

cokatielシステム

# SmartAudio16

NativeActivityガイド

2013.02.21

株式会社ピュアサウンド

## 目次

<b>1.はじめに</b> .....	3
<b>2.準備(NativeActivity版)</b> .....	4
2.1.プロジェクトへの登録 .....	4
2.2.定義されている型 .....	6
<b>3.SmartAudio16の初期化(NativeActivity版)</b> .....	7
<b>4.割り込み時の挙動について(NativeActivity版)</b> .....	8
4.1.通常時の挙動 .....	8
4.2.割り込み発生時の挙動とその対応 .....	8
<b>改訂履歴</b> .....	9

## 1.はじめに

この文書は補足資料として、『cockatielシステム SmartAudio16(以下SmartAudio16)』をAndroidのNativeActivity上で運用するための手順を紹介します。

## 2.準備(NativeActivity版)

SmartAudio16をNativeActivity上で使用するためには、ライブラリ本体、ライブラリヘッダ、リソースファイルの三つのファイルが必要です。

リソースファイルはiOSとAndroidOS(Java)とで共通の『spkgサウンドパッケージファイル(以下サウンドパッケージファイル)』を使用します。

### 2.1.プロジェクトへの登録

AndroidOS(NativeActivity)では、次の三つのファイルを使用します。

ライブラリ本体: libSmartAudio16Library.a(静的ライブラリファイル)

ライブラリヘッダ: SmartAudio16\_android\_na.h

リソースファイル: spkgサウンドパッケージファイル

- 1) ライブラリ本体、クラスライブラリ、リソースファイルをプロジェクトに登録します。

[フォルダ配置例]

```
PSSsampleNA/
  +---assets ...このフォルダにサウンドパッケージファイルを登録します。
  +---jni/
    |      +---external/
    |      |              +---include/
    |      |              |              +---SmartAudio16_android_na.h
    |      |              +---lib/
    |      |              +---libSmartAudio16Library.a
    |      +---Android.mk
  +---AndroidManifest.xml
```

以降は、上記配置例に沿って説明をします。

- 2) NativeActivity用にAndroidManifest.xmlの編集をします。

manifestタグ内に以下の内容を追加してください。

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

SmartAudio16は内部的にオーディオ設定を変更するため、オーディオ設定を変更するアプリケーションを許可する上記設定が必要となります。

- 3) メインとなるC言語のソースコード(main.cなど)内にライブラリヘッダをインクルードします。

```
#include "SmartAudio16_android_na.h"
```

- 4) ここからは、Android.mk(makeファイル)内でライブラリのコンパイルおよびリンク指定を行います。まず、LOCAL\_PATH設定以降の行に以下の内容を追加します。

```
include $(CLEAR_VARS)
LOCAL_MODULE := prebuild-libSmartAudio16
LOCAL_SRC_FILES := external/lib/libSmartAudio16Library.a
include $(PREBUILT_STATIC_LIBRARY)
```

LOCAL\_SRC\_FILESではライブラリファイルのパスを指定しています。

- 5) 次に、NativeActivityのコンパイル設定部分に内容を追加をします。下記のように、ヘッダファイルがあるパスを指定します。

```
LOCAL_C_INCLUDES := jni/external/include
```

- 6) さらに、SmartAudio16が内部処理で使用しているOpenSLライブラリを下記のように追加します。

```
LOCAL_LDLIBS := 〇〇 〇〇 -lOpenSLES
```

※"〇〇"は、アプリケーションで必要とするOpenSL以外のライブラリです。

- 7) 最後に、NativeActivityのライブラリと、SmartAudio16のライブラリ同士のリンク設定を行います。以下の内容を"LOCAL\_STATIC\_LIBRARIES"が定義された行から、"include \$(BUILD\_SHARED\_LIBRARY)"が記述された行までの間に追記します。

```
LOCAL_STATIC_LIBRARIES += prebuild-libSmartAudio16
```

これで、NativeActivity版のSmartAudio16をプロジェクト内で使用するための設定は完了です。

一連の記述を行うと、Android.mkファイル内部は以下のような内容となります。"#"のついた行が、初期状態のAndroid.mkから追記・変更を行った箇所です。

```
LOCAL_PATH := $(call my-dir)

# SmartAudio16 library compile(copy)
include $(CLEAR_VARS) #
LOCAL_MODULE := prebuild-libSmartAudio16 #
LOCAL_SRC_FILES := external/lib/libSmartAudio16Library.a #
include $(PREBUILT_STATIC_LIBRARY) #

# Native-Activity library compile
```

```

include $(CLEAR_VARS)
LOCAL_MODULE := native-activity
LOCAL_C_INCLUDES := jni/external/include #
LOCAL_SRC_FILES := main.c
LOCAL_LDLIBS := -llog -landroid -IEGL -IGLESv1_CM -lOpenSLES #
LOCAL_STATIC_LIBRARIES := android_native_app_glue

# Library Link Setting
LOCAL_STATIC_LIBRARIES += prebuild-libSmartAudio16 #
include $(BUILD_SHARED_LIBRARY)

$(call import-module, android/native_app_glue)

```

## 2.2.定義されている型

AndroidOS(NativeActivity)では、あらかじめ整数の型が定義されています。

定義された型名	内容	値の範囲
SInt8	符号付き8bit値	-128~127
SInt16	符号付き16bit値	-32768~32767
SInt32	符号付き32bit値	-2147483648~2147483647
UInt8	符号なし8bit値	0~255
UInt16	符号なし16bit値	0~65535
UInt32	符号なし32bit値	0~4294967295

### 3.SmartAudio16の初期化(NativeActivity版)

準備が完了したら、実際にSmartAudio16を使用するために初期化を行います。

初期化をするためには、次の関数を呼び出します。

```
SInt32 cktlInit(ANativeActivity *activity,  
               UInt32      heapSize,  
               SInt8      player,  
               const char* filePath,  
               bool       debugFlag);
```

基本的にはAndroid(Java)版、iOS版と同様の記述なのですが、引数『ANativeActivity』を必要とする点に注意してください。

引数『ANativeActivity』には、NativeActivityのメイン関数となる『android\_main関数』で渡されるネイティブ定義構造体『state』のメンバ変数『activity』を指定します。

具体的な初期化は、以下のような記述となります。

```
SInt32 error = cktlInit(state->activity, 5000000, 8, "spkg.spkg", false);
```

※引数activityは、android\_main関数で渡されるstate変数から参照する。

## 4.割り込み時の挙動について(NativeActivity版)

SmartAudio16(NativeActivity)における、着信などの割り込み時の挙動を説明します。

### 4.1.通常時の挙動

SmartAudio16(Java)と同じ挙動になります。割り込みが発生していない状態では、特に留意する点はありません。

### 4.2.割り込み発生時の挙動とその対応

アプリケーションがバックグラウンド状態に移行する場合の挙動も、SmartAudio16(Java)と同様です。そのため、アプリケーション制作者が適切なタイミングでSmartAudio16の出力を停止・再開させる必要があります。

以下は出力停止用のメソッドです。

```
SInt32 cktlSuspendSystem();
```

また、以下は出力再開用のメソッドです。

```
SInt32 cktlResumeSystem();
```

## 改訂履歴

---

日付	内容
2013.2.21	2.1.プロジェクトへの登録 でのmakeファイルの内容に補足
2013.2.4	ドキュメント改訂。
2012.8.4	新規ドキュメント作成。

※ 「iPhone」「iPad」「Objective-C」はApple Inc.の米国およびその他の国における登録商標です。

また「IOS」の商標は、Ciscoの米国およびその他の国のライセンスに基づき使用されています。

※ 「Android」はGoogle Inc.の商標または登録商標です。

※ 「Java」は、Oracle America, Inc. の商標です。

※ その他記載されている会社名、製品名は各社の登録商標または商標です。