

ARM AAC Encode Middleware for Linux

RTM0AC0000AEAACMZ1SL32C

ユーザーズマニュアル

RTM0AC0000AEAACMZ1SL32J-01

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したものです。誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、
 家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、
 防災・防犯装置、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1.本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2.本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

このマニュアルの使い方

1. 目的と対象者

このマニュアルは、本製品のインタフェース仕様をユーザに理解していただくためのマニュアルです。本製品を用いた応用システムを設計するユーザを対象にしています。関連するマニュアルと合わせてご使用ください。

本ミドルウェアは、注意事項を十分確認の上、使用してください。注意事項は、各章の本文中、各章の最後、注意事項の章に記載しています。

改訂記録は旧版の記載内容に対して訂正または追加した主な箇所をまとめたものです。改訂内容すべてを記録したものではありません。詳細は、このマニュアルの本文でご確認ください。

2. 本製品のご使用について

本ミドルウェアに含まれる AAC の技術および特許の使用に際しては、お客様自身で別途、下記のライセンス契約を締結して頂く必要があります。

Via Licensing Corporation (<http://www.vialicensing.com/>)

本製品のご使用にあたっては、弊社とソフトウェア使用許諾（ライセンス）契約を取り交わして頂く必要があります。

お客様は、別途契約のある場合を除き、本製品の一部または全部について再配布することはできません。

本ソフトウェアを組み込んだ製品は、外国為替および外国貿易管理法に定められる規制貨物技術に該当する場合があります。輸出する（日本国外への持ち出しを含む）場合は、規制貨物・技術の該当確認ならびに同法に従い日本国政府の輸出許可申請等必要な手続きをお取りください。

3. 関連マニュアル

[1] Linux Interface Specification Yocto recipe Start-Up Guide

[2] R-Car Series, 2nd Generation User's Manual: Hardware

4. 略語／略称の説明

略語／略称	英語名
AAC	Advanced Audio Coding
PCM	Pulse Code Modulation
CRC	Cyclic Redundancy Check
SCE	Single channel element
CPE	Channel Pair Element

すべての商標および登録商標は、それぞれの所有者に帰属します。

目次

1. 概要	1
1.1. 基本仕様	1
2. API 仕様	3
2.1. API 関数とデータ構造一覧	3
2.2. API 関数詳細	4
2.2.1. RSAACE_Init 関数	4
2.2.2. RSAACE_Encode 関数	5
2.2.3. RSAACE_get_version 関数	7
2.3. データ構造詳細	8
2.3.1. RSAACE_AAC 型構造体	8
2.3.2. RSAACE_AACInfo 型構造体	9
3. 入出力データフォーマット	11
3.1. 入力データフォーマット	11
3.2. 出力データフォーマット	11
3.2.1. ADTS	12
3.2.2. ADIF	12
3.2.3. raw data	12
3.3. 入力バッファ(16 ビット PCM)	13
3.4. 出力バッファ(ビットストリーム)	13
4. 戻り値とエラー回復	14
4.1. API 関数の戻り値とステータスコード	14
4.1.1. RSAACE_Init 関数のステータスコード	15
4.1.2. RSAACE_Encode 関数のステータスコード	16
5. 処理フロー	17
6. 組み込み手順	18
6.1. 製品構成	18
6.2. 開発環境	18
6.3. アプリケーションプログラムの作成	19
6.4. 標準ライブラリのインクルード	19
6.5. ライブラリのリンク	19
6.6. コンパイルオプションの設定	19
7. 注意事項	20
7.1. 予約語	20
7.2. リエントラント性について	20
7.3. エンコード処理の監視	20

1. 概要

1.1. 基本仕様

本ミドルウェアは、16 ビット PCM を国際標準規格である ISO/IEC 13818-7: 2006、ISO/IEC13818-7: 2006/Amd.1: 2007 に準拠した AAC 形式に符号化したビットストリームに変換するソフトウェアです。本ミドルウェアの基本仕様を表 1.1 表 1.2 に記載します。

表 1.1 基本仕様 (1)

項目	内容	
製品型名	RTM0AC0000AEAACMZ1SL32C	
製品名	ARM AAC Encode Middleware for Linux	
バージョン情報	0x20010211 ^(*)	
対象プロセッサ	ARM	
仕様	準拠規格	MPEG-2 Advanced Audio Coding + ISO/IEC13818-7:2006 Forth Edition + ISO/IEC13818-7:2006/Amd.1:2007
	プロファイル	ISO/IEC-13818-7:2006 Low Complexity
	入力フォーマット	16 ビット PCM
	出力フォーマット	AAC ビットストリーム(ADIF, ADTS, raw data)
	チャンネル数	1: モノラル
		2: ステレオ, デュアルモノラル
	サンプリング周波数	8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48 [kHz]
	ビットレート ^(*)	8k - 288k [ビット/秒/チャンネル]
	サンプリング周波数[kHz]	対応ビットレート [ビット/秒/チャンネル]
	8	8 k - 48 k
	11.025	8 k - 66.15 k
	12	8 k - 72 k
	16	8 k - 96 k
	22.05	8 k - 132.3 k
	24	8 k - 144 k
	32	8 k - 192 k
	44.1	8 k - 264.6 k
	48	8 k - 288 k

(*) RSAACE_get_version 関数で取得できるバージョン情報

(*) 必要に応じてビットレートを調整してください。

表 1.2 基本仕様 (2)

項目	内容	
処理負荷	26.10MHz ^(*1)	
プログラム 容量	ROM	202626 バイト
	RAM	56584 バイト
	STACK	2000 バイト
	入力バッファ	4,096 バイト以上（ユーザによる割り当て）
	出力バッファ	1,536 バイト以上（ユーザによる割り当て）
利用形態	C言語インタフェース（ライブラリ関数）	
制限事項	+ ビットストリームには CRC（Cyclic Redundancy Code）は存在しません。 + リエントラントに対応しています。	

(*1) 処理負荷はすべてのケースでの決定的な値ではなく、R-Car M2 の Cortex-A15 で 44.1[kHz]、ステレオの一般的な楽曲 CD 音源を 128[k ビット/秒]でエンコードして測定した場合の参考値あり、ピーク値は提示する値の限りではありません。

2. API 仕様

この章では本ミドルウェアが提供する API 関数およびデータ構造を説明します。

2.1. API 関数とデータ構造一覧

本ミドルウェアで提供する API 関数およびデータ構造の一覧を表 2.1 および表 2.2 に示します。

表 2.1 API 関数一覧

No.	関数名	機能概要
1	RSAACE_Init	初期化処理をおこなう
2	RSAACE_Encode	1 フレームのエンコード処理をおこなう
3	RSAACE_get_version	バージョン番号を取得する

表 2.2 データ構造一覧

No.	データ構造名	種別	概要
1	RSAACE_AAC	構造体	エンコード処理で使用するワーク構造体
2	RSAACE_AACInfo	構造体	エンコードパラメータを格納する構造体

2.2. API 関数詳細

2.2.1. RSAACE_Init 関数

本関数は、エンコード処理で使用するワーク領域の初期化をおこないます。本関数が正常に実行された場合、戻り値に `RSAACE_R_GOOD` を返し、ステータスコード `RSAACE_NO_ERROR` をワーク構造体のメンバ `AAC_status` に設定します。アプリケーションプログラムから `RSAACE_AACInfo` 構造体 `info` のメンバに設定したエンコードパラメータの値にエラーがあった場合、戻り値に `RSAACE_E_ERROR` を返し、ステータスコードをメンバ `AAC_status` に設定します。戻り値とステータスコードの詳細は、4.1 節を参照してください。

本関数は、`RSAACE_Encode` 関数の実行に先立って呼び出す必要があります。なお、サンプリング周波数、チャンネルモードが異なる音源を連続してエンコード処理する場合は `RSAACE_Encode` 関数の実行に先立って、その都度呼び出す必要があります。

<関数形式>

```
RSAACE_TYPE_ERROR RSAACE_Init(  
    RSAACE_AAC *work,  
    RSAACE_AACInfo *info  
);
```

<引数>

<code>work</code>	[入力/出力]	エンコード処理で使用するワーク構造体へのポインタ。
<code>info</code>	[入力]	アプリケーションプログラムが、エンコードパラメータの値を設定する構造体へのポインタ。

<戻り値>

(正常)	
<code>RSAACE_R_GOOD</code>	正常に終了しました。
(異常)	
<code>RSAACE_E_ERROR</code>	設定したエンコードパラメータに異常があります。エンコードパラメータの設定値は 表 2.3、ステータスコードの詳細は 4.1 節を参照してください。

2.2.2. RSAACE_Encode 関数

本関数は、引数 `input` が示すアドレスから PCM データを取得して図 2.1 に示すように 2 フレーム前に対応する 1 フレーム分の PCM のエンコード処理を行い、引数 `output` で指定された出力バッファへビットストリームを出力します。ビットストリームのバイト数は、引数 `bcnt` が指すアドレスに出力します。

引数 `pcnt` で指定された PCM のサンプル数が 1 フレーム以上の場合は戻り値には、`RSAACE_R_GOOD` を返し、ステータスコード `RSAACE_R_NO_ERROR` をワーク構造体のメンバ `AAC_status` に設定します。

ただし、図 2.1 に示すように `RSAACE_Init` 関数実行直後の本関数の呼び出し①では、ビットストリームを出力しません（引数 `bcnt` が指すアドレスには 0 を出力します）。このとき、戻り値に `RSAACE_C_CHECK` を返し、ステータスコードには `RSAACE_C_PRE_ENCODE` を設定します。また、本関数の中で呼び出し②では、`RSAACE_AACInfo` 構造体 `info` のメンバ `gap` の値に 0 を設定しているときは呼び出し①と同様となりますが、`RSAACE_AACInfo` 構造体 `info` のメンバ `gap` の値に 1 を設定しているときは PCM の先頭に 1 フレーム分の無音フレームを加えてエンコード処理した生成フレームを出力します。

引数 `pcnt` で指定された PCM のサンプル数が 1 フレーム分に満たない場合は不足しているサンプルの値を 0 として PCM データを補完して、エンコード処理は実行されます。このとき、戻り値には `RSAACE_C_CHECK` を返し、ステータスコードに `RSAACE_C_LACK_FRAME` を設定します。入力 PCM が末端の場合を除いて、入力バッファには 1 フレーム分のサンプル数（3.1 節参照）の PCM を用意してください。

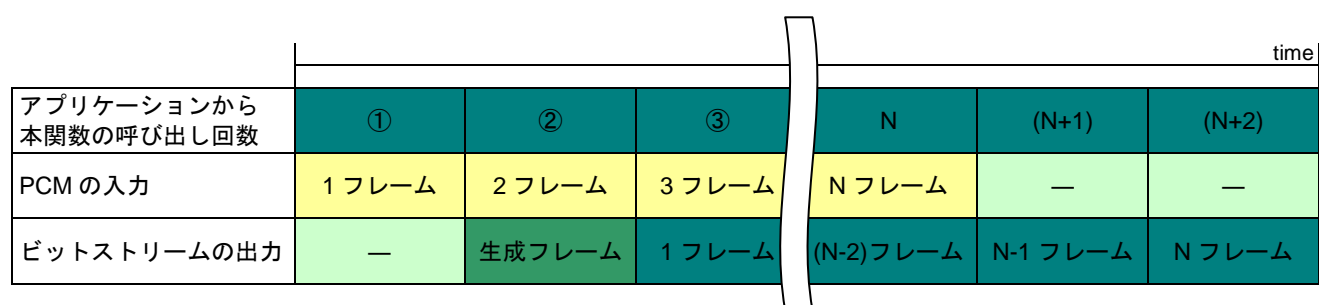
入力 PCM 終了時は、引数 `pcnt` で指定された PCM のサンプル数を 0 として本関数を連続して実行し、戻り値 `RSAACE_C_CHECK` およびステータスコード `RSAACE_C_END_OF_FILE` を確認してください。

入力 PCM 終了時、引数 `pcnt` で指定された PCM のサンプル数が 0 の場合では、PCM 最終フレームの 1 つ前のフレームがエンコード処理され、戻り値には `RSAACE_C_CHECK` を返し、ステータスコードには `RSAACE_C_BEFORE_END_OF_FILE` を設定します。ただし、図 2.1 に示すように `RSAACE_Init` 関数実行直後の本関数の呼び出し①ではエンコード処理は実行されず、戻り値に `RSAACE_C_CHECK` を返し、ステータスコードには `RSAACE_C_END_OF_FILE` を設定します。

入力 PCM 終了後、再度、引数 `pcnt` で指定された PCM のサンプル数を 0 とした場合、PCM 最終フレームがエンコード処理され、戻り値には `RSAACE_C_CHECK` を返し、ステータスコードには `RSAACE_C_END_OF_FILE` を設定します。

本関数が呼び出される前に `RSAACE_Init` 関数が実行されている必要があります。

ステータスコードが `RSAACE_C_LACK_FRAME`、`RSAACE_C_BEFORE_END_OF_FILE` および `RSAACE_C_END_OF_FILE` の場合、入力バッファに PCM データを補充すればエンコード処理を継続できますが、本関数を次回呼び出す前に `RSAACE_Init` 関数を実行することを推奨します。



(注) N は、PCM の総フレーム数。生成フレームとは、RSAACE_AACInfo 構造体 info のメンバ gap の値に 1 を設定して呼び出し②でエンコード処理したときのみ生成するフレームです。

図 2.1 本関数の呼び出し回数と PCM の入力およびビットストリームの出力の関係

<関数形式>

```
RSAACE_TYPE_ERROR RSAACE_Encode (
    RSAACE_AAC *work,
    short *input,
    unsigned int *pcnt,
    unsigned char *output,
    unsigned int *bcnt
);
```

<引数>

work	[入力/出力]	本ミドルウェアが使用するワーク構造体へのポインタ。
input	[入力]	PCM の先頭アドレス。
pcnt	[入力/出力]	[入力] PCM の有効サンプル数へのポインタ。入力形式の詳細は 3 章を参照してください。 [出力] 処理したサンプル数へのポインタ。
output	[出力]	ビットストリームを格納する出力バッファの先頭アドレス。出力形式の詳細は 3 章を参照してください。
bcnt	[出力]	出力したビットストリームのバイト数へのポインタ。

<戻り値>

(正常)

RSAACE_R_GOOD 正常に終了しました。

(異常)

RSAACE_E_ERROR 設定した引数に異常があります。

(確認)

RSAACE_C_CHECK ステータスコードを確認してください。

2.2.3. RSAACE_get_version 関数

本関数は、本ミドルウェアのバージョン番号を返します。

<関数形式>

```
unsigned int RSAACE_get_version ( void );
```

<戻り値>

本ミドルウェアのバージョン番号として、16 進数 0xabbbccde の形式で返します。

a :	バージョン番号
bbb :	リビジョン番号
cc :	ビルド番号
d :	予約
e :	予約

2.3. データ構造詳細

2.3.1. RSAACE_AAC 型構造体

RSAACE_AAC 型構造体は本ミドルウェアが使用するワーク構造体です。構造体のメンバ AAC の内容は、非公開としておりますので、内容の参照、変更はおこなわないでください。
なお、この構造体の領域は、偶数アドレスに必ず配置してください。

<形式>

```
#define WORKSIZE          13986
typedef struct _RSAACE_AAC
{
    int AAC_status;
    int AAC[WORKSIZE];
} RSAACE_AAC;
```

<メンバ>

AAC_status	ステータスコード
AAC[WORKSIZE]	エンコード処理で使用するワーク領域

2.3.2. RSAACE_AACInfo 型構造体

RSAACE_AACInfo 型構造体は、アプリケーションプログラムがエンコードパラメータを本ミドルウェアへ渡すために使用する構造体です。この構造体の各メンバにエンコードパラメータを設定した後、RSAACE_Init 関数を呼び出してください。

表 2.3 に RSAACE_AACInfo 型構造体に設定するエンコードパラメータの値を示します。

<形式>

```
typedef struct _RSAACE_AACInfo
{
    unsigned int    channelMode;
    unsigned int    sampleRate;
    unsigned int    bitRate;
    unsigned int    outputFormat;
    unsigned int    enable_cbr;
    unsigned int    gap;
    unsigned int    mode;
    unsigned int    home;
    unsigned int    original_copy;
    unsigned int    copyright_id_present;
    unsigned char    copyright_id[9];
} RSAACE_AACInfo;
```

<メンバ>

channelMode	PCM のチャンネルモード
sampleRate	PCM のサンプリング周波数[Hz]
bitRate	ビットストリームのビットレート[ビット/秒/チャンネル]
outputFormat	ビットストリームのフォーマット
enable_cbr	CBR 設定
gap	先頭無音データ付加設定（2.2.2 節参照）
mode	速度優先モード設定
home	複製か原本かを示すビット
original_copy	著作権が保護されているか存在しないかを示すビット
copyright_id_present	copyright_id[] が存在するか否かを示すビット
copyright_id[9]	国際規格に基づいた著作権などの識別コードを格納するビットフィールド。 例えば、ISAN(International Standard Audiovisual Number)など

表 2.3 RSAACE_AACInfo 型構造体への設定値

パラメータ	内容	設定値
channelMode	モノラル	0 (初期値)
	ステレオ	1
	デュアルモノラル	2
sampleRate	サンプリング周波数 [Hz]	8000
		11025
		12000
		16000
		22050
		24000
		32000
		44100
		48000
bitRate ^(*)	各 sampleRate のビットレート [ビット/秒/チャンネル]	fs = 8000 8000 - 48000
		fs = 11025 8000 - 66150
		fs = 12000 8000 - 72000
		fs = 16000 8000 - 96000
		fs = 22050 8000 - 132300
		fs = 24000 8000 - 144000
		fs = 32000 8000 - 192000
		fs = 44100 8000 - 264600
		fs = 48000 8000 - 288000
outputFormat	ADTS	0 (初期値)
	ADIF	1
	raw data	2
cbr_enable	可変ビットレート(VBR) ^(*)	0 (初期値)
	一定ビットレート(CBR)	1
gap	先頭無音データ付加なし	0 (初期値)
	先頭無音データ付加あり	1
mode	音質優先 (標準)	0 (初期値)
	速度優先	1
home	ビットストリームは複製である	0 (初期値)
	ビットストリームは原本である	1
original_copy	ビットストリームの著作権は存在しない	0 (初期値)
	ビットストリームの著作権は保護されている	1
copyright_id_present	copyright_id フィールドは存在しない	0 (初期値)
	copyright_id フィールドは存在する	1
copyright_id[9] ^(*)	8-bit copyright_identifier、 64-bit copyright_number	任意のビット列、あるいは 0 例) copyright_identifier: I.S.A.N. (for audiovisual works) copyright_number: 1234567890123456 (ISAN Number)

- (*) ビットレートには、1 チャンネルあたりの値[ビット/秒/チャンネル]を設定してください。表中には、サンプリング周波数毎に下限値、上限値を記載しています。
- (*) 可変ビットレート「cbr_enable = 0」の場合、音源によって出力ストリームの実際のビットレートが、設定したビットレートの値「bitRate」に不足する傾向にあります。
- (*) 識別コードは、“Registration Authority”によって管理されており、取得するためには“Request of a Registered Identifier (RID)”の手続きを必要とします。詳細は、“Registration Authority”にお問い合わせください。

3. 入出力データフォーマット

3.1. 入力データフォーマット

図 3.1 に PCM のデータ構造を示します。1 フレームのエンコードには、1024 サンプル（モノラルは 2048 バイト、ステレオ・デュアルモノラルは 4096 バイト）を使用します。

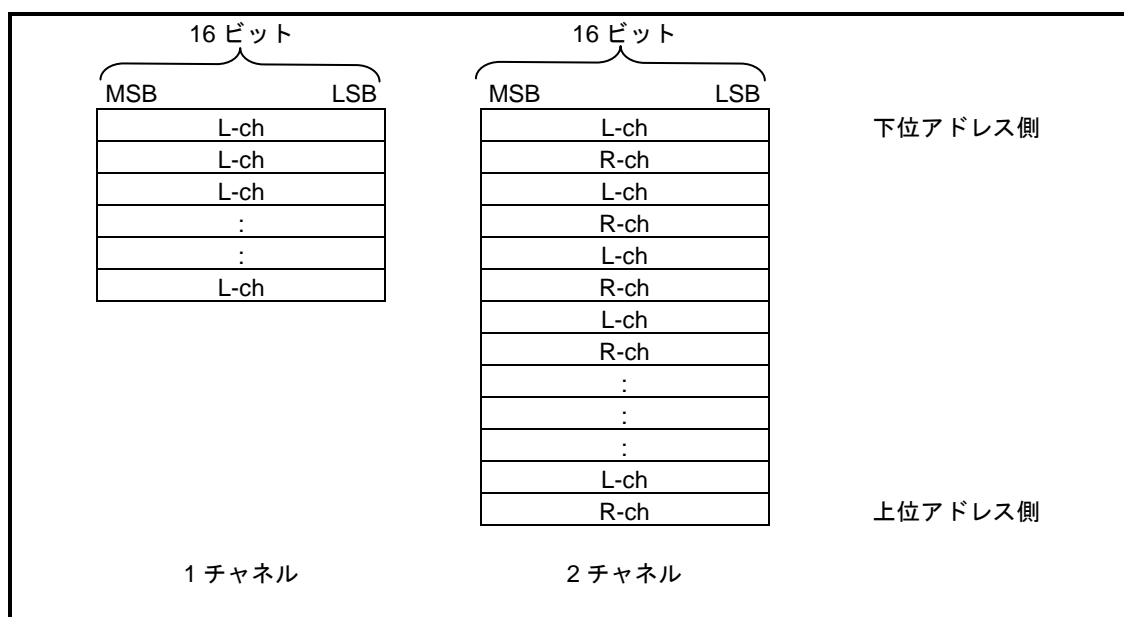


図 3.1 PCM のフォーマット

3.2. 出力データフォーマット

AAC 形式に符号化されたビットストリームは連続するフレームで構成されます。AAC 形式は、ADTS、ADIF または raw data のいずれかです。ビットストリームの構造は、次のようになります。

3.2.1. ADTS

ADTS は、ビットストリームを ADTS フレーム毎に 1 ブロック(raw_data_block)のみで構成します。ADTS フレームの先頭ヘッダには、必ず 12 ビットの同期語が挿入されています。図 3.2 に ADTS のビットストリーム構造を示します。一定ビットレートの場合にのみブロックのセグメントにエレメント<FIL>が入ることがあります。

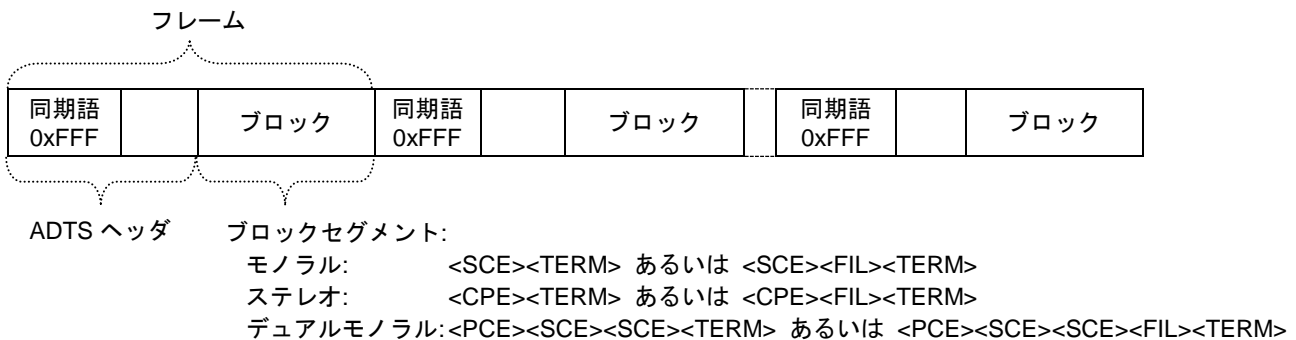


図 3.2 ADTS

3.2.2. ADIF

ADIF のビットストリーム構造は、ビットストリームの先頭のみ ADIF ヘッダが存在し、その後ヘッダは存在せずブロック (raw_data_block) が最後まで続きます。ADIF ヘッダは、1 つのエレメント <PCE(program_config_element)>を含んでいます。図 3.3 に ADIF のビットストリーム構造を示します。一定ビットレートの場合にのみエレメント<FIL>がブロックのセグメントに入ることがあります。

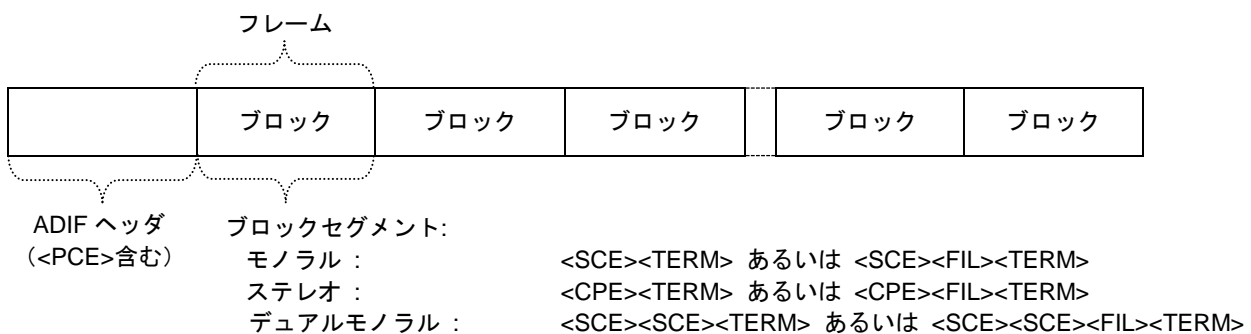


図 3.3 ADIF

3.2.3. raw data

raw data のビットストリームは、ブロック (raw_data_block)のみで構成されます。従って、図 3.3 の ADIF ヘッダが存在しないことを除けば、ADIF と同じ構造になります。この形式ではチャンネル数およびサンプリング周波数の情報をビットストリーム内に含みませんので、アプリケーションプログラムにてそれらを管理してください。

3.3. 入力バッファ(16 ビット PCM)

入力バッファは、short 型配列の連続した領域とします。リングバッファ形式には対応していません。入力バッファサイズは 1024 サンプル以上を確保してください。また、アプリケーションプログラムで本バッファを偶数アドレスに配置してください。奇数アドレスに配置した場合、その動作は保証できません。

RSAACE_Encode 関数は、RSAACE_AACInfo 型構造体に設定したエンコードパラメータの値に応じて、1 フレームのエンコード処理に必要な PCM を入力バッファから取得します。入力バッファ内に 1 フレーム分の PCM がない場合、RSAACE_Encode 関数は不足分を無音データ (0 データ) で補完して 1 フレームのエンコードをおこないます。そのため、連続してエンコードする場合、PCM の終端以外では、入力バッファには RSAACE_Encode 関数実行前に 1 フレーム以上の PCM を用意しておく必要があります。

3.4. 出力バッファ(ビットストリーム)

出力バッファは、unsigned char 型配列の連続した領域とします。リングバッファ形式には対応していません。出力バッファのサイズは 1536 バイト以上としてください。アプリケーションプログラムで本バッファを確保してください。

RSAACE_Encode 関数は、引数で指定されたアドレスの先頭から出力します。

4. 戻り値とエラー回復

4.1. API 関数の戻り値とステータスコード

RSAACE_Init 関数、RSAACE_Encode 関数の戻り値とステータスコードのマクロ定義を以下に示します。ステータスコードは、RSAACE_AAC 型構造体のメンバ AAC_status に設定します。

<定義>

// 戻り値

```
#define RSAACE_R_GOOD ( 0)
#define RSAACE_C_CHECK ( 1)
#define RSAACE_E_ERROR (-1)
```

// ステータスコード

```
#define RSAACE_R_NO_ERROR ( 0)
#define RSAACE_C_END_OF_FILE ( 100)
#define RSAACE_C_LACK_FRAME ( 101)
#define RSAACE_C_PRE_ENCODE ( 102)
#define RSAACE_C_BEFORE_END_OF_FILE ( 103)
#define RSAACE_E_CHANNEL_MODE (-100)
#define RSAACE_E_SAMPLE_RATE (-101)
#define RSAACE_E_BIT_RATE (-102)
#define RSAACE_E_STREAM_FORMAT (-104)
#define RSAACE_E_ENABLE_CBR (-105)
#define RSAACE_E_HOME (-107)
#define RSAACE_E_ORIGINAL_COPY (-108)
#define RSAACE_E_COPY_ID_PRST (-112)
#define RSAACE_E_GAP (-114)
#define RSAACE_E_MODE (-115)
#define RSAACE_E_NULL (-116)
```

4.1.1. RSAACE_Init 関数のステータスコード

表 4.1 に RSAACE_Init 関数のステータスコードとその説明、およびアプリケーションプログラムでの対応方法を記載します。なお異常のステータスコードが設定された際は、表 2.3 を参照し、アプリケーションプログラムで RSAACE_AACInfo 型構造体の各メンバに正しい値を設定してください。

表 4.1 RSAACE_Init 関数のステータスコード

分類	戻り値	ステータスコード	説明	アプリケーションプログラムの対応
正常	RSAACE_R_GOOD	RSAACE_R_NO_ERROR	ワーク領域の初期化に成功しました。	RSAACE_Encode 関数を実行できます。
異常	RSAACE_E_ERROR	RSAACE_E_CHANNEL_MODE	channelMode に設定された値は仕様に違反しています。	RSAACE_Encode 関数を実行できませんので、表 2.3 を確認して、RSAACE_AACInfo 型構造体の各メンバに正しく値を設定後、本関数を再実行してください。
		RSAACE_E_SAMPLE_RATE	sampleRate に設定された値は仕様に違反しています。	
		RSAACE_E_BIT_RATE	bitRate に設定された値は仕様に違反しています。	
		RSAACE_E_STREAM_FORMAT	outputFormat に設定された値は仕様に違反しています。	
		RSAACE_E_ENABLE_CBR	enable_cbr に設定された値は仕様に違反しています。	
		RSAACE_E_HOME	home に設定された値は仕様に違反しています。	
		RSAACE_E_ORIGINAL_COPY	original_copy に設定された値は仕様に違反しています。	
		RSAACE_E_COPYRIGHT_ID_PRESENT	copyright_id_present に設定された値は仕様に違反しています。	
		RSAACE_E_GAP	gap に設定された値は仕様に違反しています。	
		RSAACE_E_MODE	mode に設定された値は仕様に違反しています。	
		RSAACE_E_NULL	引数が NULL ポインタです。	

(注) RSAACE_AAC 型構造体へのポインタが NULL ポインタである場合には、戻り値としては RSAACE_E_ERROR を返しますが、ステータスコードは設定しません。

4.1.2. RSAACE_Encode 関数のステータスコード

表 4.2 に RSAACE_Encode 関数のステータスコードとその説明、およびアプリケーションプログラムでの対応方法を記載します。

表 4.2 RSAACE_Encode 関数のステータスコード

分類	戻り値	ステータスコード	説明	アプリケーションプログラムの対応
正常	RSAACE_R_GOOD	RSAACE_R_NO_ERROR	1 フレーム分のエンコード処理が正常に終了しました。	出力バッファからビットストリームを取得できます。
確認	RSAACE_C_CHECK	RSAACE_C_END_OF_FILE	入力バッファに PCM がありません。引数 pcnt が指す値が 0 に設定され、エンコード処理しました。ビットストリームの出力は、最終フレームです。	出力バッファからビットストリームを取得できます。エンコード処理を継続する場合は、本関数の実行に先立って RSAACE_Init 関数を呼び出してください。
		RSAACE_C_LACK_FRAME	1 フレーム分のエンコード処理が正常に終了しましたが、PCM のサンプル数が、1 フレーム分に満たないため、無音データを補充してエンコード処理を実行しました。PCM の最終フレームの可能性がありま	出力バッファからビットストリームを取得できます。エンコード処理を継続する場合、入力が終わりのときは引数 pcnt が指す値に 0 を設定するか、そうでないときは入力バッファに PCM を補充し、本関数を実行してください
		RSAACE_C_PRE_ENCODE	RSAACE_AACInfo 型構造体メンバ gap に“0”が設定されている場合は、RSAACE_Init 関数直後の PCM 先頭 1 あるいは 2 フレーム目が処理されたことを示し、gap が“1”の場合は、PCM 先頭 1 フレーム目が処理されたことを示します。	出力バッファにはビットストリームを出していません。エンコード処理を継続する場合は、入力バッファに PCM を補充して、本関数を実行してください。
		RSAACE_C_BEFORE_END_OF_FILE	入力バッファに PCM がありません。引数 pcnt が指す値が 0 に設定され、エンコード処理しました。ビットストリームの出力は最終フレームの 1 フレーム前です。エンコード処理を継続した場合、最終フレームの出力を得られます。	出力バッファからビットストリームを取得できます。エンコード処理を継続するには、引数 pcnt が指す値に 0 を設定し、本関数を実行してください。
異常	RSAACE_E_ERROR	RSAACE_E_NULL	引数が NULL ポインタです。	正しく値を設定し、本関数を再実行してください。

(注) RSAACE_AAC 型構造体へのポインタが NULL ポインタである場合には、戻り値としては RSAACE_E_ERROR を返しますが、ステータスコードは設定しません。

5. 処理フロー

API 関数を使用したエンコード処理の一例を図 5.1 に示します。なお、入力データの最終フレームであるかどうかの判定、および入力データのサイズが 1 フレーム以上かどうかの判定、戻り値およびステータスコードの判定は、アプリケーションプログラムでおこなってください。(*1)

- (1) エンコードパラメータを RSAACE_AACInfo 構造体にセットする。
 - (2) RSAACE_Init 関数を呼び出す (初期化)
 - (3) PCM とサンプル数をセットする。
 - (4) RSAACE_Encode 関数を呼び出す (1 フレームのエンコード)。
 - (5) ビットストリームを取得する。
- (*1) (3)~(5)を繰り返し実行してください。(3)においてその PCM の終端では最終の 2 フレームをサンプル数 0 と設定して、(4)でエンコードを繰り返し、(5)の後、ステータスコード RSAACE_C_END_OF_FILE を確認して終了してください。

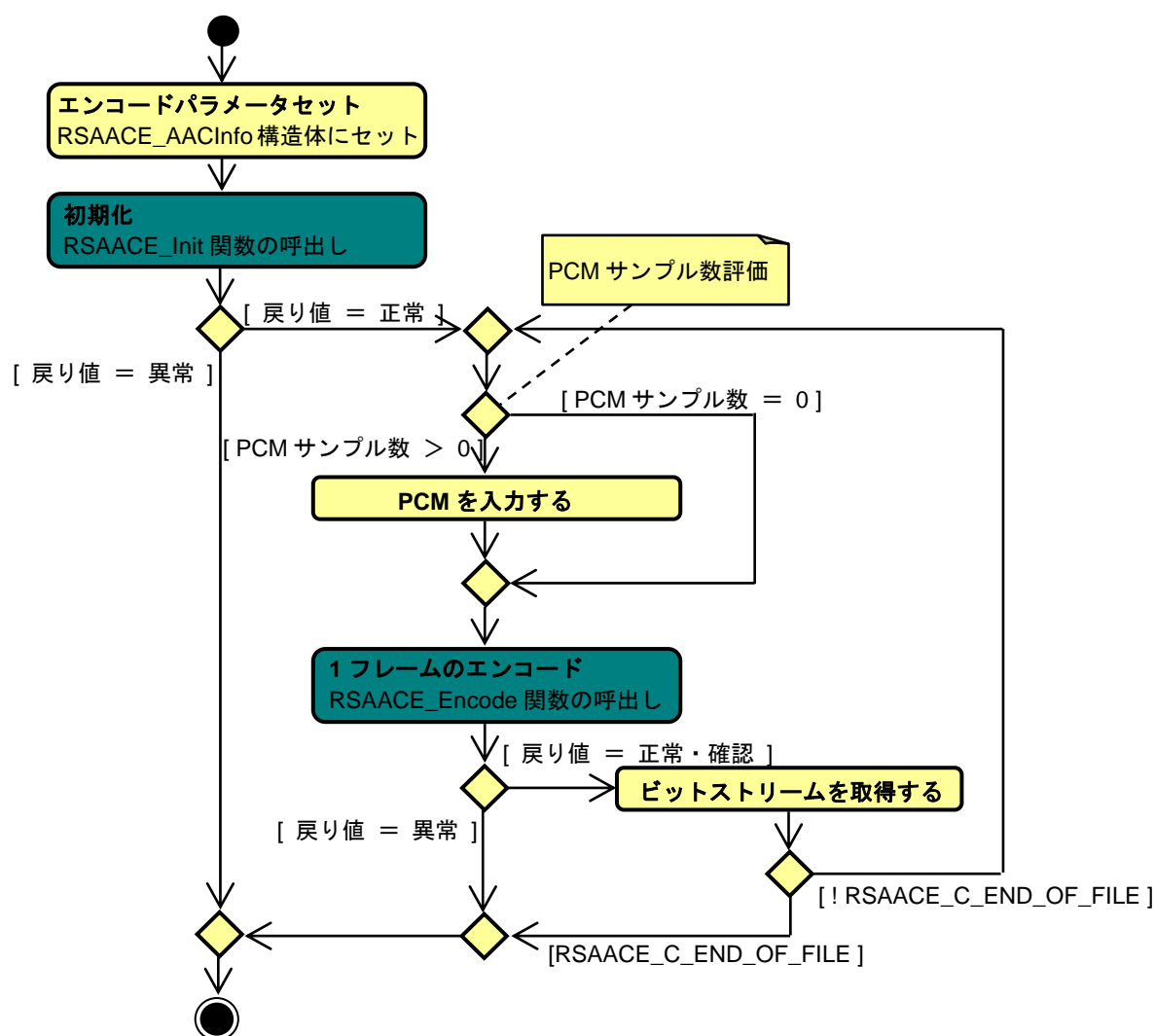


図 5.1 エンコード処理フロー

6. 組み込み手順

6.1. 製品構成

本ミドルウェアは CD-R で提供します。また、CD-R には、表 6.1 のファイルを収録しています。

表 6.1 ファイル一覧

No.	ファイル名	内容
1	RSAACE_AAC.h	API 組み込み用ヘッダファイル
2	libRSAACELA_L.so.2.1	ダイナミックリンクライブラリ
3	RTM0AC0000AEAACMZ1SL32J-01.pdf	ユーザーズマニュアル（日本語）
4	RTM0AC0000AEAACMZ1SL32E-01.pdf	ユーザーズマニュアル（英語）
5	RSAACE_sample.c	サンプルプログラム

6.2. 開発環境

本ミドルウェアをアプリケーションプログラムに組み込む際には、表 6.2 の開発環境、あるいは互換性のある開発環境を使用してください。

表 6.2 開発環境

項目	内容／型名
OS	Linux kernel release 3.10
コンパイラ	cortexa15hf-vfp-neon-poky-linux-gnueabi gcc version 4.8.3 20140401 (prerelease) (Linaro GCC 4.8-2014.04)

6.3. アプリケーションプログラムの作成

本ミドルウェアの API 関数を呼び出すアプリケーションプログラムを作成してください。API 関数を呼び出す際には、本ミドルウェアが提供するヘッダファイルをインクルードしてください。

アプリケーションプログラムでメモリ上に確保する必要のあるバッファおよび構造体を表 6.3 に示します。

表 6.3 アプリケーションプログラムで配置する領域

	名称	型	サイズ[バイト]	備考
1	入力バッファ	short	4,096	ステレオの場合
2	出力バッファ	unsigned char	1,536	
3	ストリーム情報	RSAACE_AACInfo	52	
4	ワークエリア	RSAACE_AAC	55,948	

6.4. 標準ライブラリのインクルード

標準ライブラリの string.h をインクルードしてください。

6.5. ライブラリのリンク

本エンコードミドルウェアライブラリをアプリケーションプログラムとリンクしてください。

6.6. コンパイルオプションの設定

表 6.4 のコンパイルオプションを設定してください。その他はデフォルトの設定に従ってください。

表 6.4 コンパイルオプション

No	コンパイルオプション	設定内容
1	-fsigned-char	char 型を符号付きとする

7. 注意事項

7.1. 予約語

本ミドルウェアでは、他のミドルウェアやアプリケーションプログラムと区別するため、関数名、変数名、およびユーザ公開マクロ名の先頭に“RSAACE_”を付加しています。競合を避けるため、本ミドルウェアを使用するアプリケーションプログラムでは、“RSAACE_”ではじまるシンボルを使用しないでください。

7.2. リエントラント性について

本ミドルウェアは、リエントラントに対応しています。複数のタスクでエンコードを実行する場合は、それぞれに対応するワーク領域等を確保してください。

7.3. エンコード処理の監視

本ミドルウェアの組み込みに際しては、エンコード処理時間をタイマなどで監視し、アプリケーションがハングアップしないようなシステム構成にしてください。

改訂記録	ARM AAC Encode Middleware for Linux ユーザーズマニュアル
------	------------------------------------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
0.01	2014.06.25	-	初版発行
0.02	2014.08.20	1	表 1.1 基本仕様 (1) バージョン情報を追記
		2	表 1.2 基本仕様 (2) 処理負荷とプログラム容量を明記
1.00	2014.08.22	-	このマニュアルの使い方 4. 数や記号の表記を削除
		18	表 6.1 ファイル一覧 ダイナミックリンクライブラリ名を修正

ARM AAC Encode Middleware for Linux
RTM0AC0000AEAACMZ1SL32C
ユーザーズマニュアル

発行年月日 2014 年 06 月 25 日 Rev. 0.01
 2014 年 08 月 22 日 Rev. 1.00

発行 ルネサス エレクトロニクス株式会社



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/contact/>

ARM AAC Encode Middleware for Linux