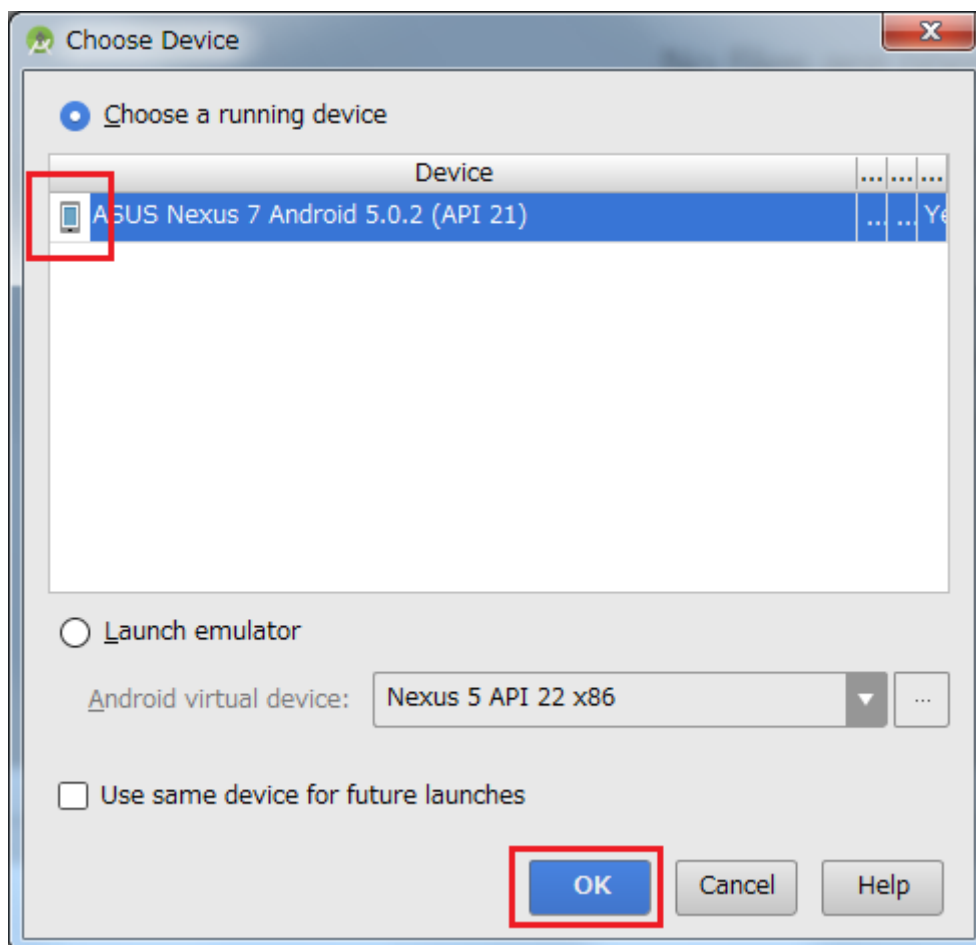


プロジェクトを実行する

1. 上部メニューバーの"Run"をクリックして"Run 'app'"を選択します。



2. デバイスを選択します。



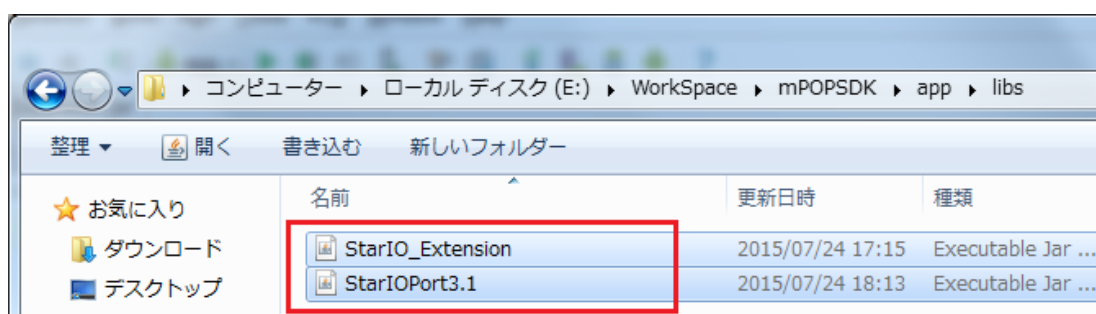
2 mPOP アプリケーション構成方法

Star mPOP Android SDK プロジェクトには、既に StarIO ライブラリ / StarIO_Extension ライブラリは含まれています。(この SDK をテストする場合、そのまま使用できます)

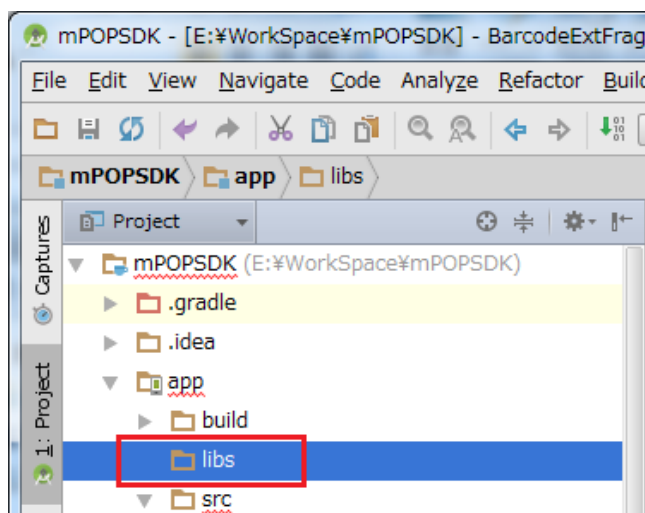
ただし、新規のアプリケーションを作成する場合、プロジェクトに必要なライブラリを追加する必要があります。

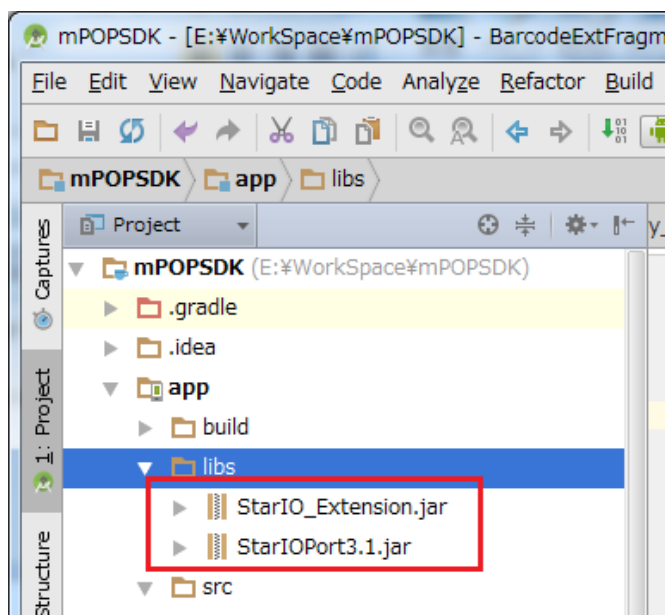
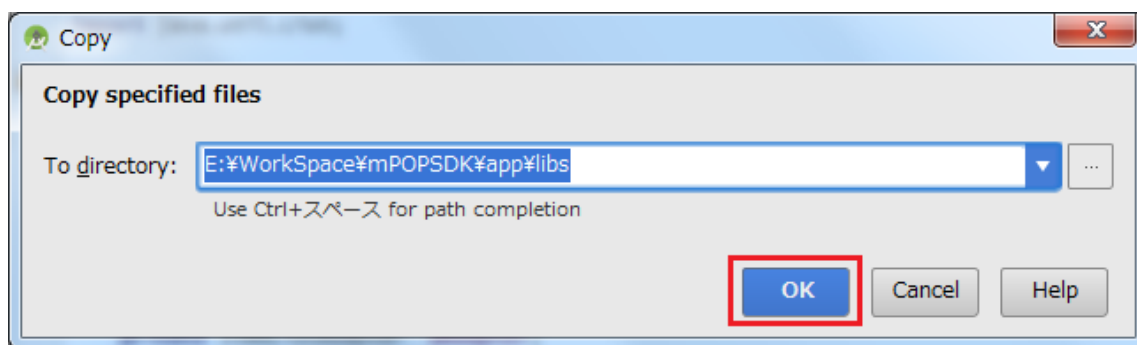
2.1 StarIO_Extension ライブラリを追加する

1. mPOP SDK の libs フォルダにあるライブラリを選択します。



2. プロジェクトの libs フォルダにライブラリをコピーします。

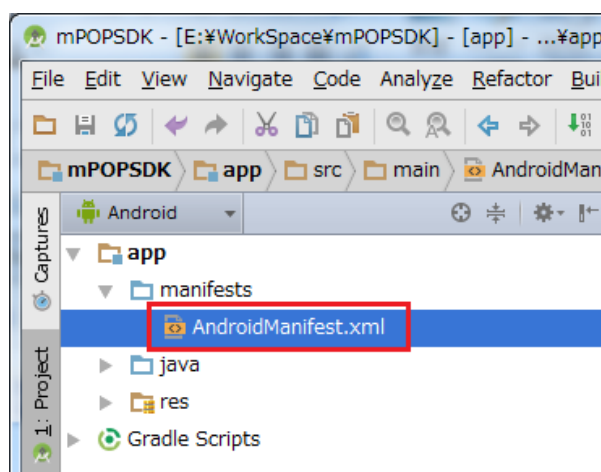




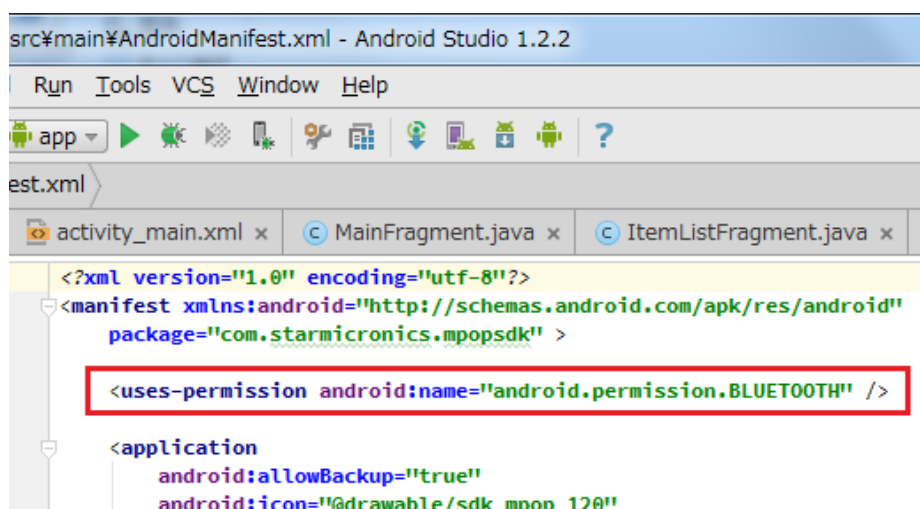
3. ライブラリはプロジェクトに追加されます。

2.2 AndroidManifest.xml へ項目を設定する

1. AndroidManifest.xml ファイルをダブルクリックします。

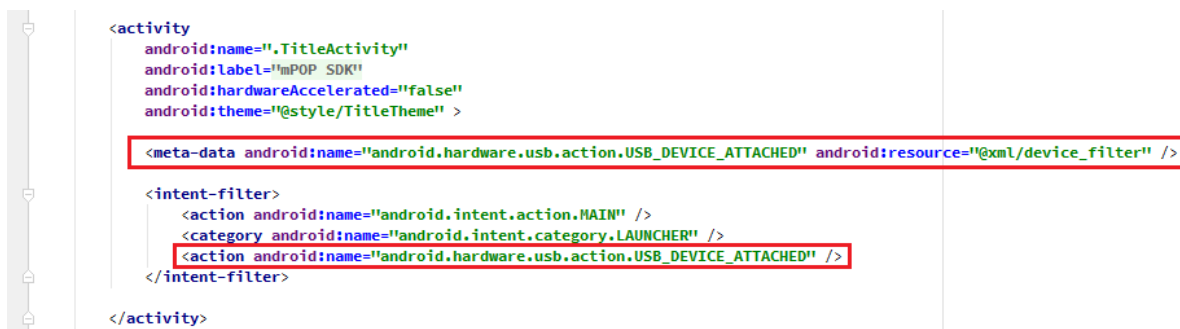


2. Bluetooth 接続の場合、Bluetooth 接続の利用に関する追記を行います。



```
<uses-permission android:name="android.permission.BLUETOOTH" />
```

3. USB 接続の場合、USB 接続の利用に関する追記を行います。



```

<activity
    android:name=".TitleActivity"
    android:label="@string/app_name"
    android:hardwareAccelerated="false"
    android:theme="@style/TitleTheme" >

    <meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" android:resource="@xml/device_filter" />

    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
    </intent-filter>
</activity>

```

```

<meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
android:resource="@xml/device_filter" />

```

```

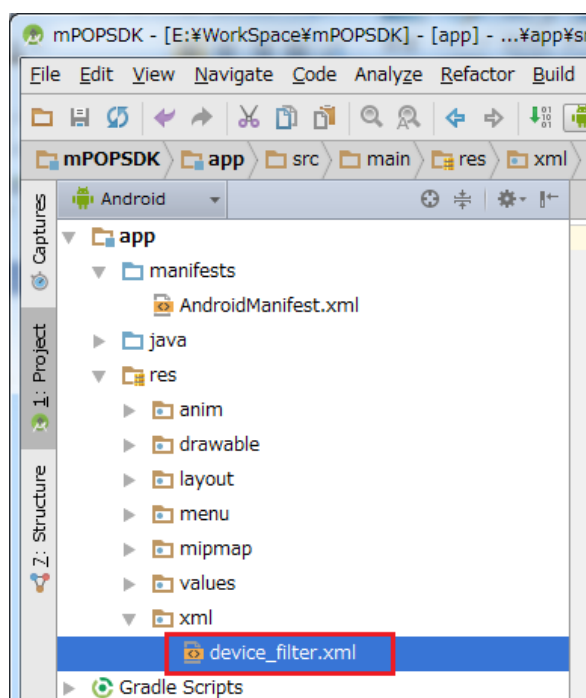
<action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />

```

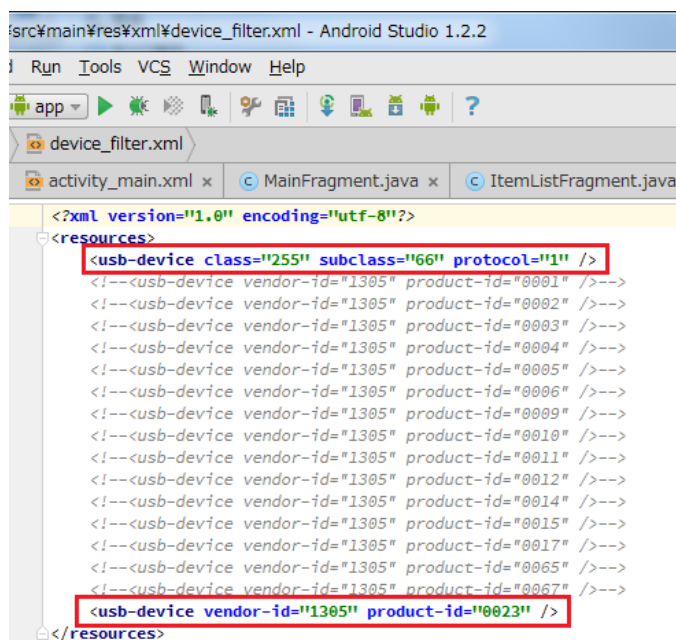
4. AndroidManifest.xml ファイルの設定は完了です。

2.3 device_filter.xml へ項目を設定する

1. device_filter.xml ファイルをダブルクリックします。



2. USB 接続の場合、USB 接続の利用に関する追記を行います。



```
<usb-device class="255" subclass="66" protocol="1" />
<usb-device vendor-id="1305" product-id="0023" />
```

3. device_filter.xml ファイルの設定は完了です。

3 StarloExtManager

コンストラクタ

名称	説明
StarloExtManager	StarloExtManager を初期化します。

メソッド

名称	説明
connect	マネージメントを開始します。
disconnect	マネージメントを停止します。
getPort	StarloPort オブジェクトを取得します。
getPrinterOnlineStatus	プリンタオンラインステータスを取得します。
getPrinterPaperReadyStatus	プリンタ用紙レディステータスを取得します。
getPrinterCoverOpenStatus	プリンタカバーオープンステータスを取得します。
getCashDrawerOpenStatus	キャッシュドロアオープンステータスを取得します。
getBarcodeReaderConnectStatus	バーコードリーダー接続ステータスを取得します。
getCashDrawerOpenActiveHigh	キャッシュドロア開閉センサのモードを取得します。
setCashDrawerOpenActiveHigh	キャッシュドロア開閉センサのモードを設定します。
setListener	StarloExtManager のリスナーオブジェクトを設定します。

コンスタント

名称	説明
Type	マネージャタイプ定数。
Status	ステータス定数。

3.1 StarIoExtManager コンストラクタ

StarIoExtManager を初期化します。

宣言

```
StarIoExtManager(Type type, String portName, String portSetting, int ioTimeoutMillis, Context context);
```

引数

名称	説明	型
type	マネージャタイプ <ul style="list-style-type: none"> Standard ... プリンタ、キャッシュドロフのマネージメント WithBarcodeReader ... プリンタ、キャッシュドロフ、バーコードリーダーのマネージメント OnlyBarcodeReader ... バーコードリーダーのマネージメント 	Type
portName	プリンタポート名	String
portSettings	ポートセッティング	String
ioTimeoutMillis	内部制御および API のタイムアウト値	int

戻り値

説明	型
StarIoExtManager オブジェクト	StarIoExtManager

Example

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    PrinterSetting setting = new PrinterSetting(getActivity());

    mStarIoExtManager = new StarIoExtManager(StarIoExtManager.Type.Standard, setting.getPortName(),
    setting.getPrinterType(), 10000, getActivity()); // 10000mS!!!

    mStarIoExtManager.setListener(mStarIoExtManagerListener);
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

◆portName、portSettings、ioTimeoutMills 詳細を知りたい場合、StarIO Android SDK をダウンロードし、ドキュメントを参照してください。

3.2 connect メソッド

マネージメントを開始します。

宣言

```
boolean connect();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
リザルト <ul style="list-style-type: none"> true ... 成功 false ... 失敗 	boolean

Example

```
public void onResume() {
    super.onResume();

    mStarloExtManagerListener.didAccessoryDisconnect();

    AsyncTask<Void, Void, Boolean> asyncTask = new AsyncTask<Void, Void, Boolean>() {
        @Override
        protected void onPreExecute() {

            ...

        }

        @Override
        protected Boolean doInBackground(Void... params) {
            return mStarloExtManager.connect();
        }

        @Override
        protected void onPostExecute(Boolean result) {

            ...

            if (! result) {
                AlertDialog.Builder dialogBuilder = new AlertDialog.Builder(getActivity());

                dialogBuilder.setTitle("Communication Result");
                dialogBuilder.setMessage("Fail to Open Port.");

                dialogBuilder.setPositiveButton("OK", null);

                dialogBuilder.show();
            }
        }
    };

    asyncTask.execute();
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

3.3 disconnect メソッド

マネージメントを停止します。

宣言

```
boolean disconnect();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
リザルト <ul style="list-style-type: none"> true ... 成功 false ... 失敗 	boolean

Example

```
public void onPause() {
    super.onPause();

    mStarIoExtManager.disconnect();
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

3.4 getPort メソッド

StarIoPort オブジェクトを取得します。

宣言

```
StarIoPort getPort();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
StarIoPort オブジェクト	StarIoPort

◆StarIo オブジェクトの詳細を知りたい場合、StarIo Android SDK をダウンロードし、ドキュメントを参照してください。

3.5 getPrinterOnlineStatus メソッド

プリンタオンラインステータスを取得します。

宣言

```
Status getPrinterOnlineStatus();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
プリンタオンラインステータス <ul style="list-style-type: none"> Invalid ... 不定 Impossible ... 機器使用不能 PrinterOnline ... プリンタオンライン検出 PrinterOffline ... プリンタオフライン検出 	Status

3.6 getPrinterPaperReadyStatus メソッド

プリンタ用紙レディステータスを取得します。

宣言

```
Status getPrinterPaperReadyStatus();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
プリンタ用紙レディステータス <ul style="list-style-type: none"> Invalid ... 不定 Impossible ... 機器使用不能 PaperReady ... プリンタ用紙レディ検出 PrinterPaperNearEmpty ... プリンタ用紙ニアエンド検出 PrinterPaperEmpty ... プリンタ用紙エンプティ検出 	Status

3.7 getPrinterCoverOpenStatus メソッド

プリンタカバーオープンステータスを取得します。

宣言

```
Status getPrinterCoverOpenStatus();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
プリンタカバーオープンステータス <ul style="list-style-type: none"> Invalid ... 不定 Impossible ... 機器使用不能 PrinterCoverOpen ... プリンタカバーオープン検出 PrinterCoverClose ... プリンタカバークローズ検出 	Status

3.8 getCashDrawerOpenStatus メソッド

キャッシュドロアーオープンステータスを取得します。

宣言

```
Status getCashDrawerOpenStatus();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
キャッシュドロアーオープンステータス <ul style="list-style-type: none"> Invalid ... 不定 Impossible ... 機器使用不能 CashDrawerOpen ... キャッシュドロアーオープン検出 CashDrawerClose ... キャッシュドロアークローズ検出 	Status

3.9 getBarcodeReaderConnectStatus メソッド

バーコードリーダー接続ステータスを取得します。

宣言

```
Status getBarcodeReaderConnectStatus();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
バーコードリーダー接続ステータス <ul style="list-style-type: none"> Invalid ... 不定 Impossible ... 機器使用不能 BarcodeReaderConnect ... バーコードリーダー接続検出 BarcodeReaderDisconnect ... バーコードリーダー切断検出 	Status

3.10 getCashDrawerOpenActiveHigh メソッド

キャッシュドロアー開閉センサのモードを取得します。

宣言

```
boolean getCashDrawerOpenActiveHigh();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
キャッシュドロアー開閉センサモード <ul style="list-style-type: none"> YES ... High アクティブ NO ... Low アクティブ 	BOOL
※ mPOP メカニズム構造においてはデフォルト値 (YES) のこと。	

3.11 setCashDrawerOpenActiveHigh メソッド

キャッシュドロアー開閉センサのモードを設定します。

宣言

```
void setCashDrawerOpenActiveHigh(boolean isActiveHigh);
```

引数

名称	説明	型
isActiveHigh	キャッシュドロアー開閉センサモード <ul style="list-style-type: none"> • YES ... High アクティブ • NO ... Low アクティブ ※ mPOP メカニズム構造においてはデフォルト値 (YES) のこと。	boolean

戻り値

説明	型
-	-

3.12 setListener メソッド

StarloExtManager のリスナーオブジェクトを設定します。

宣言

```
void setListener(StarloExtManagerListener listener);
```

引数

名称	説明	型
listener	StarloExtManager のリスナーオブジェクト	StarloExtManagerListener

戻り値

説明	型
-	-

Example

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    PrinterSetting setting = new PrinterSetting(getActivity());

    mStarloExtManager = new StarloExtManager(StarloExtManager.Type.Standard, setting.getPortName(), "",
    setting.getTimeout(), getActivity());

    mStarloExtManager.setListener(mStarloExtManagerListener);
}

StarloExtManagerListener mStarloExtManagerListener = new StarloExtManagerListener() {

    ...

};
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java /
CombinationExtFragment.java を参照ください。

3.13 Type コンスタント

マネージャタイプ定数。

宣言

```
public enum Type {
    Standard,
    WithBarcodeReader,
    OnlyBarcodeReader
}
```

定数

名称	説明
Standard	プリンタ、キャッシュドロワのマネージメント
WithBarcodeReader	プリンタ、キャッシュドロワ、バーコードリーダーのマネージメント
OnlyBarcodeReader	バーコードリーダーのマネージメント

3.14 Status コンスタント

ステータス定数。

宣言

```
public enum Status {
    Invalid,
    Impossible,
    PrinterOnline,
    PrinterOffline,
    PrinterPaperReady,
    PrinterPaperNearEmpty,
    PrinterPaperEmpty,
    PrinterCoverOpen,
    PrinterCoverClose,
    CashDrawerOpen,
    CashDrawerClose,
    BarcodeReaderConnect,
    BarcodeReaderDisconnect
}
```

定数

名称	説明
Invalid	不定
Impossible	機器使用不能
PrinterOnline	プリンタオンライン検出
PrinterOffline	プリンタオフライン検出
PrinterPaperReady	プリンタ用紙レディ検出
PrinterPaperNearEmpty	プリンタ用紙ニアエンド検出
PrinterPaperEmpty	プリンタ用紙エンpty検出
PrinterCoverOpen	プリンタカバーオープン検出
PrinterCoverClose	プリンタカバークローズ検出
CashDrawerOpen	キャッシュドロアーオープン検出
CashDrawerClose	キャッシュドロークローズ検出
BarcodeReaderConnect	バーコードリーダー接続検出
BarcodeReaderDisconnect	バーコードリーダー切断検出

4 StarloExtManagerListener

メソッド

名称	説明
didPrinterImpossible	プリンタ使用不能を通知。 * Override
didPrinterOnline	プリンタオンラインを通知。 * Override
didPrinterOffline	プリンタオフラインを通知。 * Override
didPrinterPaperReady	プリンタ用紙レディを通知。 * Override
didPrinterPaperNearEmpty	プリンタ用紙ニアエンプティを通知。 * Override
didPrinterPaperEmpty	プリンタ用紙エンプティを通知。 * Override
didPrinterCoverOpen	プリンタカバーオープンを通知。 * Override
didPrinterCoverClose	プリンタカバークローズを通知。 * Override
didCashDrawerOpen	キャッシュドロアオープンを通知。 * Override
didCashDrawerClose	キャッシュドロアクローズを通知。 * Override
didBarcodeReaderImpossible	バーコードリーダー使用不能を通知。 * Override
didBarcodeReaderConnect	バーコードリーダー接続を通知。 * Override
didBarcodeReaderDisconnect	バーコードリーダー切断を通知。 * Override
didBarcodeDataReceive	バーコードデータ受信を通知。 * Override
didAccessoryConnectSuccess	アクセサリ（Bluetooth / USB）接続成功を通知。 * Override
didAccessoryConnectFailure	アクセサリ（Bluetooth / USB）接続失敗を通知。 * Override
didAccessoryDisconnect	アクセサリ（Bluetooth / USB）切断を通知。 * Override

4.1 didPrinterImpossible メソッド

プリンタ使用不能を通知。 * Override

宣言

```
void didPrinterImpossible();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
public void didPrinterImpossible() {
    mComment.setText("Printer Impossible.");
    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / CombinationExtFragment.java を参照ください。

4.2 didPrinterOnline メソッド

プリンタオンラインを通知。* Override

宣言

```
void didPrinterOnline();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didPrinterOnline() {
    mComment.setText("Printer Online.");
    mComment.setTextColor(Color.BLUE);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

4.3 didPrinterOffline メソッド

プリンタオフラインを通知。* Override

宣言

```
void didPrinterOffline();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didPrinterOffline() {
    mComment.setText("Printer Offline.");
    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

4.4 didPrinterPaperReady メソッド

プリンタ用紙レディを通知。* Override

宣言

```
void didPrinterPaperReady();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didPrinterPaperReady() {
    mComment.setText("Printer Paper Ready.");

    mComment.setTextColor(Color.BLUE);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

4.5 didPrinterPaperNearEmpty メソッド

プリンタ用紙ニアエンptyを通知。* Override

※ mPOP においてはニアエンptyセンサが搭載されていないため通知されません。

宣言

```
void didPrinterPaperNearEmpty();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didPrinterPaperNearEmpty() {
    mComment.setText("Printer Paper Near Empty.");

    mComment.setTextColor(0xffffa500); // Orange
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

4.6 didPrinterPaperEmpty メソッド

プリンタ用紙エンプティを通知。* Override

宣言

```
void didPrinterPaperEmpty();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didPrinterPaperEmpty() {
    mComment.setText("Printer Paper Empty.");
    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

4.7 didPrinterCoverOpen メソッド

プリンタカバーオープンを通知。* Override

宣言

```
void didPrinterCoverOpen();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didPrinterCoverOpen() {
    mComment.setText("Printer Cover Open.");
    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

4.8 didPrinterCoverClose メソッド

プリンタカバークローズを通知。* Override

宣言

```
void didPrinterCoverClose();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didPrinterCoverClose() {
    mComment.setText("Printer Cover Close.");

    mComment.setTextColor(Color.BLUE);
}
```

PrinterExtFragment.java / CombinationExtFragment.java を参照ください。

4.9 didCashDrawerOpen メソッド

キャッシュドロアーオープンを通知。* Override

宣言

```
void didCashDrawerOpen();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didCashDrawerOpen() {
    mComment.setText("Cash Drawer Open.");

    // mComment.setTextColor(Color.RED);
    mComment.setTextColor(Color.MAGENTA);
}
```

CashDrawerExtFragment.java / CombinationExtFragment.java を参照ください。

4.10 didCashDrawerClose メソッド

キャッシュドロアークローズを通知。* *Override*

宣言

```
void didCashDrawerClose();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didCashDrawerClose() {
    mComment.setText("Cash Drawer Close.");

    mComment.setTextColor(Color.BLUE);
}
```

CashDrawerExtFragment.java / *CombinationExtFragment.java* を参照ください。

4.11 didBarcodeReaderImpossible メソッド

バーコードリーダー使用不能を通知。* *Override*

宣言

```
void didBarcodeReaderImpossible();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didBarcodeReaderImpossible() {
    mComment.setText("Barcode Reader Impossible.");

    mComment.setTextColor(Color.RED);
}
```

BarcodeReaderExtFragment.java / *CombinationExtFragment.java* を参照ください。

4.12 didBarcodeReaderConnect メソッド

バーコードリーダー接続を通知。* *Override*

宣言

```
void didBarcodeReaderConnect();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didBarcodeReaderConnect() {
    mComment.setText("Barcode Reader Connect.");

    mComment.setTextColor(Color.BLUE);
}
```

BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

4.13 didBarcodeReaderDisconnect メソッド

バーコードリーダー切断を通知。* *Override*

宣言

```
void didBarcodeReaderDisconnect();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didBarcodeReaderDisconnect() {
    mComment.setText("Barcode Reader Disconnect.");

    mComment.setTextColor(Color.RED);
}
```

BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

4.14 didBarcodeDataReceive メソッド

バーコードデータ受信を通知。* Override

宣言

```
void didBarcodeDataReceive(byte[] data);
```

引数

名称	説明	型
data	受信したバーコードデータ	byte[]

戻り値

説明	型
-	-

Example

```
@Override
public void onBarcodeDataReceive(byte[] data) {
    String[] barcodeDataArray = new String(data).split("\n");

    for(String barcodeData:barcodeDataArray) {
        ...
    }
}
```

BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

4.15 didAccessoryConnectSuccess メソッド

アクセサリ（Bluetooth / USB）接続成功を通知。* Override

宣言

```
void didAccessoryConnectSuccess();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didAccessoryConnectSuccess() {
    mComment.setText("Accessory Connect Success.");

    mComment.setTextColor(Color.BLUE);
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

4.16 didAccessoryConnectFailure メソッド

アクセサリー（Bluetooth / USB）接続失敗を通知。* *Override*

宣言

```
void didAccessoryConnectFailure();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didAccessoryConnectFailure() {
    mComment.setText("Accessory Connect Failure.");

    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

4.17 didAccessoryDisconnect メソッド

アクセサリー（Bluetooth / USB）切断を通知。* *Override*

宣言

```
void didAccessoryDisconnect();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
@Override
void didAccessoryDisconnect() {
    mComment.setText("Accessory Disconnect.");

    mComment.setTextColor(Color.RED);
}
```

PrinterExtFragment.java / CashDrawerExtFragment.java / BarcodeReaderExtFragment.java / CombinationExtFragment.java を参照ください。

5 SCBFactory

メソッド

名称	説明
createBuilder	コマンド生成オブジェクトを生成します。

コンスタント

名称	説明
Emulation	エミュレーション指定定数。

5.1 createBuilder メソッド

コマンド生成オブジェクトを生成します。

宣言

```
public static ISCBBuilder createBuilder(Emulation emulation);
```

引数

名称	説明	型
emulation	エミュレーション指定 <ul style="list-style-type: none"> Star ... Star エミュレーション EscPos ... ESC/POS エミュレーション 	Emulation

戻り値

説明	型
ISCBBuilder オブジェクト	ISCBBuilder

Example

```
static byte[] createCommandsRasterModeCoupon(Resources resource ) {
    CommandDataList commands = new CommandDataList();

    Bitmap bitmap = BitmapFactory.decodeResource( resource, R.drawable.coupon_image);

    ISCBBuilder builder = SCBFactory.createBuilder(SCBFactory.Emulation.Star);

    builder.appendBitmap(bitmap, false, 384, SCBBitmapConverter.Rotation.Right90);

    List<byte[]> listBuf = builder.getBuffer();

    for(byte[] buf:listBuf) {
        commands.add(buf);
    }

    return commands.getByteArray();
}
```

PrinterFunctions.java を参照ください。

5.2 Emulation コンスタント

エミュレーション指定定数。

宣言

```
public enum Emulation {
    Star,
    EscPos
}
```

定数

名称	説明
Star	Star エミュレーション
EscPos	ESC/POS エミュレーション

6 ISCBBuilder

メソッド

名称	説明
appendBitmap	ビットマップの印刷コマンドを生成し、commands プロパティに追加します。
appendAbsolutePositionBitmap	絶対位置指定のビットマップ印刷コマンドを生成し、commands プロパティに追加します。
appendAlignmentBitmap	位置揃えビットマップの印刷コマンドを生成し、commands プロパティに追加します。
getBuffer	commands プロパティを取得します。

6.1 appendBitmap メソッド

ビットマップの印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
void appendBitmap(Bitmap bitmap, boolean diffusion, int width, boolean bothScale, SCBBitmapConverter.Rotation rotation);
```

```
void appendBitmap(Bitmap bitmap, boolean diffusion, int width, SCBBitmapConverter.Rotation rotation);
```

```
void appendBitmap(Bitmap bitmap, boolean diffusion, SCBBitmapConverter.Rotation rotation);
```

```
void appendBitmap(Bitmap bitmap, boolean diffusion, int width, boolean bothScale);
```

```
void appendBitmap(Bitmap bitmap, boolean diffusion, int width);
```

```
void appendBitmap(Bitmap bitmap, boolean diffusion);
```

引数

名称	説明	型
bitmap	元となるビットマップオブジェクト	Bitmap
diffusion	誤差拡散 <ul style="list-style-type: none"> true ... 誤差拡散を行う false ... 誤差拡散を行わない 	boolean
width	変換後のビットマップ幅（ドット数単位）	int
bothScale	width プロパティによる変換率に合わせた高さ変換 <ul style="list-style-type: none"> true ... 高さ変換を行う false ... 高さ変換を行わない 	boolean
rotation	変換後の回転 <ul style="list-style-type: none"> Normal ... 回転なし Right90 ... 右 90 度回転 Left90 ... 左 90 度回転 Rotate180 ... 180 度回転 	SCBBitmapConverter.Rotation

戻り値

説明	型
-	-

Example

```
static byte[] createCommandsRasterModeCoupon(Resources resource ) {
    CommandDataList commands = new CommandDataList();

    Bitmap bitmap = BitmapFactory.decodeResource( resource, R.drawable.coupon_image);

    ISCBBuilder builder = SCBFactory.createBuilder(SCBFactory.Emulation.Star);

    builder.appendBitmap(bitmap, false, 384, false, SCBBitmapConverter.Rotation.Right90);

    List<byte[]> listBuf = builder.getBuffer();

    for(byte[] buf:listBuf) {
        commands.add(buf);
    }

    return commands.getByteArray();
}
```

PrinterFunctions.java を参照ください。

6.2 appendAbsolutePositionBitmap メソッド

絶対位置指定のビットマップの印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
void appendAbsolutePositionBitmap(Bitmap bitmap, boolean diffusion, int width, boolean bothScale, SCBBitmapConverter.Rotation rotation, int position);
```

```
void appendAbsolutePositionBitmap(Bitmap bitmap, boolean diffusion, int width, SCBBitmapConverter.Rotation rotation, int position);
```

```
void appendAbsolutePositionBitmap(Bitmap bitmap, boolean diffusion, SCBBitmapConverter.Rotation rotation, int position);
```

```
void appendAbsolutePositionBitmap(Bitmap bitmap, boolean diffusion, int width, boolean bothScale, int position);
```

```
void appendAbsolutePositionBitmap(Bitmap bitmap, boolean diffusion, int width, int position);
```

```
void appendAbsolutePositionBitmap(Bitmap bitmap, boolean diffusion, int position);
```

引数

名称	説明	型
bitmap	元となるビットマップオブジェクト	Bitmap
diffusion	誤差拡散 <ul style="list-style-type: none"> true ... 誤差拡散を行う false ... 誤差拡散を行わない 	boolean
width	変換後のビットマップ幅（ドット数単位）	int
bothScale	width プロパティによる変換率に合わせた高さ変換 <ul style="list-style-type: none"> true ... 高さ変換を行う false ... 高さ変換を行わない 	boolean
rotation	変換後の回転 <ul style="list-style-type: none"> Normal ... 回転なし Right90 ... 右 90 度回転 Left90 ... 左 90 度回転 Rotate180 ... 180 度回転 	SCBBitmapConverter.Rotation
position	絶対位置指定（ドット数単位）	int

戻り値

説明	型
-	-

Example

```
static byte[] createCommandsRasterModeCoupon(Resources resource ) {  
    CommandDataList commands = new CommandDataList();  
  
    Bitmap bitmap = BitmapFactory.decodeResource( resource, R.drawable.coupon_image);  
  
    ISCBBuilder builder = SCBFactory.createBuilder(SCBFactory.Emulation.Star);  
  
    builder.appendAbsolutePositionBitmap(bitmap, false, 384, false, SCBBitmapConverter.Rotation.Right90, 100);  
  
    List<byte[]> listBuf = builder.getBuffer();  
  
    for(byte[] buf:listBuf) {  
        commands.add(buf);  
    }  
  
    return commands.getByteArray();  
}
```

6.3 appendAlignmentBitmap メソッド

位置揃えビットマップの印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
void appendAlignmentBitmap(Bitmap bitmap, boolean diffusion, int width, boolean bothScale, SCBBitmapConverter.Rotation rotation, SCBAlignment.Position position);
```

```
void appendAlignmentBitmap(Bitmap bitmap, boolean diffusion, int width, SCBBitmapConverter.Rotation rotation, SCBAlignment.Position position);
```

```
void appendAlignmentBitmap(Bitmap bitmap, boolean diffusion, SCBBitmapConverter.Rotation rotation, SCBAlignment.Position position);
```

```
void appendAlignmentBitmap(Bitmap bitmap, boolean diffusion, int width, boolean bothScale, SCBAlignment.Position position);
```

```
void appendAlignmentBitmap(Bitmap bitmap, boolean diffusion, int width, SCBAlignment.Position position);
```

```
void appendAlignmentBitmap(Bitmap bitmap, boolean diffusion, SCBAlignment.Position position);
```

引数

名称	説明	型
bitmap	元となるビットマップオブジェクト	Bitmap
diffusion	誤差拡散 <ul style="list-style-type: none"> true ... 誤差拡散を行う false ... 誤差拡散を行わない 	boolean
width	変換後のビットマップ幅（ドット数単位）	int
bothScale	width プロパティによる変換率に合わせた高さ変換 <ul style="list-style-type: none"> true ... 高さ変換を行う false ... 高さ変換を行わない 	boolean
rotation	変換後の回転 <ul style="list-style-type: none"> Normal ... 回転なし Right90 ... 右 90 度回転 Left90 ... 左 90 度回転 Rotate180 ... 180 度回転 	SCBBitmapConverter.Rotation
position	位置揃え指定 <ul style="list-style-type: none"> Left ... 左揃え Center ... 中心揃え Right ... 右揃え 	SCBAlignment.Position

戻り値

説明	型
-	-

Example

```
static byte[] createCommandsRasterModeCoupon(Resources resource ) {
    CommandDataList commands = new CommandDataList();

    Bitmap bitmap = BitmapFactory.decodeResource( resource, R.drawable.coupon_image);

    ISCBBuilder builder = SCBFactory.createBuilder(SCBFactory.Emulation.Star);

    builder.appendAlignmentBitmap(bitmap, false, 384, false, SCBBitmapConverter.Rotation.Right90,
SCBAAlignment.Position.Right);

    List<byte[]> listBuf = builder.getBuffer();

    for(byte[] buf:listBuf) {
        commands.add(buf);
    }

    return commands.getByteArray();
}
```

6.4 getBuffer メソッド

commands プロパティを取得します。

宣言

```
List<byte[]> getBuffer();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
command プロパティ	List<byte[]>-

Example

```
static byte[] createCommandsRasterModeCoupon(Resources resource ) {
    CommandDataList commands = new CommandDataList();

    Bitmap bitmap = BitmapFactory.decodeResource( resource, R.drawable.coupon_image);

    ISCBBuilder builder = SCBFactory.createBuilder(SCBFactory.Emulation.Star);

    builder.appendAlignmentBitmap(bitmap, false, 384, false, SCBBitmapConverter.Rotation.Right90,
    SCBAAlignment.Position.Right);

    List<byte[]> listBuf = builder.getBuffer();

    for(byte[] buf:listBuf) {
        commands.add(buf);
    }

    return commands.getByteArray();
}
```

PrinterFunctions.java を参照ください。

7 SCBAlignment

コンスタント

名称	説明
Position	位置揃え指定定数。

7.1 Position コンスタント

位置揃え指定定数。

宣言

```
public enum Position {
    Left,
    Center,
    Right
}
```

定数

名称	説明
Left	左揃え
Center	中心揃え
Right	右揃え

8 SCBBitmapConverter

コンスタント

名称	説明
Rotation	回転指定定数。

8.1 Rotation コンスタント

回転指定定数。

```
public enum Rotation {
    Normal,
    Right90,
    Left90,
    Rotate180
}
```

定数

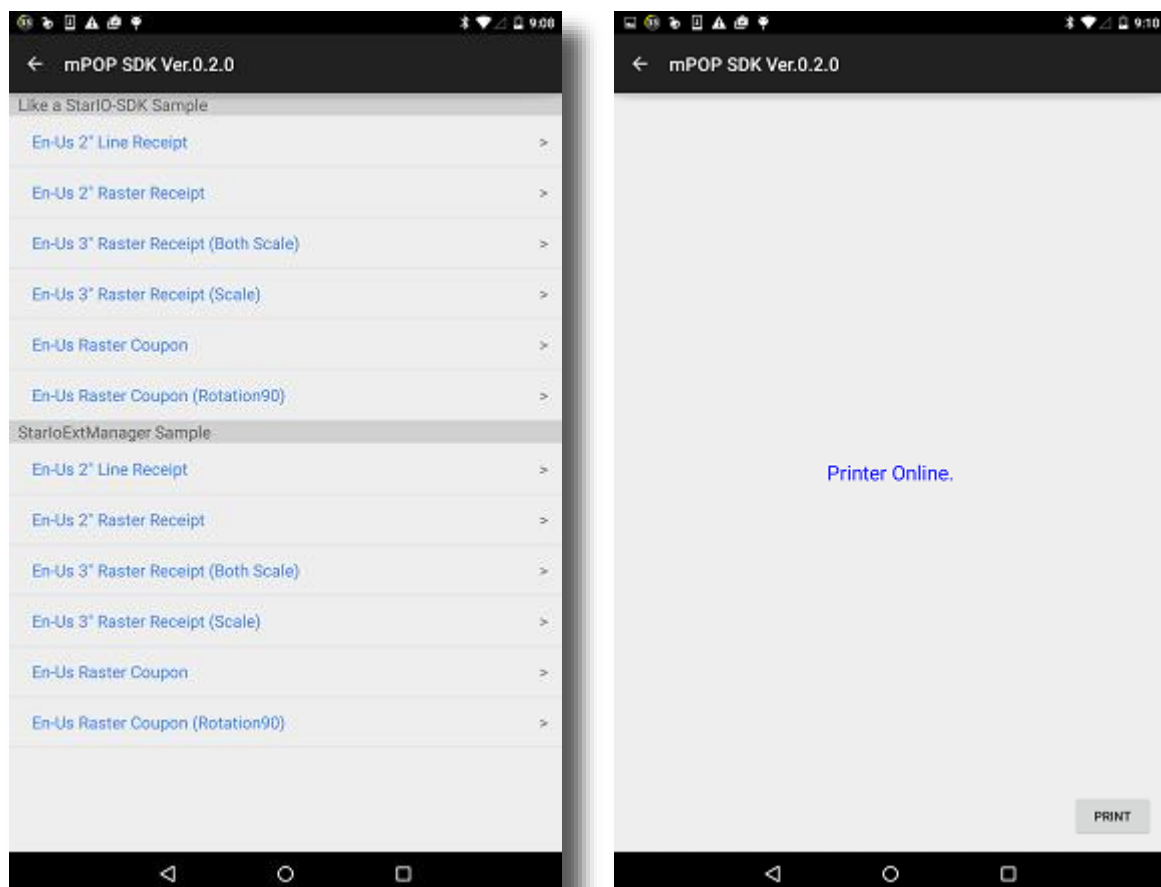
名称	説明
Normal	回転なし
Right90	右 90 度回転
Left90	左 90 度回転
Rotate180	180 度回転

9 Star mPOP Android SDK 機能

9.1 PrinterFragment.java / PrintExtFragment.java

PrinterFragment.java にてレシートサンプルの選択を行います。

そして PrintExtFragment.java にて StarIoExtManager.Type.Standard タイプの StarIoExtManager オブジェクトを使用した印刷およびリスナーによる監視を行います。



Initialize of StarIoExtManager

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    ...

    PrinterSetting setting = new PrinterSetting(getActivity());

    mStarIoExtManager = new StarIoExtManager(StarIoExtManager.Type.Standard, setting.getPortName(),
    setting.getPrinterType(), 10000, getActivity()); // 10000mS!!!

    mStarIoExtManager.setListener(mStarIoExtManagerListener);

    ...
}
```

Connect

```
public void onResume() {
    super.onResume();

    ...

    mStarloExtManagerListener.didAccessoryDisconnect();

    AsyncTask<Void, Void, Boolean> asyncTask = new AsyncTask<Void, Void, Boolean>() {
        @Override
        protected void onPreExecute() {

            ...

        }

        @Override
        protected Boolean doInBackground(Void... params) {
            return mStarloExtManager.connect();
        }

        @Override
        protected void onPostExecute(Boolean result) {

            ...

            if (! result) {
                AlertDialog.Builder dialogBuilder = new AlertDialog.Builder(getActivity());

                dialogBuilder.setTitle("Communication Result");
                dialogBuilder.setMessage("Fail to Open Port.");

                dialogBuilder.setPositiveButton("OK", null);

                dialogBuilder.show();
            }
        }
    };

    asyncTask.execute();

    ...
}
```

Disconnect

```
public void onPause() {
    super.onPause();

    ...

    mStarloExtManager.disconnect();

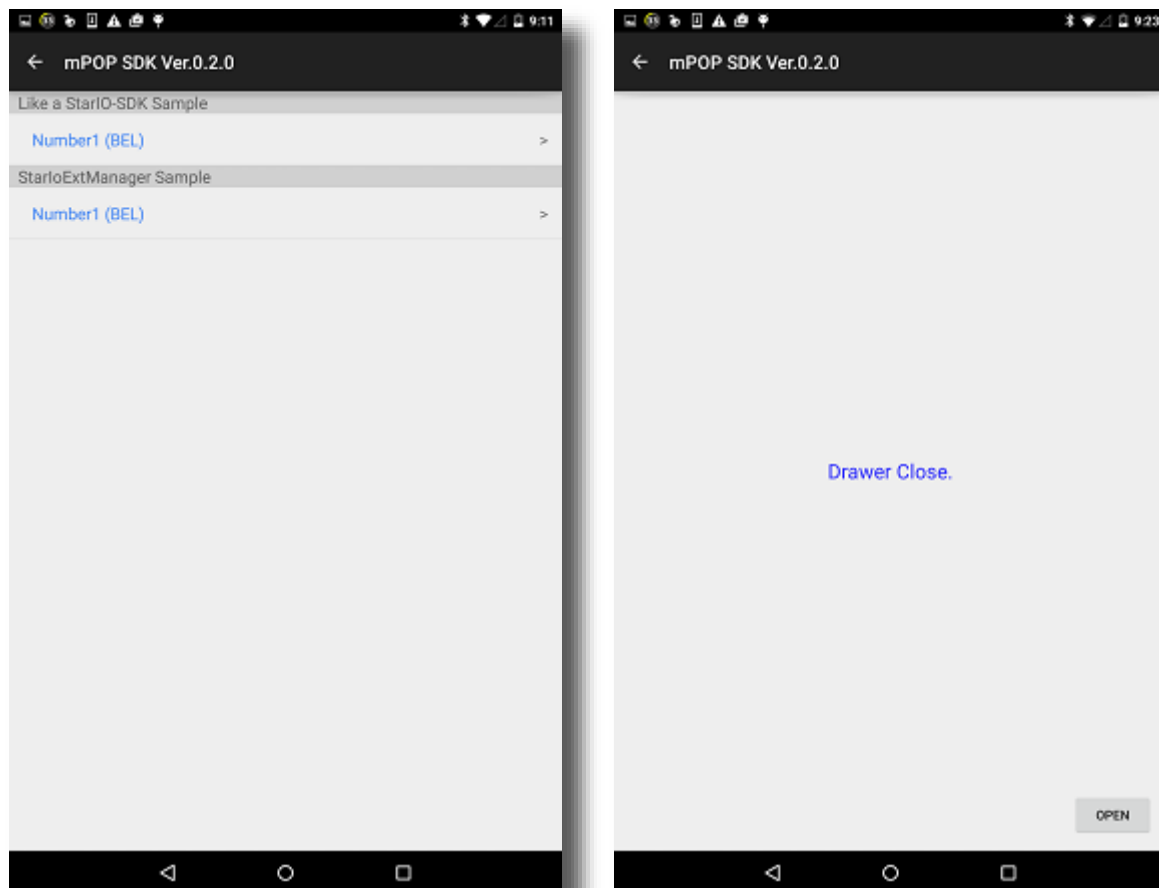
    ...
}
```

Send Data (Print)

```
private void sendCommands() {  
    ...  
    Communication.Result result;  
    synchronized (mStarIoExtManager) {  
        result = Communication.sendCommands(data, mStarIoExtManager.getPort(), getActivity()); // 10000mS!!!  
    }  
    ...  
}
```

9.2 CashDrawerFragment.java / CashDrawerExtFragment.java

CashDrawerExtFragment.java にて StarloExtManager.Type.Standard タイプの StarloExtManager オブジェクトを使用したキャッシュドロアーオープンおよびリスナーによる監視を行います。



Initialize of StarloExtManager

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    ...

    PrinterSetting setting = new PrinterSetting(getActivity());

    mStarloExtManager = new StarloExtManager(StarloExtManager.Type.Standard, setting.getPortName(),
    setting.getPrinterType(), 10000, getActivity()); // 10000mS!!!

    mStarloExtManager.setListener(mStarloExtManagerListener);

    ...
}
```

Connect

```
public void onResume() {
    super.onResume();

    ...

    mStarIoExtManagerListener.didAccessoryDisconnect();

    AsyncTask<Void, Void, Boolean> asyncTask = new AsyncTask<Void, Void, Boolean>() {
        @Override
        protected void onPreExecute() {

            ...

        }

        @Override
        protected Boolean doInBackground(Void... params) {
            return mStarIoExtManager.connect();
        }

        @Override
        protected void onPostExecute(Boolean result) {

            ...

            if (! result) {
                AlertDialog.Builder dialogBuilder = new AlertDialog.Builder(getActivity());

                dialogBuilder.setTitle("Communication Result");
                dialogBuilder.setMessage("Fail to Open Port.");

                dialogBuilder.setPositiveButton("OK", null);

                dialogBuilder.show();
            }
        }
    };

    asyncTask.execute();

    ...
}
```

Disconnect

```
public void onPause() {
    super.onPause();

    ...

    mStarIoExtManager.disconnect();

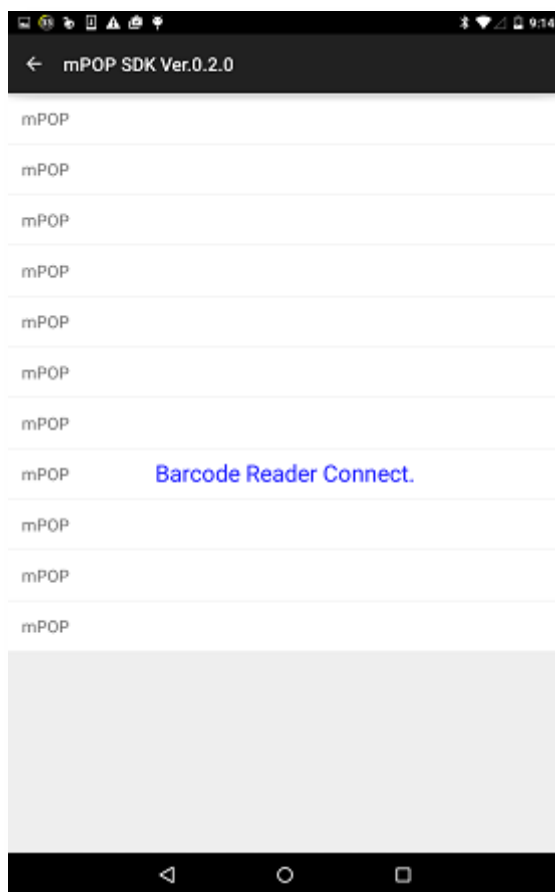
    ...
}
```

Send Data (Open Cash Drawer)

```
private void sendCommands() {  
    ...  
    Communication.Result result;  
    synchronized (mStarIoExtManager) {  
        result = Communication.sendCommands(data, mStarIoExtManager.getPort(), getActivity()); // 10000mS!!!  
    }  
    ...  
}
```

9.3 BarcodeExtFragment.java

BarcodeExtFragment.java にて StarloExtManager.Type.OnlyBarcodeReader タイプの StarloExtManager オブジェクトを使用したバーコード読み取りおよびリスナーによる監視を行います。



Initialize of StarloExtManager

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    ...

    PrinterSetting setting = new PrinterSetting(getActivity());

    mStarloExtManager = new StarloExtManager(StarloExtManager.Type.OnlyBarcodeReader,
    setting.getPortName(), setting.getPrinterType(), 10000, getActivity()); // 10000mS!!!

    mStarloExtManager.setListener(mStarloExtManagerListener);

    ...
}
```


Connect

```
public void onResume() {
    super.onResume();

    ...

    mStarIoExtManagerListener.didAccessoryDisconnect();

    AsyncTask<Void, Void, Boolean> asyncTask = new AsyncTask<Void, Void, Boolean>() {
        @Override
        protected void onPreExecute() {

            ...

        }

        @Override
        protected Boolean doInBackground(Void... params) {
            return mStarIoExtManager.connect();
        }

        @Override
        protected void onPostExecute(Boolean result) {

            ...

            if (! result) {
                AlertDialog.Builder dialogBuilder = new AlertDialog.Builder(getActivity());

                dialogBuilder.setTitle("Communication Result");
                dialogBuilder.setMessage("Fail to Open Port.");

                dialogBuilder.setPositiveButton("OK", null);

                dialogBuilder.show();
            }
        }
    };

    asyncTask.execute();

    ...
}
```

Disconnect

```
public void onPause() {
    super.onPause();

    ...

    mStarIoExtManager.disconnect();

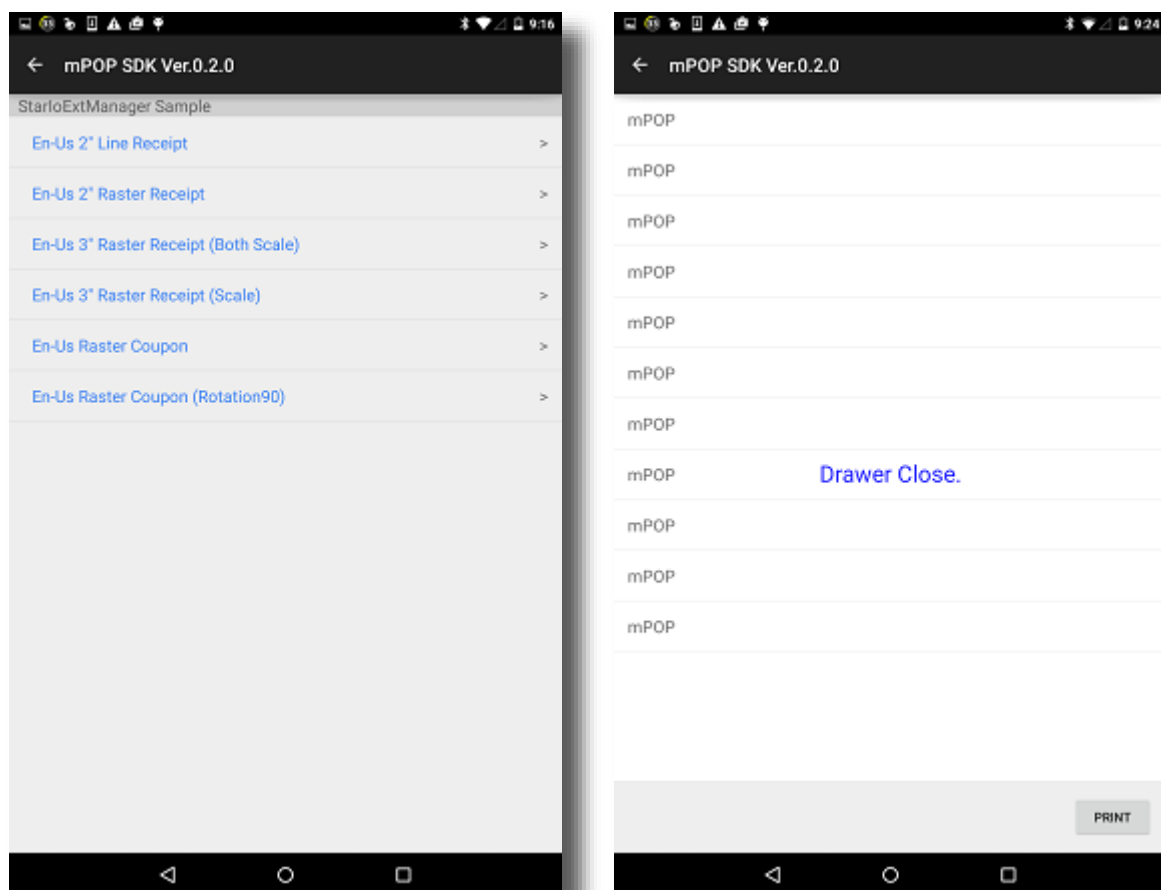
    ...
}
```

Receive Barcode Data リスナー

```
StarloExtManagerListener mStarloExtManagerListener = new StarloExtManagerListener() {  
  
    ...  
  
    @Override  
    public void didBarcodeDataReceive(byte[] data) {  
        String[] barcodeDataArray = new String(data).split("\n");  
  
        for(String barcodeData:barcodeDataArray) {  
  
            ...  
        }  
    }  
  
    ...  
  
};
```

9.4 CombinationFragment.java / CombinationExtFragment.java

CombinationFragment.java にてレシートサンプルの選択を行います。
 そして CombinationExtFragment.java にて StarloExtManager.Type.WithBarcodeReader タイプの StarloExtManager オブジェクトを使用した印刷 / キャッシュドロアーオープン / バーコード読み取りおよびリスナーによる監視を行います。



Initialize of StarloExtManager

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    ...

    PrinterSetting setting = new PrinterSetting(getActivity());

    mStarloExtManager = new StarloExtManager(StarloExtManager.Type.WithBarcodeReader,
    setting.getPortName(), setting.getPrinterType(), 10000, getActivity()); // 10000mS!!!

    mStarloExtManager.setListener(mStarloExtManagerListener);
    ...
}
```

Connect

```
public void onResume() {
    super.onResume();

    ...

    mStarIoExtManagerListener.didAccessoryDisconnect();

    AsyncTask<Void, Void, Boolean> asyncTask = new AsyncTask<Void, Void, Boolean>() {
        @Override
        protected void onPreExecute() {

            ...

        }

        @Override
        protected Boolean doInBackground(Void... params) {
            return mStarIoExtManager.connect();
        }

        @Override
        protected void onPostExecute(Boolean result) {

            ...

            if (! result) {
                AlertDialog.Builder dialogBuilder = new AlertDialog.Builder(getActivity());

                dialogBuilder.setTitle("Communication Result");
                dialogBuilder.setMessage("Fail to Open Port.");

                dialogBuilder.setPositiveButton("OK", null);

                dialogBuilder.show();
            }
        }
    };

    asyncTask.execute();

    ...
}
```

Disconnect

```
public void onPause() {
    super.onPause();

    ...

    mStarIoExtManager.disconnect();

    ...
}
```

Send Data (Print and Open Cash Drawer)

```
private void sendCommands() {
    ...

    Communication.Result result;

    synchronized (mStarloExtManager) {
        result = Communication.sendCommands(data, mStarloExtManager.getPort(), getActivity()); // 10000mS!!!
    }

    ...
}
```

Receive Barcode Data リスナー

```
StarloExtManagerListener mStarloExtManagerListener = new StarloExtManagerListener() {

    ...

    @Override
    public void didBarcodeDataReceive(byte[] data) {
        String[] barcodeDataArray = new String(data).split("\n");

        for(String barcodeData:barcodeDataArray) {

            ...
        }
    }

    ...
};
```

9.5 Communication

Communication にて mPOP との通信を行います。

StarIoExtManager オブジェクトを使用した印刷データ送信と StarIO Android SDK 相当の印刷データ送信の一例が用意されています。

Example (StarIoExtManager オブジェクトを使用)

```
public static Result sendCommands(byte[] commands, StarIOPort port, Context context) {
    Result result = Result.ErrorUnknown;

    try {
        // try {
        //     Thread.sleep(100);
        // } catch (InterruptedException e) {
        // }

        StarPrinterStatus status;

        result = Result.ErrorBeginCheckedBlock;

        status = port.beginCheckedBlock();

        if (status.offline) {
            throw new StarIOPortException("A printer is offline");
        }

        result = Result.ErrorWritePort;

        port.writePort(commands, 0, commands.length);

        result = Result.ErrorEndCheckedBlock;

        port.setEndCheckedBlockTimeoutMillis(30000); // 30000mS!!!

        status = port.endCheckedBlock();

        if (status.coverOpen) {
            throw new StarIOPortException("Printer cover is open");
        }
        else if (status.receiptPaperEmpty) {
            throw new StarIOPortException("Receipt paper is empty");
        }
        else if (status.offline) {
            throw new StarIOPortException("Printer is offline");
        }

        result = Result.Success;
    }
    catch (StarIOPortException e) {
        ...
    }

    return result;
}
```

Example (StarIO Android SDK 相当)

```
public static Result sendCommands(byte[] commands, String portName, String portSettings, int timeout, Context
context) {
    Result result = Result.ErrorUnknown;

    StarIOPort port = null;

    try {
        result = Result.ErrorOpenPort;

        port = StarIOPort.getPort(portName, portSettings, timeout, context);

        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
        }

        StarPrinterStatus status;

        result = Result.ErrorBeginCheckedBlock;

        status = port.beginCheckedBlock();

        if (status.offline) {
            throw new StarIOPortException("A printer is offline");
        }

        result = Result.ErrorWritePort;

        port.writePort(commands, 0, commands.length);

        result = Result.ErrorEndCheckedBlock;

        port.setEndCheckedBlockTimeoutMillis(30000); // 30000mS!!!

        status = port.endCheckedBlock();

        if (status.coverOpen) {
            throw new StarIOPortException("Printer cover is open");
        }
        else if (status.receiptPaperEmpty) {
            throw new StarIOPortException("Receipt paper is empty");
        }
        else if (status.offline) {
            throw new StarIOPortException("Printer is offline");
        }

        result = Result.Success;
    }
    catch (StarIOPortException e) {
        ...
    }
    finally {
        if (port != null) {
            try {
                StarIOPort.releasePort(port);

                port = null;
            }
            catch (StarIOPortException e) {
            }
        }
    }
}
```

```
}  
    return result;  
}
```

StarIO.framework 詳細を知りたい場合、StarIO Android SDK をダウンロードし、ドキュメントを参照してください。