

cokatielシステム

SmartAudio16

ver. 1.4.x
イントロダクション

目次

| | |
|--------------------------------------|----|
| 1.はじめに | 3 |
| 2.仕様・特徴 | 4 |
| 3.動作条件 | 6 |
| 3.1.iOS版 | 6 |
| 3.2.AndroidOS(Java)版 | 6 |
| 3.3.AndroidOS(NativeActivity)版 | 6 |
| 4.spkgサウンドパッケージファイルについて | 7 |
| 4.1.オーディオデータ | 7 |
| 4.2.インストゥルメンツ | 7 |
| 4.3.バンク | 7 |
| 4.4.シーケンスデータ | 7 |
| 4.5.コールラベル | 8 |
| 4.6.グループ | 8 |
| 5.シーケンスプレイヤーについて | 9 |
| 5.1.シーケンスプレイヤーとは | 9 |
| 5.2.シーケンスプレイヤーの最大数 | 9 |
| 5.3.シーケンス同時演奏数の制限 | 9 |
| 5.4.トラックについて | 9 |
| 6.プライオリティについて | 11 |
| 6.1.プライオリティとは | 11 |
| 6.2.シーケンスプライオリティによるプライオリティ判定 | 11 |
| 6.3.トラックプライオリティによるプライオリティ判定 | 12 |
| 7.メモリ管理について | 13 |
| 7.1.サウンドヒープ | 13 |
| 7.2.スタック方式 | 13 |
| 7.3.レイヤー数の制限 | 14 |
| 7.4.実際の運用例 | 14 |
| 8.資料 | 15 |
| 改訂履歴 | 16 |

1.はじめに

『cockatielシステム SmartAudio16(以下SmartAudio16)』は、iOS、AndroidOSが搭載された端末上でサウンド演奏を行なうためのライブラリです。

『独自で構築した音源を、独自シーケンスを使用して演奏する』という点において、OS標準のサウンド演奏用サービスとは大きく異なります。

SmartAudio16は旧来の家庭用ゲーム機や携帯ゲーム機、また着信メロディなどと同様の設計思想を持っており、少ないリソース容量でありながら多彩なサウンドの演奏を実現します。

また、SmartAudio16で使用するメモリ領域のサイズは初期化時に任意のサイズを指定できるため、プロジェクトごとに柔軟なメモリ設計が可能です。

特に大量のサウンドを必要とする大規模なゲームアプリケーションや、発音数管理やメモリ管理が煩雑で、プログラマーの負担となるようなアクション性、インタラクティブ性の強いアプリケーションのサウンド開発において効果を発揮します。

2.仕様・特徴

SmartAudio16は、iOSやAndroidOSの標準サウンド演奏サービスにはない、次の特徴を備えています。

・シーケンスによる演奏

SmartAudio16では、スタンダードMIDIファイルを圧縮した非常にコンパクトなシーケンスデータと、音源データを使用して楽曲を演奏します。

演奏に必要な全てのサウンドデータは、『spkgサウンドパッケージファイル』という1つのファイルにまとめられるので、プログラマーは大量のサウンドデータのファイル管理に煩わされる必要がありません。

・最大16個のシーケンスを同時演奏可能

SmartAudio16は、シーケンスを演奏するための『シーケンスプレイヤー』という仕組みを備えています。シーケンスの演奏を開始する際、SmartAudio16がシーケンスに対して自動的に1つのシーケンスプレイヤーを割り当てます。

シーケンスプレイヤーは16個まで用意することができ、それぞれ並列して動作するため、最大で16個のシーケンスを同時演奏することが可能です。

・最大16音同時発音可能

たとえばシューティングゲームのように、短時間のうちに多数の効果音が鳴らされるアプリケーションの場合、効果音を際限なく演奏すると、メモリを予想以上に消費したり、CPUの負荷が過剰となります。CPUの負荷を抑えるために、同時に演奏される効果音の数を制限することが考えられますが、そのような処理を追加することは、プログラマーにとって煩雑な作業であるといえます。

SmartAudio16では、あらかじめ最大の同時発音数を16音に制限しており、後述するプライオリティにより自動的に発音数管理を行うので、プログラマーもサウンド制作者も、シビアに発音数を管理する必要はありません。

・プライオリティを用いた発音制御

SmartAudio16は、シーケンスや発音数の最大値を超えた場合、どのサウンドを優先して演奏するかを決めるための『プライオリティ(優先度)』という仕組みを備えています。

プライオリティはサウンド制作者が任意に設定することができます。

そのプライオリティ設定に基づき、SmartAudio16が自動的に発音の管理を行うので、プログラマーは優先度を一切意識せずに最適なサウンド演奏が行われます。

プライオリティには2タイプあり、シーケンスプライオリティはシーケンスプレイヤーの同時演奏数の制御に、トラックプライオリティは16音という発音数の制御に使用されます。

・簡潔で安心なメモリ管理

SmartAudio16は、初期化時に連続した動的メモリ領域を『サウンドヒープ』として確保します。

SmartAudio16: イントロダクション

演奏のために必要なデータはすべてサウンドヒープ内に配置されるので、サウンドヒープ外の領域へ無作為に、また予期しない場所に配置されることはありません。

またメモリの使用状況を確認するためのメソッドが用意されているので、メモリ管理も容易です。

・ 大量のシーケンスも手軽に一括ロード

サウンドヒープへのデータのロードは、複数のシーケンスと、シーケンスの演奏に必要なオーディオデータを一かたまりにした『グループ』を単位として行います。

たとえ100個を超える効果音でも、ひとつのグループにまとめれば、たった一回のメソッド呼び出しで演奏に必要なすべてのデータをサウンドヒープへロードすることができます。

・ 既存のMIDIファイルやオーディオファイルからデータ作成可能

近日公開予定の専用オーサリングツールを使用することにより、spkgサウンドパッケージファイルを独自に作成することが可能です。

spkgサウンドパッケージファイルはスタンダードMIDIファイルやaif、wavファイルといった一般的なデータを使用して作成するので、着信メロディやゲーム機等で使用した過去のリソースを活かすことが出来ます。

またADPCM形式のwavファイルにも対応しています(注1)。

データサイズは、シーケンスデータでは通常のMIDIファイルの約50%、オーディオデータではADPCM使用時に限りaifの約25%のサイズに削減されます(注2)。

(注1. ただし、サウンドヒープへのロード時に、リニアPCM形式にデコードされます。)

(注2. シーケンスデータの容量削減効率はデータ量が多ければ多いほど上がります。

そのため、極端に短いシーケンスデータでは容量削減効果が得られません。)

・ 安定性の高いCPU負荷

SmartAudio16の1秒間のCPU使用率平均値は、通常9%程度です。(注)

OS標準のサウンド演奏サービスでは、同時演奏するオーディオデータの数が増えるほどCPU使用率も増大していきますが、SmartAudio16では同時発音数の制限内で処理を行うため、極端にCPU使用率が上昇することはありません。

そのため、アプリケーションを通じて安定した動作を実現します。

(注. iPhone4での計測値)

3.動作条件

SmartAudio16の動作環境を次に示します。

3.1.iOS版

機種: iPhone 3GS以降 / iPad (armv7アーキテクチャに対応したCPUを搭載している機種)

言語: Objective-C

OS: iOS4.0以上

3.2.AndroidOS(Java)版

機種: ARMアーキテクチャを採用しているCPUが搭載されたAndroidOS端末

言語: Java

OS: AndroidOS2.1以上(API level 7)

3.3.AndroidOS(NativeActivity)版

機種: ARMアーキテクチャを採用しているCPUが搭載されたAndroidOS端末

言語: C

OS: AndroidOS2.3.3以上(API level 10)

4.spkgサウンドパッケージファイルについて

SmartAudio16では、演奏に使用する全てのデータを『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)』という1つのファイルに収めます。

サウンド制作者はこのサウンドパッケージファイルを作成し、プログラマーに渡します。

プログラマーは受け取ったサウンドパッケージファイルをリソースとしてプロジェクトへ組み込み、ファイル内部に納められた各データを演奏するための手続きを行います。

サウンドパッケージファイルの構成要素を以下に紹介します。

4.1.オーディオデータ

オーディオデータは、楽曲を演奏する際の“音源”に相当し、オーディオ波形データを格納しています。音声信号のデータのみで構成されているので、オーディオデータ自身には何の音程か、何の楽器に相当するかといった情報は含まれていません。

鳴り方の設定は、後述のインストゥルメンツにて行います。

4.2.インストゥルメンツ

オーディオデータが音声信号そのものを格納しているのに対し、インストゥルメンツはオーディオデータの鳴り方の定義を格納しています。

具体的には楽器の定義やエンベロープ、キー情報などといった内容で構成されています。

4.3.バンク

バンクは、あるシーケンスにおいて使用されるインストゥルメンツを集めたデータ単位です。

複数のシーケンスで一つのバンクを共用することもできます。

バンクはサウンドパッケージファイル内に最大255個まで作成することができます。

シーケンスは演奏中に音色を切り替える際に、バンクの中から一つのインストゥルメンツを特定します。

バンク内の構成次第では、複数の音源を一つの楽器として扱うドラムキットのような構成も可能です。

通常、プログラマーはバンクを意識する必要はありません。

4.4.シーケンスデータ

シーケンスデータは、楽曲を演奏する際の“楽譜”に相当します。

どの楽器をどの音程で鳴らすか、どれくらいの長さで発音し続けるか、などといった演奏情報と、どのバンクを使用して演奏するかといった定義が収められています。

各シーケンスデータは最大16個のトラックで構成されますので、最大16個の楽器パートを同時に演奏することが可能です。

ただし、SmartAudio16全体の最大発音数は16音までです。

後述するコールラベルの要素によって、ひとつのシーケンスに対して複数の鳴らし方を設定することが可

能です。

4.5.コールラベル

コールラベルは、前述のシーケンスデータを、どのようなシーケンスプライオリティ、音量で演奏するか定義したものです。

コールラベルには、必ずユニークなID番号が与えられます。

SmartAudio16では、音楽、効果音の演奏開始処理は全てコールラベル番号を指定して実行されます。

4.6.グループ

グループは、ひとつもしくは複数のコールラベルをひとまとめにしたものです。

SmartAudio16では、メモリ上へのデータロードはすべてグループを単位として行います。

グループ番号を指定してロードを行った際には、グループに含まれるコールラベルが演奏に必要とする、すべてのシーケンスデータとオーディオデータが一括してメモリ上にロードされます。

5. シーケンスプレイヤーについて

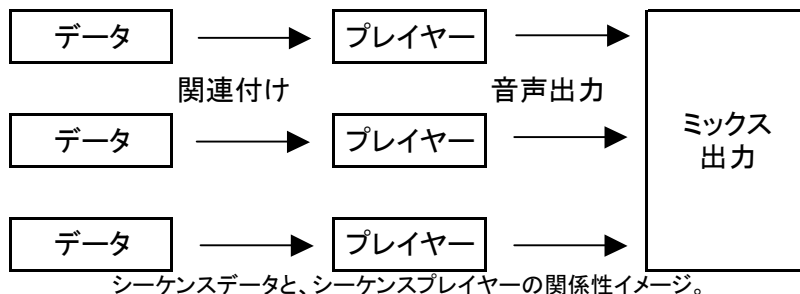
ここでは、シーケンスデータ演奏時に必要な『シーケンスプレイヤー』について解説をします。

5.1. シーケンスプレイヤーとは

SmartAudio16では、シーケンスデータを演奏するための『シーケンスプレイヤー』という仕組みが用意されています。

シーケンスデータとシーケンスプレイヤーの関係は、CDとCDプレイヤーの関係に似ており、1つのシーケンスプレイヤーで、同時に1つのシーケンスデータを演奏することが出来ます。

それぞれのシーケンスプレイヤーは並列して演奏を行えるので、シーケンスプレイヤーの数だけシーケンスデータを同時演奏することが可能だといえます。



どのシーケンスプレイヤーを用いてシーケンスデータを演奏するかは、演奏開始時にSmartAudio16が自動的に割り当てを行いますので、プログラマーが意識する必要はありません。

シーケンスデータの演奏が終了、もしくはプログラム側で演奏停止をした際には、シーケンスプレイヤーは自動的に解放されます。

5.2. シーケンスプレイヤーの最大数

シーケンスプレイヤーの最大数は1～16の範囲で、SmartAudio16の初期化時に指定することができます。

5.3. シーケンス同時演奏数の制限

SmartAudio16の初期化時に指定したシーケンスプレイヤーの最大数を超えてシーケンスを同時演奏しようとした場合は、後述するプライオリティ(優先度)によってどのシーケンスを優先して演奏するかが決まります。

プライオリティの高いシーケンスが優先的に演奏され、プライオリティの判定で負けたシーケンスは演奏されません。

5.4. トラックについて

各シーケンスデータには、スタンダードMIDIファイルと同様『トラック』という概念があります。

1つのシーケンスデータは一つもしくは複数のトラックで構成されています。

SmartAudio16: イントロダクション

1つのシーケンスデータでのトラック数は最大16です。

6. プライオリティについて

ここでは、複数のシーケンスの演奏要求、複数のボイスの発音要求があったときに、どれを優先するかを決める仕組み『プライオリティ』について説明します。

6.1. プライオリティとは

プライオリティ(優先度)とは、シーケンスもしくは発音数が許容の数を超えた場合に、どれを優先して演奏するかを判定するための値です。SmartAudio16では、2種類のプライオリティがあります。

- ・ **シーケンスプライオリティ**

シーケンス同時演奏数がSmartAudio16の初期化時に指定したシーケンスプレイヤーの最大数を超えた場合、どのシーケンスを優先して演奏するかを判定に使用します。

- ・ **トラックプライオリティ**

全シーケンスプレイヤーを合わせた同時発音数が、SmartAudio16で制限されている16音を超えた場合、各シーケンスのトラックごとに設定されたトラックプライオリティを比較して、どのトラックの音を優先して発音するかを判定を行います。

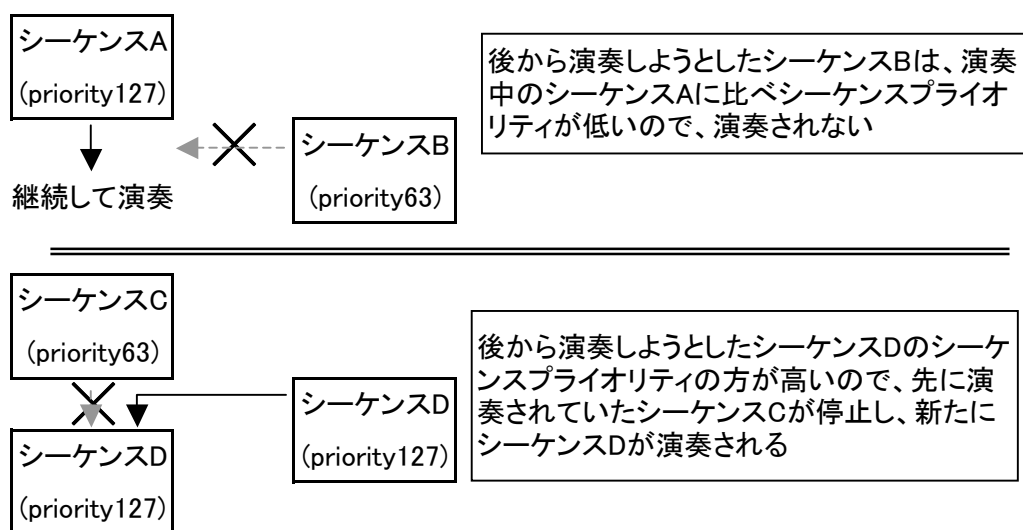
6.2. シーケンスプライオリティによるプライオリティ判定

シーケンスプレイヤーの最大数以上のシーケンスを同時演奏しようとした際、シーケンスプライオリティのチェックが行なわれます。

演奏中のシーケンスの中で一番プライオリティの低いシーケンスと、これから演奏しようとするシーケンスのプライオリティを比較して、プライオリティが高い方のシーケンスの演奏が優先されます。

プライオリティが同値のシーケンスが複数ある場合は、最も古く演奏を開始したシーケンスがプライオリティ判定の対象となります。

次に、シーケンスプレイヤーひとつに対し、二つのシーケンスを演奏しようとした場合の例を示します。

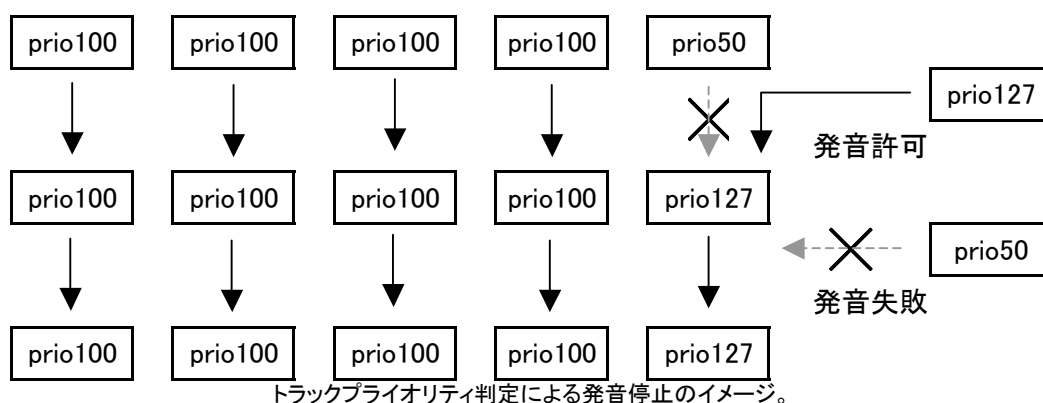


シーケンスプライオリティ判定による演奏停止のイメージ。

6.3.トラックプライオリティによるプライオリティ判定

SmartAudio16は、全シーケンスプレイヤーあわせでの同時発音数が16音に制限されています。それぞれの音には、発音を要求したトラックのプライオリティが適用され、全シーケンスプレイヤーあわせでの同時発音数が16音を超えた場合、トラックプライオリティ同士を比較します。

演奏している中で最もプライオリティの低い音と、これから発音しようとする音のプライオリティを比較して、プライオリティの高い方が演奏されます。プライオリティ判定の候補となる音が複数ある場合は、最も古く演奏を開始した音がプライオリティ判定の対象となります。



7.メモリ管理について

ここでは、SmartAudio16が扱うメモリ領域の概要と、その運用方法を説明します。

7.1.サウンドヒープ

SmartAudio16では、『サウンドヒープ』と呼ばれる連続した動的メモリ領域にデータをロードして、はじめて各サウンドの演奏が可能になります。

サウンドヒープは、SmartAudio16の初期化時にヒープ領域から確保されます。

サウンドヒープのサイズは自由な値を指定することができますが、必要以上に大きい領域を確保してしまうと、アプリケーションの起動中はその大きな領域を占有し続けることになってしまいます。

そこで、SmartAudio16ではサウンドヒープのサイズは極力最小限の大きさにとどめ、アプリケーションの場面毎に必要なサウンドデータを入れ替えて運用する手法を用意しています。

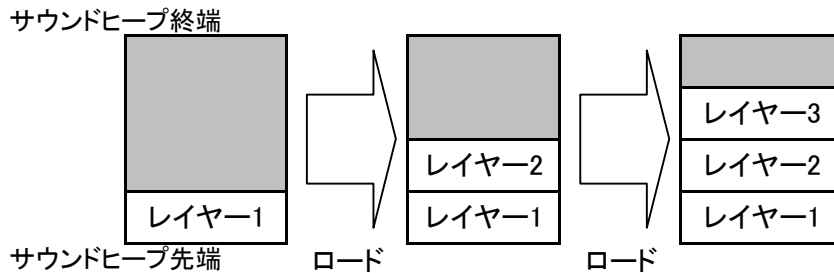
7.2.スタック方式

サウンドヒープは、スタック(積み上げ)方式のシンプルなメモリ領域です。

サウンドヒープへのデータロードは、シーケンスデータやオーディオデータのまとまりであるグループを単位として行います。

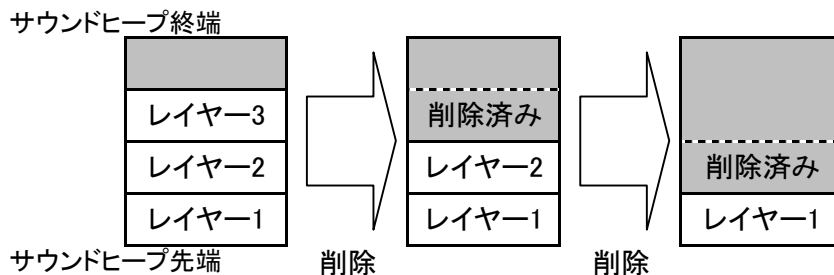
グループが配置されたサウンドヒープ内の位置のことを『レイヤー』と呼びます。

グループのロードを行うたびに、データはアドレスの若い方から順に積み上げられていきます。



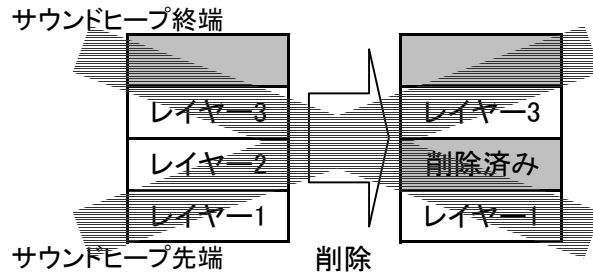
グループロードのイメージ。グループがレイヤーとして積み上げられていきます。

また、グループデータの削除は最上位のレイヤーのみを対象としています。



グループ削除のイメージ。最上部のレイヤーから削除されていきます。

最上位以外のレイヤーにあるグループを直接削除することは出来ません。



このようなグループ削除を行うことはできません。

なお、ロード対象のグループに含まれるシーケンスデータやオーディオデータがすでにサウンドヒープ内に存在している場合は、これらのシーケンスデータやオーディオデータが重複してロードされることはありません。

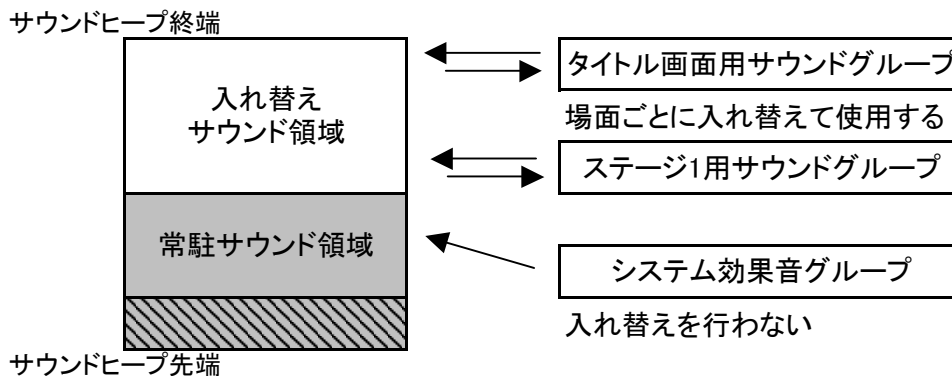
7.3. レイヤー数の制限

レイヤーは、サウンドヒープ内に255個まで作成することが可能です。

255個以上のレイヤーを超えてグループをロードしようとした場合はエラーとなります。

7.4. 実際の運用例

例として、サウンドヒープ内を“常駐”部分と“入れ替え”部分に分けて運用する方法を示します。



システム効果音は最初にロードしておき、場面ごとのグループは必要なときに入れ替えます。

常駐領域には、アプリケーション全体を通して必要となるシステム効果音などをロードします。常駐領域へのロードはアプリケーション起動時に行い、アプリケーション終了までデータを保持し続けます。

入れ替え領域には、ステージのBGMなど、ステージ固有の要素を格納します。

ステージが変わる時、画面遷移などのタイミングでこの内容を入れ替えるようにします。

必要なときに必要なデータだけを読み込むことで、メモリ効率を大幅に高めることができ、バラエティに富んだサウンド表現が可能です。

また、入れ替え頻度の高いグループを上位のレイヤーへ配置することにより、データ入れ替えの手間を軽減させることが可能です。

8.資料

ここでは、各プラットフォームごとのCPU負荷率などの資料を記載します。

| | | iOS | AndroidOS | |
|--------------------|--------------------------|----------------------------------|---|----------------------------------|
| | | iPhone4 (iOS 4.3) | IS03 (AndroidOS 2.1.1) | Xperia acro (AndroidOS 2.3.7) |
| 初期化時間 | | 約30msec | 約180msec | 約400msec |
| データロード時間 | BGM1曲 | 約190msec | 約130msec | 約60msec |
| | BGM+ 約40個 の効果 音 | 約350msec | 約170msec | 約70msec |
| シーケンス演奏中 CPU負荷率 | 平均 | 約8% | 約9% | 約8% |
| | 最大 | 約10% | 約10% | 約9% |
| メモリ使用量 | スタティック | 約26Kbyte | | |
| | ヒープ | 初期化時の指定による | | |
| 必要なファイルと サイズ | ライブラ リファイ ル | libSmartAudio16.a (約251Kbyte) | libSmartAudio16Library.so (約67Kbyte) | |
| | ヘッダ ファイル | SmartAudio16_iOS.h (約9Kbyte) | SmartAudio16.jar(クラスライブラリファイル) (約5Kbyte) | |

改訂履歴

| 日付 | 内容 |
|------------|--|
| 2014.04.12 | 『8.資料』の項を修正。 |
| 2014.02.26 | ライブラリバージョンが1.4.0へ変更されたことに伴い、『5.4.シーケンス演奏時のタイプ』の項を削除。 |
| 2014.02.20 | 誤字・脱字を修正。 |
| 2013.04.10 | バージョン1.2.1に合わせて『8.資料』の項を改訂。 |
| 2013.04.04 | バージョン1.1.2に合わせて『8.資料』の項を改訂。 |
| 2013.02.02 | ドキュメント改訂。 |
| 2012.08.04 | 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。

※ 「ARM」はARM 社の登録商標です。

※ 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。