

ARM 5.1ch aacPlus V2 Decode Middleware for Linux RTM0AC0000ADAAPMZ1SL32C

ユーザーズマニュアル

RTM0AC0000ADAAPMZ1SL32J-02

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、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。

製品ご使用上の注意事項

ここでは、ミドルウェア製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 本製品（ソフトウェア製品及びその関連ソフトウェア製品を含む。以下、同じ。）の使用に際しては、「外国為替及び外国貿易法」等、技術輸出に関する日本及び関連諸国の関係法規の遵守が必要となります。
2. 弊社は、本製品の使用に際しては、弊社もしくは第三者の特許権、著作権、商標権、その他の知的所有権等の権利に関し、別途、個別の契約書等（マニュアルの記載を含む。以下、同じ。）にて弊社による明示的な許諾がある場合を除き、その保証または実施権の許諾を行うものではありません。また本製品を使用したことにより第三者の知的所有権等の権利に関わる問題が生じた場合、弊社はその責を負いませんので予めご了承ください。
3. 本製品およびその仕様、またはマニュアルに記載されている事柄については、将来、事前の予告なしに変更することがありますので、最終的な設計、ご購入、ご使用に際しましては、事前に最新の製品規格または仕様書（マニュアルを含む）をご確認ください。
4. 本製品の使用（マニュアル記載事項に基づくものも含む）により直接または間接に生ずるいかなる損害についても、弊社は一切の責任を負いません。また、本製品の配布に使用される搭載機器や媒体が原因の損害に対しましても、弊社は一切の責任を負いません。
5. 本製品を、宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途向けには使用できません。お客様の用途がこれに該当するかどうか疑問のある場合には、事前に弊社営業担当迄ご相談をお願いいたします。
6. 本製品を使用してお客様のシステム製品を設計される際には、通常予測される故障発生率、故障モードをご考慮の上、本製品の動作が原因での事故、その他の拡大損害を生じないようにフェールセーフ等の十分なシステム上の対策を講じて頂きますようお願いいたします。
7. 本製品およびマニュアルの著作権は弊社が所有しております。お客様は、弊社から提供された本製品を、別途、個別の契約書等にて定める場合を除き、いかなる場合においても全体的または部分的に複写・解析・改変することはできないものとします。
8. お客様は、別途、個別の契約書等にて定める場合を除き、本製品のマニュアルの一部または全部を無断で使用、複製することはできません。
9. 弊社は、本製品を1台のコンピュータで使用する権利をお客様に対してのみ許諾します。よって、本製品を第三者へ譲渡、貸与、賃借することは許諾しないものとします。但し、別途、個別の契約書等にて定められる場合はその条件に従います。
10. 本製品をはじめ弊社半導体およびその関連製品についてのお問い合わせ、ご相談は弊社営業担当迄お願いいたします。

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、本製品のユーザ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。

本マニュアルは、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。



The middleware may not be copied, except to make single copies for archival purpose, and may not be decompiled or reverse engineered, except as permitted by mandatory legal provisions under applicable law.
HE AAC by Dolby Labs Inc. (<http://www.dolby.com>)

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

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

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

参照規格

- ・ ISO/IEC 13818-7:2006
- ・ ISO/IEC 14496-3:2009
- ・ ARIB (社団法人 電波産業会) 標準規格 B-32、B-21 Rev.5.2、Rev.5.3
- ・ ETSI TS 101 154 V1.9.1:2009
- ・ ETSI TS 102 563 V1.2.1:2010-05
- ・ DTG D-Book Issue 6.2.1:May 2010
- ・ Brazilian ISDB-T: ABNT NBR 15602-2:2007 and ABNT NBR 1560

4. 略語／略称の説明

略語／略称	英語名
AAC	Advanced Audio Coding
HE-AAC V1	AAC stream has SBR extension
HE-AAC V2	AAC stream has SBR and PS extension
SBR	Spectral Band Replication
PS	Parametric Stereo
LOAS	Low Overhead Audio Stream
LATM	Low overhead MPEG-4 audio transport multiplex
PCM	Pulse Code Modulation
DRC	Dynamic Range Control
CRC	Cyclic Redundancy Check
SCE	Single Channel Element
CPE	Channel Pair Element
CCE	Coupling Channel Element
LFE	LFE Channel Element
PCE	Program Config Element
DSE	Data Stream Element
I/O	Input/Output

aacPlus は、Dolby Labs Inc.の商標です。

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	概要	1
2	入出力データフォーマット	4
2.1	入力データフォーマット	4
2.1.1	ADTS 形式.....	4
2.1.2	ADIF 形式.....	4
2.1.3	LOAS/LATM 形式.....	5
2.1.4	RawDataStream 形式.....	6
2.1.5	raw_data_block	6
2.2	出力データフォーマット	7
3	API 仕様.....	9
3.1	API 関数一覧.....	9
3.2	API 関数詳細.....	12
3.2.1	RSACPD_Open	12
3.2.2	RSACPD_SetPCEArea	13
3.2.3	RSACPD_GetAdifHeader	16
3.2.4	RSACPD_GetAdtsHeader	17
3.2.5	RSACPD_GetLoasInfo	18
3.2.6	RSACPD_SetFormat	19
3.2.7	RSACPD_Decode	21
3.2.8	RSACPD_Skip	23
3.2.9	RSACPD_GetStatusCode	24
3.2.10	RSACPD_DecodeStatus	25
3.2.11	RSACPD_SetDecOpt	27
3.2.12	RSACPD_get_version	33
3.2.13	RSACPD_SetDSE	34
3.2.14	RSACPD_InterleavePCM	35
3.2.15	RSACPD_MatrixMixdown	37
3.2.16	RSACPD_SetSAC	41
3.2.17	RSACPD_SetDRC	42
4	ユーザ記述関数	43
4.1	ユーザ記述関数	43
4.2	動作概要	44
5	チャネル構成.....	45

5.1	チャンネル数定義	45
6	処理フロー	47
6.1	RawDataStream 形式デコード	47
6.2	ADTS 形式デコード	48
6.3	ADIF 形式デコード	49
6.4	LOAS 形式デコード	50
7	構造体	51
7.1	RSACPD_AAC 型構造体	51
7.2	RSACPD_PCE 型構造体	51
7.3	RSACPD_AdifHeader 型構造体	53
7.4	RSACPD_AdtsHeader 型構造体	53
7.5	RSACPD_LoasInfo 型構造体	54
7.6	RSACPD_OUT_INFO 型構造体	56
7.7	RSACPD_DSE type structure	57
7.8	RSACPD_SAC type structure	58
7.9	RSACPD_DRC type structure	59
8	ステータスコード一覧	60
8.1	ステータスコード一覧	61
8.2	API 関数とステータスコード	63
8.2.1	RSACPD_Open	63
8.2.2	RSACPD_SetPCEArea	63
8.2.3	RSACPD_GetAdifHeader	64
8.2.4	RSACPD_GetAdtsHeader	65
8.2.5	RSACPD_GetLoasInfo	66
8.2.6	RSACPD_SetFormat	67
8.2.7	RSACPD_Decode	67
8.2.8	RSACPD_Skip	71
8.2.9	RSACPD_DecodeStatus	73
8.2.10	RSACPD_SetDecOpt	73
8.2.11	RSACPD_SetDSE	74
8.2.12	RSACPD_MatrixMixdown	74
8.2.13	RSACPD_SetSAC	75
8.2.14	RSACPD_SetDRC	76
9	組込み手順	77
9.1	システム構成	77

9.2	開発環境	77
9.3	ミドルウェア構成	78
9.4	ユーザアプリケーションプログラムの作成	78
9.5	コンパイルオプションの設定	78
10	注意事項.....	79
10.1	予約語	79
10.2	エラー終了後の復帰処理	79
10.3	デコードミドルウェアの監視	79
10.4	aacPlus/AAC 混在符号化ビットストリームのデコード処理.....	79
10.5	SBR ヘッダ情報未取得時の動作.....	79
10.6	デコード途中のチャンネル構成の変化について	80
10.6.1	出力チャンネル数の変化	80
10.6.2	入力チャンネル構成の変化	80
10.7	デコード途中でのサンプリング周波数の変化について	80
	付録.....	81

1 概要

1.1 概要

本ミドルウェアは、国際標準規格である ISO/IEC 13818-7:2006、ISO/IEC 14496-3:2009 に準拠した AAC 及び HE-AAC(aacPlus)V1/V2 符号化ビットストリームをデコード(復号)する ARM 向け組み込み用デコードミドルウェアです。

本ミドルウェアの基本仕様を表 1.1に記載します。なお、本節以降、符号化ビットストリームの種別(プロファイル)については、HE-AAC(aacPlus)を aacPlus として記載します。

表 1.1 基本仕様

項目	内容	
製品名	ARM 5.1ch aacPlus V2 Decode Middleware for Linux	
製品型名	RTM0AC0000ADAAPMZ1SL32C	
対象 CPU	ARM	
OS	Linux kernel release 3.10	
メモリ容量 (注 1)	ROM 領域	約 172K [バイト]
	RAM 領域	約 2K [バイト]
	スタック	5K [バイト]
	ワーク領域	約 225K [バイト] (ユーザによる割り当て)
	入力バッファ	1 [バイト]以上 (ユーザによる割り当て)
	出力バッファ	2048/チャンネル [バイト]以上 (AAC-LC デコード時、ユーザによる割り当て) 4096/チャンネル [バイト]以上 (HE-AAC デコード時、ユーザによる割り当て)
インタフェース	C 言語インタフェース (ライブラリ関数)	

表 1.2 基本仕様(2)

項目		内容	
デコード仕様	準拠規格	ISO/IEC 13818-7:2006 Fourth Edition ISO/IEC 14496-3:2009 Fourth Edition Brazilian ISDB-T: ABNT NBR 15602-2:2007 and ABNT NBR 1560 ETSI TS 101 154 V1.9.1:2009 (DVB-T Standard) ETSI TS 102 563 V1.2.1:2010 (DAB Standard) ARIB Standard B-32、B-21 Rev.5.2、B-21 Rev.5.3 DTG D-Book Issue 6.2.1:May 2010	
	対応プロファイル (注 2)	AAC-LC HE-AAC V1 (aacPlus V1) HE-AAC V2 (aacPlus V2)	
	入力フォーマット	ADTS 形式、ADIF 形式、LOAS/LATM 形式 (注 2)、RawDataStream 形式フォーマット対応	
	出力フォーマット	16 ビットリニア PCM (非インタリーブ形式) (注 3)	
	対応チャンネル数	1ch (モノラル), 2ch (ステレオ、デュアルモノラル) 3ch (3/0, 2/1), 4ch (3/1, 2/2), 5ch (3/2), 5.1ch (3/2+LFE) ※「/」は、前方/後方スピーカのチャンネル数	
	ダウンミックス	3ch (3/0、2/1)、4 ch (3/1、2/2)、5 ch (3/2)、5.1 ch (3/2+LFE)をモノラル、または ステレオにダウンミックス可能	
	対応サンプリング周波数	AAC	8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 64, 88.2, 96 [kHz]
		HE-AAC V1/V2 (aacPlus V1/V2)	16, 22.05, 24, 32, 44.1, 48 [kHz] (注 4)
	対応ビットレート	AAC	8~576 [k ビット/秒]
		HE-AAC V1/V2 (aacPlus V1/V2)	8~128 [k ビット/秒]

表 1.3 基本仕様(3)

項目	内容	
処理性能 (注 5)/(注 6)	AAC	平均 11.9 MHz (サンプリング周波数:48kHz、 ビットレート : 128 k ビット/秒、ステレオ) 平均 26.4 MHz (サンプリング周波数:48kHz、 ビットレート : 384k ビット/秒、5.1ch)
	HE-AAC V1 (aacPlus V1)	平均 21.1 MHz (サンプリング周波数:48kHz、 ビットレート : 48 k ビット/秒、ステレオ) 平均 60.1 MHz (サンプリング周波数:48kHz、 ビットレート : 128 k ビット/秒、5.1ch)
	HE-AAC V2 (aacPlus V2)	平均 12.0 MHz (サンプリング周波数:48kHz、 ビットレート : 48 k ビット/秒、ステレオ)
エンディアン	リトルエンディアン	
リエントラント	対応	

(注 1) メモリ容量記載時、K = 1024 としています。

(注 2) 本ミドルウェアは、LOAS フォーマットの CRC エラーを検出しません。

(注 3) ステレオ PCM データは RSACPD_InterleavePCM()関数により、インタリーブ PCM に変換可能です。

(注 4) HE-AAC(aacPlus)符号化ビットストリームのサンプリング周波数は出力サンプリング周波数を示します。

(注 5) アタック音検出時など特定のフレームでビットレートが急激に高くなるような場合、処理性能が劣化する場合があります。

(注 6) R-Car H2 評価ボード(ARM Cortex A15)での測定値です。
あらゆるケースで本性能値以下であることを保証するものではありません。

2 入出力データフォーマット

2.1 入力データフォーマット

本ミドルウェアは、入力された AAC/aacPlus V1/aacPlus V2 符号化ビットストリームから、ヘッダ情報、復号時のパラメータとなるサイド情報、及び復号対象となる周波数コンポーネントから構成されるメインデータを取り出し、デコード処理を行います。

2.1.1 ADTS 形式

ADTS 形式では、1ADTS フレーム¹は 1 つ以上のブロック(raw_data_block)で構成されています。構成ブロックの数は ADTS ヘッダに含まれる「number_of_raw_data_blocks_in_frame」より取得できます。ADTS フレームの先頭には、必ず 12 ビットの同期語が挿入されており、次に ADTS ヘッダが存在します。

図 2.1に ADTS 形式のビットストリーム構造例を示します。

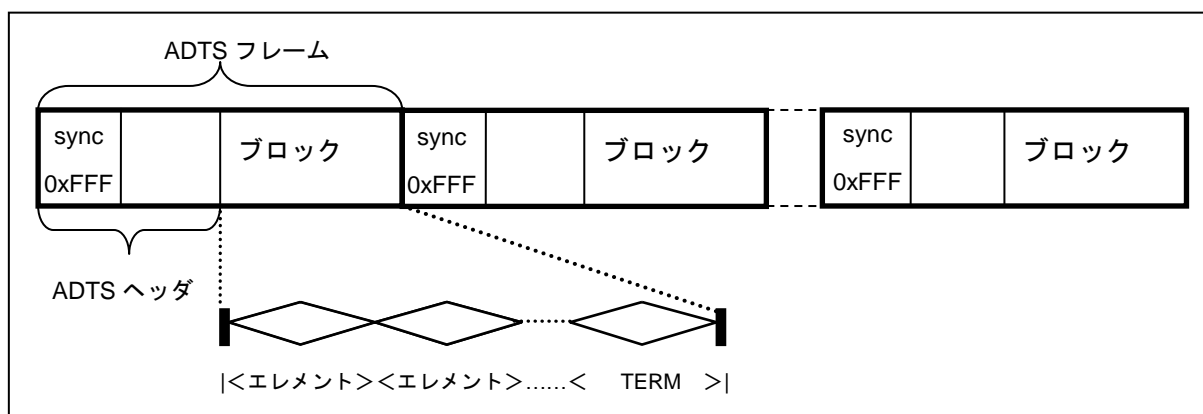


図 2.1 AAC 符号化ビットストリーム(ADTS 形式)

2.1.2 ADIF 形式

ADIF 形式のビットストリーム構造は、ビットストリームの先頭に 1 回のみ 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 ヘッダには、1 つ以上の PCE(program_config_element)を含みます。

¹ 本マニュアルにおいては、“フレーム”と“ADTS フレーム”は区別して使用しています。“1 フレーム”は、1024/2048、960/1920 ワード/チャネルのオーディオデータの出力単位を示します。“ADTS フレーム”は、本節で説明の通り 1 つの ADTS ヘッダと 1 つ以上の raw_data_block で構成されます。

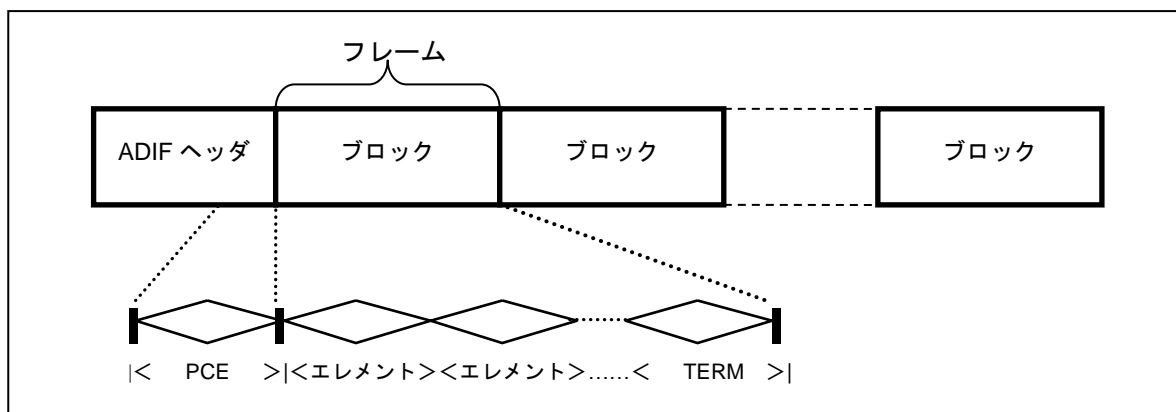


図 2.2 AAC 符号化ビットストリーム(ADIF 形式)の構造

2.1.3 LOAS/LATM 形式

図 2.3に MPEG-4 オーディオトランスポートの構造を示します。MPEG-4 オーディオトランスポートは、以下の 2 つのレイヤで構成されています。

- MPEG-4 オーディオ ペイロードと AudioSpecificConfig エレメントを多重化して扱う Multiplex レイヤ (LATM:Low-overhead MPEG-4 Audio Transport Multiplex)
- 自己同期をとるためのシンタックス(LOAS :Low Overhead Audio Stream) が規定された Synchronization レイヤ

本ミドルウェアは、Multiplex レイヤ(LATM)と Synchronization レイヤ(LOAS) をともに含むストリームのみをサポートしています。

本マニュアルにおいて LOAS 形式と記載した場合、LOAS/LATM とともに含むストリームを意味しています。

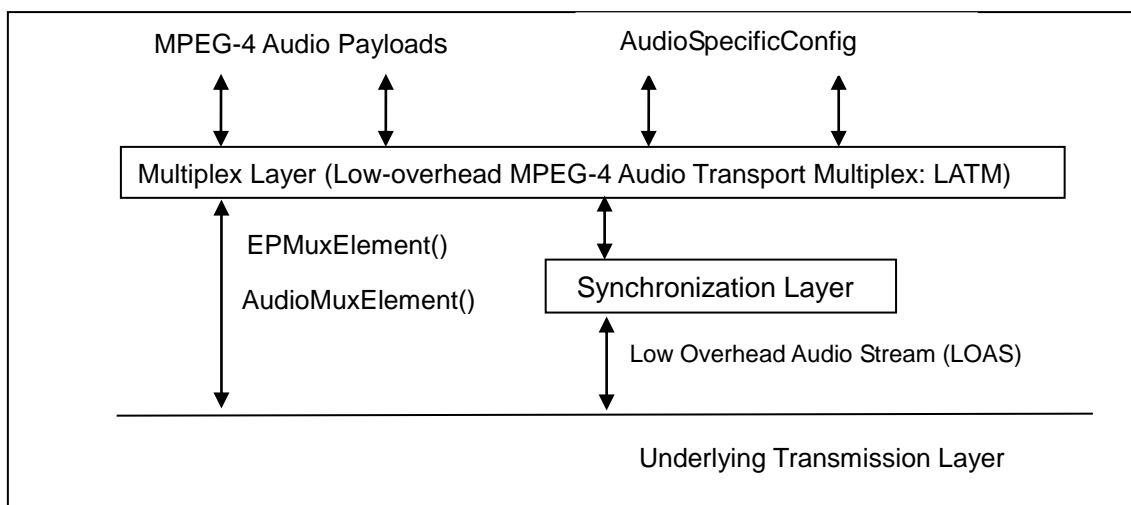


図 2.3 MPEG-4 オーディオトランスポート構造

LOAS 形式は、11 ビットの同期語と 1 つ以上の サブフレーム(raw_data_block) で構成されています。
図 2.4に LOAS 形式の構造を示します。

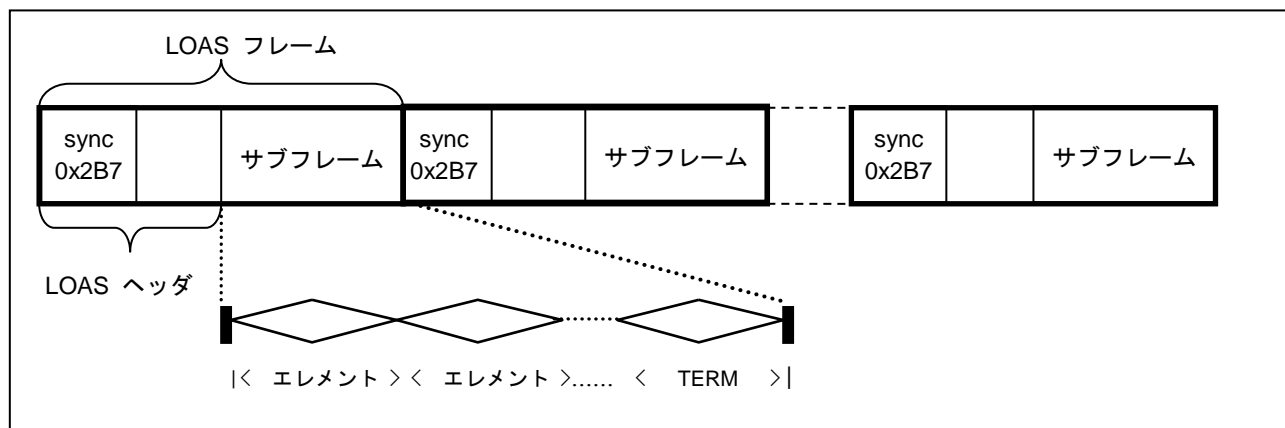


図 2.4 LOAS 形式のビットストリーム構造例

2.1.4 RawDataStream 形式

RawDataStream 形式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 2.2の ADIF ヘッダが存在しないことを除けば、ADIF 形式と同じ構造になります。

2.1.5 raw_data_block

1 つのブロック(raw_data_block)は、7 種類のエレメントで構成されます。エレメントの種類を以下に示します。

<エレメント>

SCE	(single_channel_element)
CPE	(channel_pair_element)
CCE	(coupling_channel_element)
LFE	(lfe_channel_element)
PCE	(program_config_element)
FILL	(fill_element)
DSE	(data_stream_element)
TERM	(ID_END)

2.2 出力データフォーマット

本ミドルウェアは、16 ビット符号付整数型 PCM データを出力します。PCM データは、出力バッファの先頭より順に出力されます。出力 PCM データのエンディアンは CPU のエンディアンと一致します。

本ミドルウェアは、1 ブロック(`raw_data_block`)単位でデコードし PCM データを出力します。PCM データの出力のワード数は、入力ビットストリームの種別(AAC/aacPlus)、デコードモード(AAC アップサンプル/ダウンサンプル SBR)に依存します。詳細については、「3.2.7 RSACPD_Decode」を参照してください。

出力 PCM データは、`RSACPD_OUT_INFO` 型構造体メンバの内の `pcm_cf`, `pcm_lf`, `pcm_rf`, `pcm_ls`, `pcm_rs`, `pcm_lfe` のそれぞれのポインタが指すアドレスから 1024 または、2048 (960 または、1920) ワード出力されます。下図に、出力データフォーマットの例(5.1ch 時)を示します。

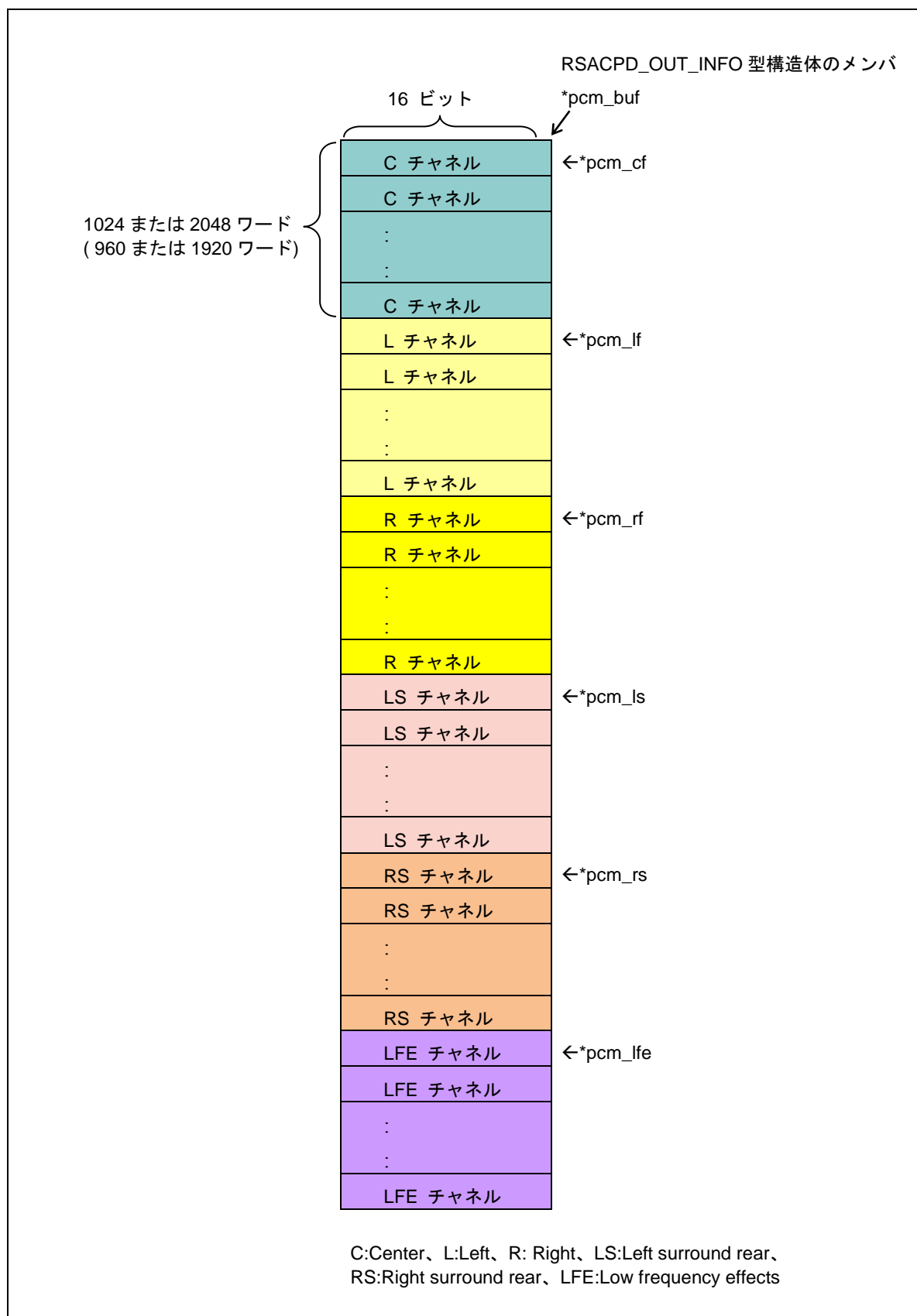


図 2.5 出力 PCM データフォーマット (5.1ch の例)

3 API 仕様

3.1 API 関数一覧

表 3.1に API 関数の定義、機能概要、および 1 つのビットストリームのデコード時に、API 関数の実行が必須、または任意かを示します。

表 3.1 API 関数一覧

No.	関数の定義	機能概要	必須/任意
1	int RSACPD_Open(RSACPD_AAC *aac, unsigned char *buf_adr, int buf_len, unsigned int (*RSACPD_GetData), int key)	本デコードミドルウェアの初期化を行う。	必須
2	int RSACPD_SetPCEArea(RSACPD_AAC *aac, RSACPD_PCE *pce, int pce_cnt)	PCE 情報を取得するための領域を設定する。	PCE のチャンネル構成情報に従う場合、必須
3	int RSACPD_GetAdifHeader(RSACPD_AAC *aac, RSACPD_AdifHeader *header, int *bcnt)	ビットストリームの ADIF 形式ヘッダを取得する。	ADIF 形式の場合、必須
4	int RSACPD_GetAdtsHeader(RSACPD_AAC *aac, RSACPD_AdtsHeader *header, int *bcnt)	ビットストリームの ADTS 形式ヘッダを取得する。	ADTS 形式の場合、必須
5	int RSACPD_GetLoasInfo(RSACPD_AAC *aac, RSACPD_LoasInfo *header, int *bcnt)	ビットストリームの LOAS 形式ヘッダを取得します。	LOAS 形式の場合、必須
6	int RSACPD_SetFormat(RSACPD_AAC *aac, int sampleRateIdx, int decodingType)	RawDataStream 形式のビットストリームをデコードするために必要な情報をセットする。	RawDataStream 形式の場合、必須

7	int RSACPD_Decode(RSACPD_AAC *aac, int *bcnt, RSACPD_OUT_INFO *outInfo, int *pnum)	ビットストリームの1ブロックをデコードする。	必須
8	int RSACPD_Skip(RSACPD_AAC *aac, int *bcnt)	ビットストリームの1ブロックをスキップする。	任意
9	int RSACPD_GetStatusCode(RSACPD_AAC *aac)	API 関数実行後のステータスコードを返す。	任意
10	int RSACPD_DecodeStatus(RSACPD_AAC *aac, int *decodeStatus)	ビットストリームの1ブロックのデコードまたは、スキップ後にデコードステータスを確認する。	任意
11	int RSACPD_SetDecOpt(RSACPD_AAC *aac, int decopt)	デコードオプションの設定を行う。	任意
12	int RSACPD_get_version(void)	本デコードミドルウェアのバージョンを取得する。	任意
13	int RSACPD_SetDSE (RSACPD_AAC *aac, RSACPD_DSE *dse, int dse_cnt)	ビットストリーム中に含まれるDSE(data_stream_element)を取得するための領域の設定を行う。	任意
14	int RSACPD_InterleavePCM(short *pcm_l, short *pcm_r, int pcnt, short *outpcm)	チャンネルごとに出力された非インターリーブPCMデータを2chインターリーブPCMデータに変換する。	任意
15	int RSACPD_MatrixMixdown(RSACPD_AAC *aac, int* sel_std, int* mixdown_mode, RSACPD_OUT_INFO* outInfo, int scale)	マルチチャンネルストリームをステレオまたはモノラルにダウンミックスする。	任意

16	int RSACPD_SetSAC(RSACPD_AAC *aac, RSACPD_SAC *sac)	ビットストリーム中に含まれる MPEG サラウンド情報(スペーシャル 情報)を取得するための領域の設定を 行う。	任意
17	int RSACPD_SetDRC(RSACPD_AAC *aac, RSACPD_DRC *drc, int hi, int lo, int ref_level)	DRC を実行時の設定を行う。	任意

仮引数名は、説明を簡単にするために設定したものであるため自由に設定可能です。

<引数の入出力記号(I/O)について>

本ユーザーズマニュアルでは、各 API 関数の引数の入出力区別(I/O)は、以下の規則で記載されています。

- I : API 関数は、該当引数または該当引数(ポインタ変数の場合)が指定する領域の内容を参照する (read only)
- O : API 関数は、該当引数(ポインタ変数)が指定する領域の内容を設定する
- I/O : API 関数は、該当引数(ポインタ変数)が指定する領域の内容を参照/設定(更新)する

3.2 API 関数詳細

本ミドルウェアの API 関数について、以下に説明します。

3.2.1 RSACPD_Open

構文	int RSACPD_Open(RSACPD_AAC *aac, unsigned char *buf_adr, int buf_len, unsigned int (*RSACPD_GetData), int key)		
機能	本ミドルウェアが使用するワーク領域の初期化処理を行います。		
引数	I/O	意味	
RSACPD_AAC *aac	O	RSACPD_AAC 型構造体へのポインタ	
unsigned char *buf_adr	I	入力バッファ先頭アドレス	
int buf_len	I	入力バッファサイズ (バイト)	
unsigned int (*RSACPD_GetData)	I	ユーザ記述関数(コールバック関数)のポインタ	
int key	I	予約 (0 を設定して下さい)	
戻り値	マクロ名	意味	
0	RSACPD_RTN_GOOD	正常終了	
-1	RSACPD_RTN_ERROR	異常終了	
説明	<p><本関数の実行について></p> <p>本関数は、本ミドルウェアの使用に先立ち、必ず実行する必要があります。これにより、本ミドルウェアのワーク領域である RSACPD_AAC 型構造体が初期化されます。</p> <p><機能詳細></p> <p>第 4 引数には、ユーザ記述関数(コールバック関数)へのポインタを指定して下さい。デコードタスクごとに独立したユーザ記述関数(コールバック関数)を指定できます。</p> <p>第 5 引数(予約)には、必ず 0 を設定してください。</p> <p><注意事項></p> <p>(1) 入力バッファサイズは、1 バイト以上を設定して下さい。それ以外の場合はエラー終了します。</p> <p>(2) 異なるビットストリームをデコードする際には、必ず本関数を実行する必要があります。(例えば、異なる楽曲を再生する場合は本関数を実行する必要があります。)</p> <p>(3) ワーク領域(RSACPD_AAC 型構造体)、および入力バッファはアプリケーションプログラムで領域確保することになります。詳細については「7.1 RSACPD_AAC 型構造体」を参照して下さい。</p> <p>(4) RSACPD_Decode0関数、または RSACPD_Skip0関数がエラー終了した後、再度デコード処理を続ける場合は、必ず本関数を実行して下さい。</p>		

3.2.2 RSACPD_SetPCEArea

構文	<pre>int RSACPD_SetPCEArea(RSACPD_AAC *aac, RSACPD_PCE *pce, int pce_cnt)</pre>	
機能	PCE 情報を取得するための領域を設定する	
引数	I/O	意味
RSACPD_AAC *aac	I/O	RSACPD_AAC 型構造体へのポインタ
RSACPD_PCE *pce	O	RSACPD_PCE 型構造体へのポインタ
int pce_cnt	I	利用可能な PCE 情報の数（設定値：1～16）
戻り値	マクロ名	意味
0	RSACPD_RTN_GOOD	正常終了
-1	RSACPD_RTN_ERROR	異常終了
説明	<p><本関数の実行について> PCE 情報取得領域を設定するためには、RSACPD_Open0関数の実行直後に本関数を実行して下さい。</p> <p><機能詳細> PCE 情報は raw_data_block 内の PCE エlement、または ADIF ヘッダ、LOAS ヘッダ内に存在します。本関数を実行することにより、RSACPD_GetAdifHeader0関数、RSACPD_GetLoasInfo0関数、RSACPD_Decode0関数、RSACPD_Skip0関数実行時に PCE を検出した場合、RSACPD_PCE 型構造体 pce に PCE 情報が出力されます。</p> <p>本ミドルウェアが参照する PCE 情報は、入力データフォーマット、及び RSACPD_SetDecOpt0関数による指定(使用するプログラム(PCE)の選択「3.2.11 RSACPD_SetDecOpt」を参照)によって異なります。</p> <ul style="list-style-type: none"> ・ ADIF 形式 RSACPD_SetDecOpt0関数により使用する PCE が指定された場合は、デコーダは選択された PCE 情報に従います。それ以外は、最初に検出した PCE 情報に従います。 但し、RSACPD_SetDecOpt0関数により指定された PCE が存在しなかった場合は、全ての PCE 情報を無視しデフォルトのチャンネル構成に従います。 ・ LOAS 形式 LOAS ヘッダに PCE が存在する場合は、常に LOAS ヘッダ内の PCE に従います。LOAS ヘッダに PCE が存在しない場合は、最初に検出した PCE 情報、または、RSACPD_SetDecOpt0関数により指定された PCE 情報に従います。 但し、RSACPD_SetDecOpt0関数により指定された PCE が存在しなかった場合は、デフォルトのチャンネル構成に従います。 ・ ADTS 形式、RawDataStream 形式 最初に検出した PCE 情報、または、RSACPD_SetDecOpt0関数により指定された PCE 情報に従います。 但し、RSACPD_SetDecOpt0関数により指定された PCE が存在しなかった場合は、デフォルトのチャンネル構成に従います。 	

<注意事項>

- (1) デフォルトのチャンネル構成情報については、表 5.1参照を参照してください。
- (2) RSACPD_PCE 型構造体は、アプリケーションプログラムで領域確保することになります。詳細については「7.2 RSACPD_PCE 型構造体」を参照して下さい。
 - ・ 第2引数の RSACPD_PCE 型構造体 pce は要素数 pce_cnt(第3引数)の配列として宣言して下さい。
 - ・ ADTS/LOAS/RawDataStream 形式の場合、第3引数 pce_cnt には1を設定してください(1個分の RSACPD_PCE 構造体の領域を確保)。それ以外は無視されます。ADTS/RawDataStream 形式の場合は、常に1つ分の PCE 情報のみを取得します。
 - ・ 第3引数 pce_cnt に16より大きい値、または0以下が設定された場合は、本関数はエラー終了します(ステータスコード:RSACPD_ERR_PCECNT)。
 - ・ 1ブロック(raw_data_block)に複数の PCE エlementが存在する場合、RSACPD_SetDecOpt0関数の使用するプログラムの選択オプションにより、参照する PCE Elementのインスタンスタグ(0~15)を指定することが可能です(「3.2.11 RSACPD_SetDecOpt」を参照)。
 - ・ ADIF 形式の場合に、ADIF ヘッダ内に存在する PCE の個数が第3引数 pce_cnt の値より大きい場合は、RSACPD_GetAdif0関数はエラー終了します(ステータスコード:RSACPD_ERR_PCECNT)。
- (3) 本ミドルウェアは、マトリックスミックスダウン機能のみサポートしています。その他のダウンミックス情報を検出した場合の動作は以下の通りとなります。
 - ・ monomixdown_present = 0 が必須です。それ以外を検出した場合は、RSACPD_Decode0関数はエラー終了します (ステータスコード:RSACPD_ERR_AUDIO_MODE)。
 - ・ stereo_mixdown_present = 0 が必須です。それ以外を検出した場合は、RSACPD_Decode0関数はエラー終了します (ステータスコード:RSACPD_ERR_AUDIO_MODE)。

表 3.2 ビットストリーム形式とデコードが参照する PCE

ビットストリーム形式	PCE格納位置	RSACPD_SetDecOpt()関数による使用する PCE の指定		デコーダが参照する PCE	備考
		指定あり/なし	指定した PCE が存在する/しない		
ADIF	ADIF ヘッダ (複数の PCE が存在)	指定なし	-	最初に検出した PCE	
		指定あり	存在する	選択した PCE	
			存在しない	参照しない	
	ADIF ヘッダ かつ raw_data_block	-	-	-	エラー終了
LOAS	LOAS ヘッダ	-	-	LOAS ヘッダ内の PCE (常に 1 個)	
	raw_data_block	指定なし	-	最初に検出した PCE	
		指定あり	存在する	選択した PCE	
			存在しない	参照しない	
	LOAS ヘッダ かつ raw_data_block	-	-	-	エラー終了
ADTS RawDataStream	raw_data_block	指定なし	-	最初に検出した PCE	
		指定あり	存在する	選択した PCE	
			存在しない	参照しない	

- : 動作に影響を及ぼさない

3.2.3 RSACPD_GetAdifHeader

構文	<pre>int RSACPD_GetAdifHeader(RSACPD_AAC *aac, RSACPD_AdifHeader *header, int *bcnt)</pre>	
機能	ビットストリームの ADIF 形式ヘッダを取得する	
引数	I/O	意味
RSACPD_AAC *aac	I/O	RSACPD_AAC 型構造体へのポインタ
RSACPD_AdifHeader *header	O	RSACPD_AdifHeader 型構造体へのポインタ
int *bcnt	O	ヘッダ情報取得で使用了入力データのバイト数
戻り値	マクロ名	意味
0	RSACPD_RTN_GOOD	正常終了
-1	RSACPD_RTN_ERROR	異常終了
説明	<p><本関数の実行について> ADIF 形式のビットストリームの場合は、必ず本関数を実行して下さい。 処理フローについては、「6.3 ADIF 形式デコード」を参照して下さい。</p> <p><機能詳細> 本関数は、ADIF ヘッダ情報を取得し RSACPD_AdifHeader 型構造体に格納します。</p> <p><注意事項></p> <ol style="list-style-type: none"> (1) RSACPD_AdifHeader 型構造体は、アプリケーションプログラムで領域確保することになります。詳細については「7.3 RSACPD_AdifHeader 型構造体」を参照して下さい。 (2) PCE 情報格納領域を確保するためには、本関数を実行する前に RSACPD_SetPCEArea()関数を実行してください。RSACPD_SetPCEArea()関数が実行されていない場合は、PCE 情報は無視されます。 	

3.2.4 RSACPD_GetAdtsHeader

構文	<pre>int RSACPD_GetAdtsHeader(RSACPD_AAC *aac, RSACPD_AdtsHeader *header, int *bcnt)</pre>	
機能	ビットストリームの ADTS 形式ヘッダを取得する	
引数	I/O	意味
RSACPD_AAC *aac	I/O	RSACPD_AAC 型構造体へのポインタ
RSACPD_AdtsHeader *header	O	RSACPD_AdtsHeader 型構造体のポインタ
int *bcnt	O	ヘッダ情報取得で使用した入力データのバイト数
戻り値	マクロ名	意味
0	RSACPD_RTN_GOOD	正常終了
1	RSACPD_RTN_CHECK	ワーニング
-1	RSACPD_RTN_ERROR	異常終了
説明	<p><本関数の実行について></p> <p>ADTS 形式のビットストリームの場合は、RSACPD_Open0関数実行後、RSACPD_Decode0関数、または RSACPD_Skip0関数実行前に本関数を 1 度実行してください。</p> <p>RSACPD_Decode0関数、または RSACPD_Skip0関数実行後は、RSACPD_DecodeStatus0関数により、1ADTS フレーム内の未デコードのブロック (raw_data_block) の有無を確認し、未デコードのブロック (raw_data_block) がなければ、必ず本関数を実行して下さい。</p> <p>処理フローについては、「6.2 ADTS 形式デコード」を参照して下さい。</p> <p><機能詳細></p> <p>本関数は、ADTS ヘッダ情報を取得し RSACPD_AdtsHeader 型構造体に格納します。</p> <p><注意事項></p> <ol style="list-style-type: none"> (1) RSACPD_AdtsHeader 型構造体は、アプリケーションプログラムで領域確保することになります。詳細については「7.4 RSACPD_AdtsHeader 型構造体」を参照して下さい。 (2) 擬似同期語²を検出した場合は、次の正常な同期語が見つけれずにエラー終了する可能性があります。 (3) 本関数は 4611 バイト以上のストリームを入力しても同期語を検出なかった場合、エラー終了します(ステータスコード: RSACPD_ERR_STREAM_DATA)。 (4) ADTS 形式のフォーマットは、ISO/IEC 14496-3:2001/Cor 2:2002 以降に対応しています。上記以前のフォーマットを入力した場合は、動作を保証しません。 	

²同期語(0xFFF)と同じビット並びのビットストリームのスペクトルデータ部などに含まれる同期語ではないデータ

3.2.5 RSACPD_GetLoasInfo

構文	long RSACPD_GetLoasInfo (RSACPD_AAC *aac, RSACPD_LoasInfo *header, unsigned long *bcnt)		
機能	ビットストリームの LOAS 形式ヘッダを取得する		
引数	I/O	意味	
RSACPD_AAC *aac	I/O	RSACPD_AAC 型構造体へのポインタ	
RSACPD_LoasInfo *header	O	RSACPD_LoasInfo 型構造体へのポインタ	
unsigned long *bcnt	O	ヘッダ情報取得で使用した入力データのバイト数	
戻り値	マクロ名	意味	
0	RSACPD_RTN_GOOD	正常終了	
1	RSACPD_RTN_CHECK	ワーニング	
-1	RSACPD_RTN_ERROR	異常終了	
説明	<p><本関数の実行について></p> <p>LOAS 形式のビットストリームにおいて、デコード、またはスキップを実行する場合は、本関数を RSACPD_Decode0、または RSACPD_Skip0 関数実行前に必ず実行してください。</p> <p>処理フローについては、「6.4 LOAS 形式デコード」を参照してください。</p> <p><機能詳細></p> <p>本関数は、LOAS ヘッダ情報を取得し RSACPD_LoasInfo 型構造体に格納します。</p> <p><注意事項></p> <p>(1) RSACPD_LoasInfo 型構造体は、アプリケーションプログラムで領域確保することになります。詳細については「7.5 RSACPD_LoasInfo 型構造体」を参照して下さい。</p> <p>(2) LOAS 形式のビットストリームは、ヘッダに PCE エlementが含まれる場合と含まれない場合があります。ヘッダに PCE エlementが含まれる場合、ミドルウェアが参照可能な PCE エlementは 1 個までです。</p> <p>LOAS ヘッダ、raw_data_block 内にも PCE エlementが存在する場合は、RSACPD_Decode0関数、RSACPD_Skip0関数はエラー終了します(ステータスコード: RSACPD_ERR_PCE_LOC)。PCE 情報とデコード動作の詳細は、「3.2.2 RSACPD_SetPCEArea」を参照してください。</p> <p>また 2 つ以上の PCE エlementをヘッダ内に検出した場合は、本関数はエラー終了します(ステータスコード: RSACPD_ERR_LOAS_INFO)。</p> <p>(3) 本関数は 4611 バイト以上のストリームを入力しても同期語を検出なかった場合、エラー終了します(ステータスコード: RSACPD_ERR_STREAM_DATA)。</p>		

3.2.6 RSACPD_SetFormat

構文	<pre>int RSACPD_SetFormat(RSACPD_AAC *aac, int sampleRateIdx, int decodingType)</pre>		
機能	RawDataStream 形式のビットストリームをデコードするための情報をセットする		
引数		I/O	意味
RSACPD_AAC *aac		I/O	RSACPD_AAC 型構造体へのポインタ
int sampleRateIdx		I	sampling_frequency_index の設定 (表 3.3を参照して下さい)
int decodingType		I	デコードタイプ (出力 PCM データのワード数指定) 0: 出力 PCM データのワード数=1024 (2048) /チャンネル 1: 出力 PCM データのワード数= 960(1920) /チャンネル
戻り値	マクロ名		意味
0	RSACPD_RTN_GOOD		正常終了
1	RSACPD_RTN_CHECK		ワーニング
-1	RSACPD_RTN_ERROR		異常終了
説明	<p><本関数の実行について> RawDataStream 形式のビットストリームの場合は、必ず本関数を実行して下さい。 処理フローについては、「6.1 RawDataStream 形式デコード」を参照して下さい。</p> <p><機能詳細> 本関数は RawDataStream 形式のビットストリームのデコードまたは、スキップに必要なサンプリング周波数を設定します。 また、本関数は RawDataStream 形式のビットストリームデコード時のデコードタイプ(出力 PCM データのワード数)を、第 3 引数 decodingType の値に従って設定します。 本引数に 0 が設定された場合は、デコードタイプ(出力 PCM データのワード数) : 1024 (aacPlus V1/V2 符号化ストリームデコード時は、2048)/チャンネル となります。 本引数に 1 が設定された場合は、デコードタイプ(出力 PCM データのワード数) : 960 (1920)/チャンネル となります。</p> <p><注意事項> (1) アプリケーションは事前にビットストリームのサンプリング周波数情報を取得している必要があります。 (2) 第 2 引数 sampleRateIdx の値はデコードする RawDataStream 形式のビットストリームの内容と合致した値を設定して下さい。 (3) RSACPD_SetPCEArea0関数が実行されデコード中に PCE エレメントが検出された場合は、PCE 情報のサンプリング周波数値が優先されます。 (4) 第 3 引数 decodingType に、0 または 1 以外の値が設定された場合は、本関数は、ワーニングを出力して終了します(ステータスコード: RSACPD_WARN_INVALID_DECODE_TYPE)。 デコードタイプは、出力 PCM のサンプル数 : 1024(2048)/チャンネルが設定されます。</p>		

表 3.3 サンプリング周波数一覧(Sampling_frequency_index)

Sampling_frequency_index の値	サンプリング周波数	
	AAC 符号化 ビットストリーム	aacPlus 符号化 ビットストリーム
0x0	96000	設定できません
0x1	88200	設定できません
0x2	64000	設定できません
0x3	48000	設定できません
0x4	44100	設定できません
0x5	32000	設定できません
0x6	24000	24000
0x7	22050	22050
0x8	16000	16000
0x9	12000	12000
0xa	11025	11025
0xb	8000	8000
0xc	設定できません	設定できません
0xd	設定できません	設定できません
0xe	設定できません	設定できません
0xf	設定できません	設定できません

(注 1) aacPlus 符号化ビットストリームはデコード時にアップサンプリングを行いますので、

出力 PCM データのサンプリング周波数は上表の 2 倍になります。ただし、ダウンサンプル SBR モード
使用時の出力 PCM データのサンプリング周波数は、上表記載のサンプリング周波数となります。

(注 2) AAC 符号化ビットストリームの AAC アップサンプルモード使用時には、

出力 PCM データのサンプリング周波数は上表の 2 倍になります。

3.2.7 RSACPD_Decode

構文	int RSACPD_Decode(RSACPD_AAC *aac, int *bcent, RSACPD_OUT_INFO *outInfo, int *pnum)																							
機能	ビットストリームの 1 ブロック(raw_data_block)をデコードする																							
引数	I/O	意味																						
RSACPD_AAC *aac	I/O	RSACPD_AAC 型構造体へのポインタ																						
int *bcent	O	デコード処理で使用した入力データのバイト数																						
RSACPD_OUT_INFO *outInfo	I/O	RSACPD_OUT_INFO 型構造体へのポインタ																						
int *pnum	O	1 チャンネルあたりの PCM データの出力ワード数																						
戻り値	マクロ名	意味																						
0	RSACPD_RTN_GOOD	正常終了																						
1	RSACPD_RTN_CHECK	ワーニング																						
-1	RSACPD_RTN_ERROR	異常終了																						
説明	<p><本関数の実行について></p> <p>ビットストリームのデコードを行う場合は、必ず本関数を実行して下さい。 処理フローについては、「6 処理フロー」を参照して下さい。</p> <p><機能詳細></p> <p>本関数は、1 ブロック(raw_data_block)単位でビットストリームをデコードし、PCM データを出力します。</p> <p>本関数を実行する前に、アプリケーションプログラムで必要なサイズ(2048 x デコードチャンネル数)の PCM データ格納領域と RSACPD_OUT_INFO を型構造体の領域を確保し、先頭アドレスを RSACPD_OUT_INFO 型のメンバ*pcm_buf に設定する必要があります。</p> <p>デコード終了後、チャンネル構成に合わせて RSACPD_OUT_INFO 型構造体の各メンバに、チャンネルモード、チャンネル数、PCM データの出力先の先頭アドレスが設定されます。</p> <p>出力 PCM データの構造については、「2.2 出力データフォーマット」を参照して下さい。 RSACPD_OUT_INFO 型構造体への出力内容は、表 7.6 を参照して下さい。</p> <p>PCM データの出力ワード数を下表に示します。PCM データの出力ワード数は、ビットストリームの種類、RSACPD_SetDecOpt0関数で設定するデコードモードによって変化します。異常終了した場合は、0 または不定値となります。デコードモードについては、「3.2.11 RSACPD_SetDecOpt」を参照して下さい。</p> <table><tr><th>ビットストリーム 種別</th><th>デコードモード</th><th>*pnum の値</th><th>出力サンプリング周波数</th></tr><tr><td rowspan="2">AAC</td><td>通常</td><td>1024 / 960</td><td>入力サンプリング周波数</td></tr><tr><td>AAC アップサンプル</td><td>2048 / 1920</td><td>入力サンプリング周波数×2</td></tr><tr><td rowspan="3">aacPlus V1/V2</td><td>通常</td><td>2048 / 1920</td><td>入力サンプリング周波数×2</td></tr><tr><td>ダウンサンプル SBR</td><td>1024 / 960</td><td>入力サンプリング周波数</td></tr><tr><td>強制 AAC</td><td>1024 / 960</td><td>入力サンプリング周波数</td></tr></table>			ビットストリーム 種別	デコードモード	*pnum の値	出力サンプリング周波数	AAC	通常	1024 / 960	入力サンプリング周波数	AAC アップサンプル	2048 / 1920	入力サンプリング周波数×2	aacPlus V1/V2	通常	2048 / 1920	入力サンプリング周波数×2	ダウンサンプル SBR	1024 / 960	入力サンプリング周波数	強制 AAC	1024 / 960	入力サンプリング周波数
ビットストリーム 種別	デコードモード	*pnum の値	出力サンプリング周波数																					
AAC	通常	1024 / 960	入力サンプリング周波数																					
	AAC アップサンプル	2048 / 1920	入力サンプリング周波数×2																					
aacPlus V1/V2	通常	2048 / 1920	入力サンプリング周波数×2																					
	ダウンサンプル SBR	1024 / 960	入力サンプリング周波数																					
	強制 AAC	1024 / 960	入力サンプリング周波数																					

通常モードで aacPlus V1/V2 デコード時に、AAC 部のデコードが正常終了後 SBR 部のデコードで SBR-CRC エラーなどのエラーを検出した場合は、エラーコンシール後にアップサンプルされた PCM データが自動的に出力されます。

<注意事項>

- (1) raw_data_block 内の PCE エlementは、他のElementより前に出現しなければいけません。他のElementの後に PCE Elementが検出された場合、本関数は、エラー終了します(ステータスコード：RSACPD_ERR_PCE_LOC)。
- (2) ADTS 形式の場合、RSACPD_AdtsHeader 型構造体のメンバ frame_length の値が、実際にデコードした ADTS フレーム長と一致しない場合はワーニングを返します。(ステータスコード：RSACPD_RTN_ERR_ADTS_LEN) この場合、PCM データは出力されますが、出力結果は保証しません。
- (3) ADTS 形式のフォーマットは、ISO/IEC 14496-3:2005 以降に対応しています。前記以前のフォーマットを入力した場合は、動作を保証しません。
- (4) ADIF 形式の場合、raw_data_block に PCE Elementを検出した場合には、本関数はエラー終了します(ステータスコード：RSACPD_ERR_PCE_LOC)。
また LOAS 形式の場合、LOAS ヘッダ、raw_data_block 内にともに PCE Elementが存在する場合、本関数はエラー終了します(RSACPD_ERR_PCE_LOC)。
- (5) RSACPD_SetPCEArea0関数が実行されていない場合は、PCE 情報は無視されます。

3.2.8 RSACPD_Skip

構文	int RSACPD_Skip(RSACPD_AAC *aac, int *bcnt)	
機能	ビットストリームの 1 ブロック(raw_data_block)をスキップする	
引数	I/O	意味
RSACPD_AAC *aac	I/O	RSACPD_AAC 型構造体へのポインタ
int *bcnt	O	スキップ処理で使用した入力データのバイト数
戻り値	マクロ名	意味
0	RSACPD_RTN_GOOD	正常終了
1	RSACPD_RTN_CHECK	ワーニング
-1	RSACPD_RTN_ERROR	異常終了
説明	<p><本関数の実行について> ビットストリームのスキップを行う場合は、必ず本関数を実行して下さい。</p> <p><機能詳細> 本関数は、ビットストリームの解析処理のみを行い PCM データの生成・出力は行いません。 1 ブロック(raw_data_block)単位で早送りを実行したい場合に使用します。</p> <p><注意事項></p> <ol style="list-style-type: none"> (1) raw_data_block 内の PCE エlementは、他のElementより前に出現しなければいけません。他のElementの後に PCE Elementが検出された場合、本関数は、エラー終了します(ステータスコード：RSACPD_ERR_PCE_LOC)。 (2) ADTS 形式の場合、RSACPD_AdtsHeader 型構造体のメンバ frame_length の値が、実際にデコードした ADTS フレーム長と一致しない場合、ワーニングを返します。(ステータスコード：RSACPD_RTN_ERR_ADTS_LEN) (3) ADTS 形式のフォーマットは、ISO/IEC 14496-3:2005 以降に対応しています。上記以前のフォーマットを入力した場合は、動作を保証しません。 (4) ADIF 形式の場合、raw_data_block に PCE Elementを検出した場合には、本関数はエラー終了します(ステータスコード：RSACPD_ERR_PCE_LOC)。 また LOAS 形式の場合、LOAS ヘッダ、raw_data_block 内にともに PCE Elementが存在する場合、本関数はエラー終了します(RSACPD_ERR_PCE_LOC)。 (5) RSACPD_SetPCEArea0関数が実行されていない場合は、PCE 情報は無視されます。 	

3.2.9 RSACPD_GetStatusCode

構文	int RSACPD_GetStatusCode(RSACPD_AAC *aac)		
機能	API 関数実行後のステータスコードを返します。		
引数	I/O	意味	
RSACPD_AAC *aac	I	RSACPD_AAC 型構造体へのポインタ	
戻り値			意味
「8 ステータスコード一覧」を参照		ステータスコード	
説明	<p><本関数の実行について></p> <p>各 API 関数の戻り値が異常終了(RSACPD_RTN_ERROR)、またはワーニング終了(RSACPD_RTN_CHECK)の場合に詳細なステータス情報を取得したい場合は、本関数を実行して下さい。</p> <p><機能詳細></p> <p>本関数は、API 関数実行後のステータスコードを確認するための関数です。</p>		

3.2.10 RSACPD_DecodeStatus

構文	int RSACPD_DecodeStatus(RSACPD_AAC *aac, int *decodeStatus)		
機能	ビットストリームの 1 ブロック(raw_data_block)のデコードまたは、スキップ後にデコードステータスを確認する。		
引数	I/O	意味	
RSACPD_AAC *aac	I	RSACPD_AAC 型構造体へのポインタ	
int *decodeStatus	O	デコードステータス設定フラグ 各ステータスをビットフィールドで設定します。ビットフィールドへの設定値は、表 3.4を参照して下さい。	
戻り値	マクロ名	意味	
0	RSACPD_RTN_GOOD	正常終了	
-1	RSACPD_RTN_ERROR	異常終了	
説明	<p><本関数の実行について> 入力ビットストリーム種別を取得したい場合は、本関数を実行して下さい。 処理フローについては、「6 処理フロー」を参照して下さい。</p> <p><機能詳細> 本関数は、RSACPD_Decode0関数、または RSACPD_Skip0関数実行後にデコードステータスの確認を行います。本関数で確認可能なデコードステータスを以下に示します。</p> <p>(1) raw_data_block デコードの継続確認ビット raw_data_block デコード(スキップ)終了後、ADTS フレームに未デコードの raw_data_block が存在する場合、第 2 引数のビットフィールド 4 に 1 が設定されます。 ADTS 形式以外のビットストリームデコード時には、常に 0 が設定されます。</p> <p>(2) PCE 検出フラグ PCE エlementが raw_data_block または、ヘッダに存在した場合、ビットフィールド 5 に 1 が設定されます。</p> <p>(3) SBR 検出フラグ 入力ビットストリームに SBR データを検出した場合に、ビットフィールド 13 に 1 が設定されます。本フラグが 1 の場合、入力ビットストリームは aacPlus 符号化ビットストリーム、0 の場合は AAC 符号化ビットストリームとなります。</p> <p>(4) パラメトリックステレオ(PS)検出フラグ 入力ビットストリームに PS データを検出した場合に、ビットフィールド 14 に 1 が設定されます。本フラグが 1 の場合、入力ビットストリームは aacPlus V2 符号化ビットストリーム、0 の場合は SBR 検出フラグの値に従います。</p> <p><注意事項> (1) RSACPD_AdtsHeader 構造体の number_of_raw_data_blocks_in_frame メンバを参照することにより、1ADTS フレームに含まれる raw_data_block 数を取得することが出来るため、本関数を実行しなくても、アプリケーションプログラム側で RSACPD_GetAdtsHeader0関数の実行を制御すればデコード可能です。</p>		

	<p>(2) ADTS 形式ビットストリームデコード時、raw_data_block デコードの継続確認ビットが 0 の時は、ADTS フレーム内に未処理の raw_data_block が存在しないため RSACPD_Decode()関数、RSACPD_Skip()関数を実行しないでください。実行した場合は、エラー終了(ステータスコード: RSACPD_ERR_NO_RAW_DATA_BLOCK)し、本ビットの値は不定です。</p>
--	---

表 3.4 デコードステータスのビットフィールド

ビット フィールド	フラグ	値
14	パラメトリックステレオ(PS)検出 フラグ	0: PS データなし 1: PS データあり
13	SBR 検出フラグ	0: SBR データなし 1: SBR データあり
5	PCE 検出フラグ	0: PCE エLEMENTあり 1: PCE エLEMENTなし
4	raw_data_block デコードの継続確認 ビット (ADTS 形式時)	0: ADTS フレーム内に未処理の raw_data_block なし。 または、非 ADTS 形式のビットストリーム。 1: ADTS フレーム内に未処理の raw_data_block あり。
その他	予約	0

31	30	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約(※)								予約(※)								予約(※)				

(※) 予約ビットフィールドには、0 が設定されます。

3.2.11 RSACPD_SetDecOpt

構文	int RSACPD_SetDecOpt(RSACPD_AAC *aac, int decopt)	
機能	デコードオプションの設定を行う	
引数	I/O	意味
RSACPD_AAC *aac	I/O	RSACPD_AAC 型構造体へのポインタ
int decopt	I	デコードオプション設定フラグ 各オプション機能の設定をビットフィールドで行います。ビットフィールドへの設定値は、表 3.5を参照して下さい。
戻り値	マクロ名	意味
0	RSACPD_RTN_GOOD	正常終了
1	RSACPD_RTN_CHECK	ワーニング
-1	RSACPD_RTN_ERROR	異常終了
説明	<p><本関数の実行について> デコードオプションの設定を行う場合は、RSACPD_Open0 関数実行後、はじめて RSACPD_Decode0関数を実行する前に、必ず本関数を実行して下さい。 デコードごとのデコードオプションの切り替えは、ユーザ責任となります。RSACPD_Decode0 関数実行後に本関数が実行された場合は、本関数はワーニングを返しますがデコードオプションは有効となります。</p> <p><機能詳細> 本関数は、デコードオプションの設定を行います。詳細は、表 3.5を参照してください。 設定可能なデコードオプションを以下に示します。</p> <p>(1) 強制 AAC デコードモード 強制 AAC デコードモードを有効にすると、aacPlus 用の復号データは無視され aacPlus デコード処理を実施せず、AAC デコード処理のみを実施します。</p> <p>(2) ダウンサンプル SBR モード 入力ビットストリームが aacPlus の場合、ダウンサンプル SBR モード設定フラグを有効にすると、aacPlus デコード処理時にアップサンプリングしないフィルタ処理を実施します。 (注) 強制 AAC デコードモードが有効になっていると、ダウンサンプル SBR モード設定は無効となります。</p> <p>(3) AAC アップサンプルモード 入力ビットストリームが AAC、または aacPlus かつ強制 AAC デコードモードが有効の場合、AAC アップサンプルモード設定フラグを有効にすると、AAC デコード処理時にアップサンプリングするフィルタ処理を実施します。本オプションは、入力ビットストリームのサンプリング周波数が 24kHz 以下の時に有効です。</p> <p>(4) 使用するプログラムの選択有効化 本設定を有効にすると、PCE 情報が複数存在した際に使用するプログラムを選択することが可能になります。プログラムの設定方法は、(5) 使用するプログラムの選択 を参照してください。</p>	

(5) 使用するプログラムの選択

(4) 使用するプログラムの選択有効化フラグ を有効にした場合、プログラム (PCE 情報) のインスタンスタグ(0~15)をデコードオプション設定フラグのビットフィールド[11:8] に設定することにより、使用するプログラムを指定することができます。
詳細は、「3.2.2 RSACPD_SetPCEArea」を参照してください。

(6) SBR 処理モード設定入力ビットストリームが aacPlus の場合、SBR 処理モード設定フラグによって、SBR 処理モードを選択することができます。

SBR 処理モード設定フラグを 01(13bit:0, 12bit:1)に設定すると、常に HQ-SBR(High Quality SBR:高品質帯域拡張方式)モードでデコードを行います。SBR 処理モード設定フラグを 00(13bit:0, 12bit:0)に設定すると、データ種別により SBR デコード処理を自動で切り替えます。

(注) その他の設定値(13bit:1)は予約値です。 設定した場合の動作は保証外です。

(7) エラーコンシールモード設定

エラーコンシールモード設定フラグを有効にすると、入力ビットストリームに異常を検出してもエラー終了せずに、エラーコンシール処理を実施し RSACPD_Decode0関数は PCM データを出力します。

エラーコンシール処理により出力される PCM データは直前フレームのオーバーラップバッファの内容となります。

エラーコンシールモード設定時に、入力ビットストリームに異常を検出した場合 RSACPD_Decode0関数または RSACPD_Skip0関数の戻り値は以下の通りです。

入力ビットストリームの データフォーマット	戻り値
ADTS 形式	RSACPD_RTN_CHECK
ADIF 形式	RSACPD_RTN_ERROR
LOAS 形式	RSACPD_RTN_CHECK
RawDataStream 形式	RSACPD_RTN_ERROR

(注) 入力データの終了を検出した場合の戻り値は、常に RSACPD_RTN_ERROR となります。

エラーコンシールモード設定時に入力データの終了を検出した時の RSACPD_Decode0関数、RSACPD_Skip0関数の動作は以下の通りとなります。

- ・ビットストリームの先頭で入力データの終了を検出した場合 (空のフレーム)、エラーコンシール処理は行われず、RSACPD_Decode0関数は PCM データを出力しません。
- ・ビットストリームの途中で入力データの終了を検出した場合、エラーコンシール処理が実施されます。その結果、RSACPD_Decode0関数は PCM データを出力します。

エラーコンシール設定時は、エラーを検出したフレームで PCM データを出力しますが、ADIF 形式/RawDataStream 形式の場合、以降のフレームのデコード続行はできません。

入力ビットストリームのデータフォーマットが ADTS 形式の場合は

RSACPD_GetAdtsHeader0関数を、LOAS 形式の場合は RSACPD_GetLoasInfo0関数を正常終了するまで繰り返し実行することで、デコード処理の続行が可能となります。

ただし、数フレームの欠落や、偽フレームを検出する可能性があります。

(注 1)先頭フレームおよび、スキップ直後のフレームのデコードでエラーがあった場合は、PCM データの出力は行いません。

(注 2)PS ビットストリームでエラーコンシール処理を行った場合の出力データはモノラルとなります。ステレオデータを取得するためには、L チャンネルの出力 PCM を R チャンネルにコピーする、RSACPD_InterleavePCM0関数を実行するなどの対応をとって下さい。

<注意事項>

本 API 関数は、RSACPD_Decode0関数のデフォルト動作を変更したい場合に実行する必要があります。本関数を実行しない場合の RSACPD_Decode0関数のデフォルト動作は以下の通りとなります。

デコードオプション	デフォルト設定
(1) 強制 AAC デコードモード	無効
(2) ダウンサンプル SBR モード	無効
(3) AAC アップサンプルモード	無効
(4) 使用するプログラムの選択有効化	無効
(5) 使用するプログラムの選択	無効
(6) SBR 処理モード設定	自動切り替え
(7) エラーコンシールモード設定	無効

表 3.5 デコードオプションのビットフィールド

ビット フィールド	デコードオプション	値(意味)
31-17	予約	(必ず 0 を設定して下さい)
16	エラーコンシールモード 設定フラグ	0 : エラーコンシール処理を実施しません 1 : エラーコンシール処理を実施します
15-14	予約	(必ず 0 を設定して下さい)
13-12	SBR 処理モード設定フラグ	00 : データ種別により SBR デコード処理を自動で切り替えます モノラル、パラメトリックステレオの場合は、HQ-SBR (高品質帯域拡張方式)、 その他の場合は LP-SBR (低演算量帯域拡張方式)で SBR デコード処理します 01 : 常に HQ-SBR (高品質帯域拡張方式)で SBR デコード処理します 10 : 予約 11 : 予約
11-8	使用するプログラムの選択	使用する PCE のインスタンスタグの値
7	AAC アップサンプルモード設定フラグ	0 : AAC アップサンプルモードを無効にします 1 : AAC アップサンプルモードを有効にします (入力サンプリング周波数 24kHz まで)
6-5	予約	(必ず 0 を設定して下さい)
4	ダウンサンプル SBR モード 設定フラグ	0 : ダウンサンプル SBR モードを無効にします 1 : ダウンサンプル SBR モードを有効にします
3-2	予約	(必ず 0 を設定して下さい)
1	使用する PCE の選択有効化フラグ	0 : 使用する PCE の選択を無効にします 1 : 使用する PCE の選択を有効にします
0	強制 AAC デコード モード設定フラグ	0 : 強制 AAC デコードモードを無効にします 1 : 強制 AAC デコードモードを有効にします

31	30	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約(※)							予約(※)								予約(※)			予約(※)				

(※)予約ビットフィールドには、必ず 0 を設定して下さい。それ以外は動作保証外です。

表 3.6 デコードオプションとデコード動作

入力ビット ストリーム種別	設定							
	AAC アップ サンプル モード	○	-	-	○	○	○	-
	ダウン サンプル SBR モード	-	○	-	○	-	○	○
AAC	強制 AAC デコード モード	-	-	○	-	○	○	○
	fs ≤ 24kHz	AAC アップ サンプル モード	---	---	AAC アップ サンプル モード	AAC アップ サンプル モード	AAC アップ サンプル モード	---
	32kHz ≤ fs ≤ 48kHz	---	---	---	---	---	---	---
aacPlus V1/V2	64kHz ≤ fs ≤ 96kHz	---	---	---	---	---	---	---
	fs ≤ 24kHz	---	ダウン サンプル SBR モード	強制 AAC デコード モード	ダウン サンプル SBR モード	強制 AAC デコード モード かつ AAC アップ サンプル モード	強制 AAC デコード モード かつ AAC アップ サンプル モード	強制 AAC デコード モード (注 1)
	32kHz ≤ fs ≤ 48kHz	ダウン サンプル SBR モード (注 2)	ダウン サンプル SBR モード (注 2)	強制 AAC デコード モード	ダウン サンプル SBR モード (注 2)	強制 AAC デコード モード	強制 AAC デコード モード (注 1)	強制 AAC デコード モード (注 1)
	64kHz ≤ fs ≤ 96kHz	強制 AAC デコード モード (注 3)	強制 AAC デコード モード (注 3)	強制 AAC デコード モード	強制 AAC デコード モード (注 3)	強制 AAC デコード モード	強制 AAC デコード モード	強制 AAC デコード モード

※ サンプリング周波数(fs)は、入力ビットストリームのサンプリング周波数を示します
(AAC 部のサンプリング周波数)。

(注 1) 強制 AAC モードとダウンサンプル SBR モード設定が同時に指定された場合、ダウンサ ンプルモードは
無効です。

(注 2) SBR 部のサンプリング周波数が 48kHz より高い場合、ダウンサンプル SBR モードの設定にかかわらず、
ダウンサンプル SBR モードでデコードが実施されます。ステータスコードにはワーニングが設定されます。

(注 3) SBR 部のサンプリング周波数が 96kHz より高い場合、強制 AAC デコードモードの設定にかかわらず、
強制 AAC デコードモードでデコードが実施されます。ステータスコードにはワーニングが設定されます。

○ : オプションを有効に設定


- : オプションを無効に設定

--- : 設定オプションがデコード動作に影響を与えない

表 3.7 デコードモードと出力サンプリング周波数

入力ビット ストリーム種別	設定	AAC アップ サンプル モード	-	○	-	-	○	○	○	-
	ダウン サンプル SBR モード	-	-	○	-	○	-	○	○	
	強制 AAC デコード モード	-	-	-	○	-	○	○	○	
AAC	24kHz	24kHz	48kHz	24kHz	24kHz	48kHz	48kHz	48kHz	24kHz	
	48kHz	48kHz	48kHz	48kHz	48kHz	48kHz	48kHz	48kHz	48kHz	
	96kHz	96kHz	96kHz	96kHz	96kHz	96kHz	96kHz	96kHz	96kHz	
aacPlus V1/V2	24kHz (注 1)	48kHz	48kHz	24kHz	24kHz	24kHz	48kHz	48kHz	24kHz	
	48kHz (注 2)	48kHz (注 3)	48kHz (注 3)	48kHz (注 3)	48kHz	48kHz (注 3)	48kHz	48kHz	48kHz	
	96kHz	96kHz (注 4)	96kHz (注 4)	96kHz (注 4)	96kHz	96kHz (注 4)	96kHz	96kHz	96kHz	

※ サンプリング周波数は、入力ビットストリームのサンプリング周波数を示します
(AAC 部のサンプリング周波数)。

 : 同じ入力サンプリング周波数の AAC と aacPlus の出力サンプリング周波数が不一致。

太字 : デコードオプションが出力サンプリング周波数に影響。

下線 : デコードオプションにかかわらず、自動的に出力サンプリング周波数に影響。

(注 1) SBR 部のサンプリング周波数:48kHz

(注 2) SBR 部のサンプリング周波数:96kHz

(注 3) SBR 部のサンプリング周波数が 48kHz より高い場合、ダウンサンプル SBR モードの設定にかかわらず、
ダウンサンプル SBR モードでデコードが実施されます。ステータスコードにワーニングが設定されます。

(注 4) SBR 部のサンプリング周波数が 96kHz より高い場合、強制 AAC デコードモードの設定にかかわらず、
強制 AAC デコードモードでデコードが実施されます。ステータスコードにはワーニングが設定されます。

○ : オプションを有効に設定

- : オプションを無効に設定

3.2.12 RSACPD_get_version

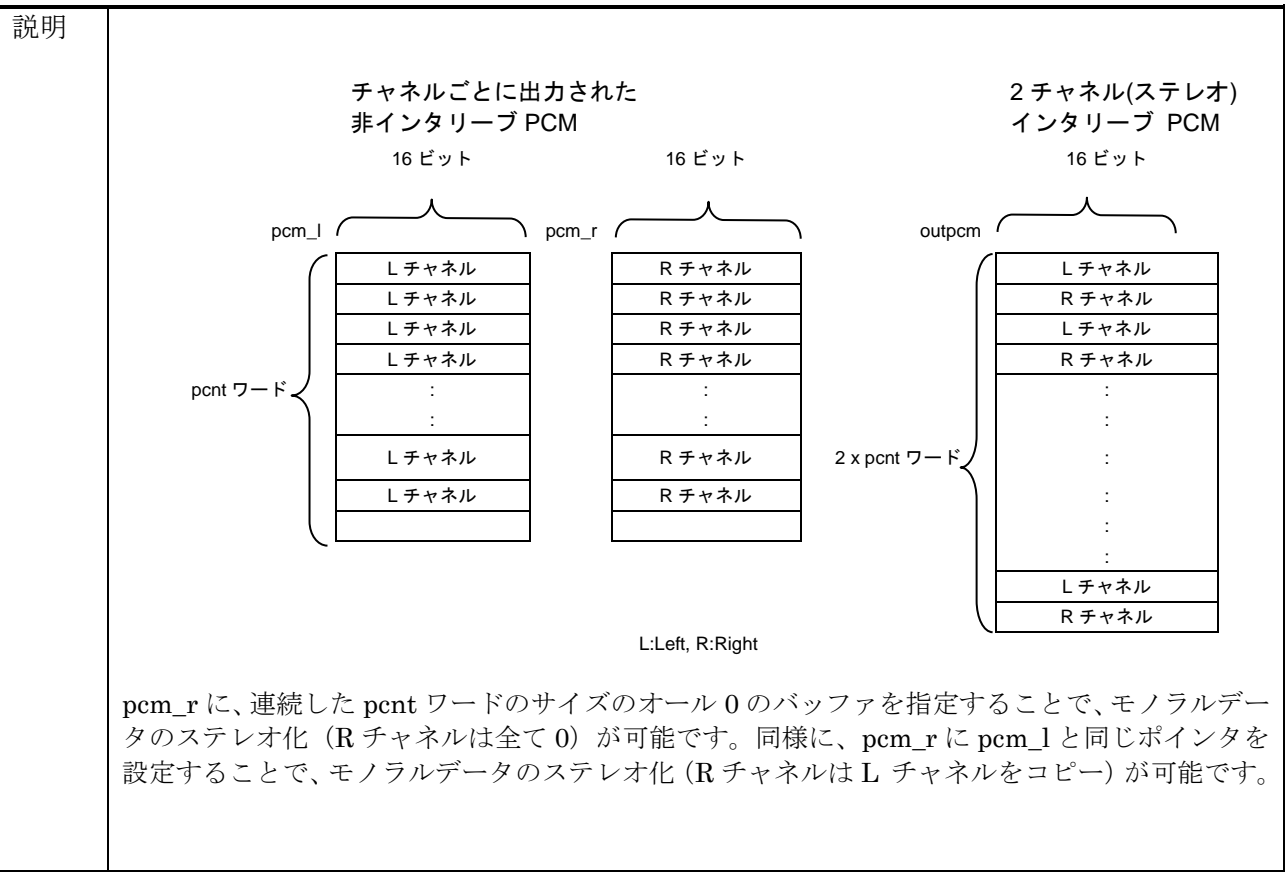
構文	int RSACPD_get_version(void)	
機能	本ミドルウェアのバージョン番号を返します。	
引数	I/O	意味
なし	—	—
戻り値	意味	
0x10010112	バージョン番号を 0xabbbccdd の形式で返します。 a : バージョン番号 bbb : リビジョン番号 cc : ビルド番号 dd : 予約	
説明	<p><本関数の実行について> 本ミドルウェアのバージョン番号を取得したい場合は、本関数を実行して下さい。</p> <p><機能詳細> 本関数は、戻り値として本ミドルウェアのバージョン番号を返します。</p>	

3.2.13 RSACPD_SetDSE

構文	int RSACPD_SetDSE(RSACPD_AAC *aac, RSACPD_DSE *dse, int dse_cnt)		
機能	ビットストリーム中に含まれる DSE(data_stream_element)を取得するための領域の設定を行う。		
引数	I/O	意味	
RSACPD_AAC *aac	I/O	RSACPD_AAC 型構造体へのポインタ	
RSACPD_DSE *dse	O	RSACPD_DSE 型構造体へのポインタ	
int dse_cnt	I	取得可能な DSE の数 (設定値：1～16)	
戻り値	マクロ名	意味	
0	RSACPD_RTN_GOOD	正常終了	
-1	RSACPD_RTN_ERROR	異常終了	
説明	<p><本関数の実行について></p> <p>ビットストリーム中に含まれる DES エlement に DVB-T 規格準拠したダウンミックス情報が格納されている場合、RSACPD_Open0関数または RSACPD_SetDecOpt0関数実行後、最初に RSACPD_Decode0関数または RSACPD_Skip0関数を実行する前に本関数を実行することで、DSE(data_stream_element)情報を取得することができます。</p> <p><機能詳細></p> <p>RSACPD_Decode0 関数、RSACPD_Skip0 関数 実行時に DSE 情報を検出した場合、RSACPD_DSE 型構造体 dse に DSE 情報が出力されます。</p> <p><注意事項></p> <p>(1) RSACPD_DSE 型構造体は、アプリケーションプログラムで領域確保することになります。詳細については「7.7 RSACPD_DSE type structure」を参照して下さい。</p> <ul style="list-style-type: none">・第 2 引数の RSACPD_DSE 型構造体 dse は要素数 dse_cnt(第 3 引数)の配列として宣言して下さい。・DSE 情報は、1 ブロック中に現れた順に、RSACPD_DSE 型構造体に格納されます。・dse_cnt で設定された値より多く DSE が現れた場合は、dse_cnt で設定された値分は、RSACPD_DSE 型構造体に格納されますが、以降の DSE は読み飛ばされます。・第 3 引数の dse_cnt に、1 より小さい値、または 16 より大きい値が設定された場合は、ミドルウェアはエラー終了します(ステータスコード:RSACPD_ERR_DSE_CNT) <p>(2) 本関数を実行しない場合、DSE(data_stream_element)を取得することはできませんが、デコード処理は続行可能です。</p> <p>(3) DSE(data_stream_element)を取得したブロック(raw_data_block)の次のブロックに DSE がなかった場合は、RSACPD_DSE 型構造体メンバの present は 0 に設定されますが、その他のメンバはクリアされません。</p> <p>(4) 本関数を実行しない場合、または DSE エlement に含まれる情報が DVB-T 規格準拠したダウンミックス情報ではない場合は、本ミドルウェアは、DSE 情報を無視します。</p> <p>(5) 複数の DSE エlement を 1 raw_data_block に検出した場合は、最初に検出した DSE 情報を参照します。</p>		

3.2.14 RSACPD_InterleavePCM

構文	int RSACPD_InterleavePCM(int *pcm_l, int *pcm_r, int pcnt, short *outpcm)		
機能	チャンネルごとに出力された非インタリーブ PCM データを 2 チャンネル(ステレオ)インタリーブ PCM データに変換します。		
引数	I/O	意味	
Int *pcm_l	I	L チャンネルの PCM データ格納バッファへのポインタ	
Int *pcm_r	I	R チャンネルの PCM データ格納バッファへのポインタ	
Int pcnt	I	1 チャンネルあたりの PCM データの出力ワード数 (設定値 : 1024、2048、960、または 1920)	
short *outpcm	O	2 チャンネル(ステレオ)にインタリーブした PCM データを出力する バッファへのポインタ	
戻り値	マクロ名	意味	
0	RSACPD_RTN_GOOD	正常終了	
-1	RSACPD_RTN_ERROR	異常終了	
説明	<p><本関数の実行について> 2 チャンネル(ステレオ)インタリーブ PCM データ変換機能を使用する場合は、RSACPD_Decode0 関数正常終了後、本関数を実行して下さい。</p> <p><機能詳細> 本関数は、RSACPD_Decode0関数によってチャンネルごとに出力された非インタリーブ PCM データを 2 チャンネル(ステレオ)インタリーブ PCM データに変換します。</p> <p>PCM データのワード数 pcnt には、RSACPD_Decode0関数の pcnt と同じ値を設定して下さい。 outpcm には連続した 2 x pcnt ワードサイズのバッファへのポインタを設定して下さい。</p>		



3.2.15 RSACPD_MatrixMixdown

構文	<pre>int RSACPD_MatrixMixdown(RSACPD_AAC *aac, int *sel_std, int *mixdown_mode, RSACPD_OUT_INFO *outInfo, int scale)</pre>	
機能	マルチチャンネルストリームを 2 チャンネルステレオまたはモノラルにダウンミックスする。	
引数	I/O	意味
RSACPD_AAC *aac	I	RSACPD_AAC 型構造体へのポインタ
int *sel_std	I	準拠規格 0: ISO/IEC13818-7、ISO/IEC14496-3 準拠 1: ARIB STD-B21 5.2 版準拠 2: ARIB STD-B21 5.3 版準拠(オーバーフロー対策を実施しない) 3: ARIB STD-B21 5.3 版準拠(オーバーフロー対策を実施する)
	O	実施したダウンミックスの準拠規格 0: ISO/IEC13818-7、ISO/IEC14496-3 準拠 1: ARIB STD-B21 5.2 版準拠 2: ARIB STD-B21 5.3 版準拠 (オーバーフロー対策を実施せず) 3: ARIB STD-B21 5.3 版準拠 (オーバーフロー対策を実施した)
int *mixdown_mode	I	ダウンミックスのモード設定 0: ステレオにダウンミックス (外部擬似サラウンドプロセッサ用ダウンミックスを実施しない) 1: ステレオにダウンミックス (外部擬似サラウンドプロセッサ用ダウンミックスを実施する) 2: モノラルにダウンミックス 3: ステレオにダウンミックス DVB-T 規格準拠、 正規化係数を使用 4: ステレオにダウンミックス DVB-T 規格準拠、 正規化係数を使用しない (オーバーフロー対策を実施しない) 5: ステレオにダウンミックス DVB-T 規格準拠 正規化係数を使用しない (オーバーフロー対策を実施する)
	O	実施したダウンミックスのモード 0: ステレオにダウンミックス (外部擬似サラウンドプロセッサ用ダウンミックスを実施せず) 1: ステレオにダウンミックス (外部擬似サラウンドプロセッサ用ダウンミックスを実施した) 2: モノラルにダウンミックス、正規化係数使用 3: ステレオにダウンミックス DVB-T 規格準拠、 正規化係数を使用した 4: ステレオにダウンミックス DVB-T 規格準拠、 正規化係数を使用せず (オーバーフロー対策を実施せず) 5: ステレオにダウンミックス DVB-T 規格準拠、 正規化係数を使用せず (オーバーフロー対策を実施した)
RSACPD_OUT_INFO *outInfo	I/O	RSACPD_OUT_INFO 型構造体へのポインタ (デコード終了後、出力チャンネル数、音声モード、各チャンネルの PCM データの出力バッファの先頭アドレスが設定されます。)

int scale	I	ダウンミックスによるオーバーフロー回避のためのスケール値 (設定値: 0~2)																										
戻り値	マクロ名	意味																										
0	RSACPD_RTN_GOOD	正常終了																										
1	RSACPD_RTN_CHK	ワーニング																										
-1	RSACPD_RTN_ERROR	異常終了																										
説明	<p><本関数の実行について></p> <p>ダウンミックス機能を使用する場合は、RSACPD_Decode()関数が正常終了した場合に、ポストプロセスとして本関数を実行して下さい。</p> <p><機能詳細></p> <p>本関数は、チャンネルモードが3チャンネル(3/0, 2/1)、4チャンネル(3/1, 2/2)、5チャンネル(3/2)、または、5.1チャンネル(3/2 + LFE)のマルチチャンネルストリームを2チャンネルステレオまたはモノラルにダウンミックスします。その他のチャンネルモードの場合に、本関数が呼び出された場合は異常終了します。</p> <p>ステレオへのダウンミックス処理が正常終了した場合は、RSACPD_OUT_INFO 型構造体の pcm_lf, pcm_rf に、モノラルの場合は pcm_cf が指すアドレスからダウンミックスされた PCM データが出力されます。</p> <p>次表に各規格で定義されたダウンミックス計算式を示します。</p> <table border="1"> <thead> <tr> <th></th><th>計算式</th><th>準拠規格</th></tr> </thead> <tbody> <tr> <td>(1)</td><td> $L' = 1 / (1 + 1/\sqrt{2} + A) \times (L + 1/\sqrt{2} \times C + A \times LS)$ $R' = 1 / (1 + 1/\sqrt{2} + A) \times (R + 1/\sqrt{2} \times C + A \times RS)$ </td><td rowspan="3">ISO/IEC13818-7、 ISO/IEC14496-3</td></tr> <tr> <td>(2)</td><td> $L' = 1 / (1 + 1/\sqrt{2} + 2 \times A) \times [L + 1/\sqrt{2} \times C - A \times (LS + RS)]$ $R' = 1 / (1 + 1/\sqrt{2} + 2 \times A) \times [R + 1/\sqrt{2} \times C + A \times (LS + RS)]$ </td></tr> <tr> <td>(3)</td><td> $M = 1 / (3 + 2 \times A) \times [L + C + R + A \times (LS + RS)]$ </td></tr> <tr> <td>(4)</td><td> $L' = 1/\sqrt{2} \times (L + 1/\sqrt{2} \times C + A \times LS)$ $R' = 1/\sqrt{2} \times (R + 1/\sqrt{2} \times C + A \times RS)$ </td><td rowspan="2">ARIB STD-B21 5.2 版</td></tr> <tr> <td>(5)</td><td> $L' = 1/\sqrt{2} \times [L + 1/\sqrt{2} \times C - A \times (LS + RS)]$ $R' = 1/\sqrt{2} \times [R + 1/\sqrt{2} \times C + A \times (LS + RS)]$ </td></tr> <tr> <td>(6)</td><td> $L' = L + 1/\sqrt{2} \times C + A \times LS$ $R' = R + 1/\sqrt{2} \times C + A \times RS$ </td><td rowspan="2">ARIB STD-B21 5.3 版</td></tr> <tr> <td>(7)</td><td> $L' = L + 1/\sqrt{2} \times C - A \times (LS + RS)$ $R' = R + 1/\sqrt{2} \times C + A \times (LS + RS)$ </td></tr> <tr> <td>(8)</td><td> $L' = 1 / (1 + B + A) \times (L + B \times C + A \times LS)$ $R' = 1 / (1 + B + A) \times (R + B \times C + A \times RS)$ </td><td>DVB-T (正規化係数を使用)</td></tr> <tr> <td>(9)</td><td> $L' = L + B \times C + A \times LS$ $R' = R + B \times C + A \times RS$ </td><td>DVB-T (正規化係数を使用しない)</td></tr> </tbody> </table> <p>C:Center L:Left R:Right M:Monaural LS:Left Surround Rear RS:Right Surround Rear ダウンミックス後のチャンネルは、L'、R'、M と記載。 A、B は各規格で定義される係数。</p>			計算式	準拠規格	(1)	$L' = 1 / (1 + 1/\sqrt{2} + A) \times (L + 1/\sqrt{2} \times C + A \times LS)$ $R' = 1 / (1 + 1/\sqrt{2} + A) \times (R + 1/\sqrt{2} \times C + A \times RS)$	ISO/IEC13818-7、 ISO/IEC14496-3	(2)	$L' = 1 / (1 + 1/\sqrt{2} + 2 \times A) \times [L + 1/\sqrt{2} \times C - A \times (LS + RS)]$ $R' = 1 / (1 + 1/\sqrt{2} + 2 \times A) \times [R + 1/\sqrt{2} \times C + A \times (LS + RS)]$	(3)	$M = 1 / (3 + 2 \times A) \times [L + C + R + A \times (LS + RS)]$	(4)	$L' = 1/\sqrt{2} \times (L + 1/\sqrt{2} \times C + A \times LS)$ $R' = 1/\sqrt{2} \times (R + 1/\sqrt{2} \times C + A \times RS)$	ARIB STD-B21 5.2 版	(5)	$L' = 1/\sqrt{2} \times [L + 1/\sqrt{2} \times C - A \times (LS + RS)]$ $R' = 1/\sqrt{2} \times [R + 1/\sqrt{2} \times C + A \times (LS + RS)]$	(6)	$L' = L + 1/\sqrt{2} \times C + A \times LS$ $R' = R + 1/\sqrt{2} \times C + A \times RS$	ARIB STD-B21 5.3 版	(7)	$L' = L + 1/\sqrt{2} \times C - A \times (LS + RS)$ $R' = R + 1/\sqrt{2} \times C + A \times (LS + RS)$	(8)	$L' = 1 / (1 + B + A) \times (L + B \times C + A \times LS)$ $R' = 1 / (1 + B + A) \times (R + B \times C + A \times RS)$	DVB-T (正規化係数を使用)	(9)	$L' = L + B \times C + A \times LS$ $R' = R + B \times C + A \times RS$	DVB-T (正規化係数を使用しない)
	計算式	準拠規格																										
(1)	$L' = 1 / (1 + 1/\sqrt{2} + A) \times (L + 1/\sqrt{2} \times C + A \times LS)$ $R' = 1 / (1 + 1/\sqrt{2} + A) \times (R + 1/\sqrt{2} \times C + A \times RS)$	ISO/IEC13818-7、 ISO/IEC14496-3																										
(2)	$L' = 1 / (1 + 1/\sqrt{2} + 2 \times A) \times [L + 1/\sqrt{2} \times C - A \times (LS + RS)]$ $R' = 1 / (1 + 1/\sqrt{2} + 2 \times A) \times [R + 1/\sqrt{2} \times C + A \times (LS + RS)]$																											
(3)	$M = 1 / (3 + 2 \times A) \times [L + C + R + A \times (LS + RS)]$																											
(4)	$L' = 1/\sqrt{2} \times (L + 1/\sqrt{2} \times C + A \times LS)$ $R' = 1/\sqrt{2} \times (R + 1/\sqrt{2} \times C + A \times RS)$	ARIB STD-B21 5.2 版																										
(5)	$L' = 1/\sqrt{2} \times [L + 1/\sqrt{2} \times C - A \times (LS + RS)]$ $R' = 1/\sqrt{2} \times [R + 1/\sqrt{2} \times C + A \times (LS + RS)]$																											
(6)	$L' = L + 1/\sqrt{2} \times C + A \times LS$ $R' = R + 1/\sqrt{2} \times C + A \times RS$	ARIB STD-B21 5.3 版																										
(7)	$L' = L + 1/\sqrt{2} \times C - A \times (LS + RS)$ $R' = R + 1/\sqrt{2} \times C + A \times (LS + RS)$																											
(8)	$L' = 1 / (1 + B + A) \times (L + B \times C + A \times LS)$ $R' = 1 / (1 + B + A) \times (R + B \times C + A \times RS)$	DVB-T (正規化係数を使用)																										
(9)	$L' = L + B \times C + A \times LS$ $R' = R + B \times C + A \times RS$	DVB-T (正規化係数を使用しない)																										

(1) 準拠規格

- 第 2 引数 `sel_std = 0` が設定された場合は、ISO/IEC13818-7、ISO/IEC14496-3 準拠のダウンミックス処理が実行されます。
- `sel_std = 1` (ARIB STD-B21 5.2 版準拠) が設定された場合は、ダウンミックス後の音量レベルが可能な限り同等な音量レベルとなるようにダウンミックスされます。
この場合、ダウンミックスの演算によって、オーバーフローによるクリッピングが発生する可能性があります。第 5 引数 `mixdown_scale` に 1 または 2 を設定することによって、入力 PCM データをあらかじめ 1/2, 1/4 にスケール後ダウンミックス処理を行い、オーバーフローを回避することが可能です。
- `sel_std = 2`、または 3 (ARIB STD-B21 5.3 版準拠、(オーバーフロー対策を実施しないまたは、実施する)) が設定された場合は、音量正規化係数がダウンミックス後の PCM データに乗算されません。
 - オーバーフロー対策を実施しない(`sel_std=2`)が指定された場合、オーバーフロー発生時、本関数は、オーバーフローした PCM を出力し、戻り値 `RSACPD_RTB_CHECK`(ステータスコード: `RSACPD_WARN_MIXDOWN_OVF`)で終了します。
 - オーバーフロー対策を実施(`sel_std=3`)が指定された場合、オーバーフロー発生時、本関数は、オーバーフロー発生時のサンプル以降、準拠規格を ARIB STD-B21 5.2 版(`sel_std=1`)に変更し音量正規化係数を乗算した PCM を出力します。
- 本関数実行後に、`*sel_std` の値を確認することで実施された規格を確認することが可能です。

(2) ダウンミックスモード

- 第 3 引数 `mixdown_mode = 2` (モノラルにダウンミックス)が設定された場合は、第 2 引数 `sel_std` の値に関わらず、ISO/IEC13818-7、ISO/IEC14496-3 準拠のモノラルダウンミックス処理が実施されます。
- `mixdown_mode = 1` が設定され、かつ PCE を取得し、`RSACPD_PCE` 型構造体メンバの `matrix_mixdown_idx_present = 1`、`pseudo_surround_enable = 1` の場合は、外部擬似サラウンドプロセッサ用ダウンミックスを実施します。
この条件を満たさずに `mixdown_mode` に 1 が設定された場合は、`mixdown_mode = 0` と同様の処理を行います。
- 外部擬似サラウンドプロセッサ用ダウンミックス処理を実施したかどうかは、本関数実行後に、`*mixdown_mode` の値によって確認することが可能です。
PCE を未取得または、`RSACPD_PCE` 型構造体メンバの `matrix_mixdown_idx_present = 0` の場合は、`matrix_mixdown_idx = 0` と同様の処理を行います。
- 本ミドルウェアは、ISO/IEC 規格、ARIB 規格、DVB-T 規格に対応したダウンミックス処理をサポートしています。
DVB-T 規格に準拠したダウンミックス処理は、ビットストリームの DSE エlement に格納された情報をもとにダウンミックス係数を計算し、ISO/IEC 準拠の `matrix_mixdown` 処理をより高解像度(詳細な係数設定)で行います。また、マルチチャネルからステレオへのダウンミックスのみをサポートしています。

DVB-T 規格に準拠したダウンミックス処理を行う場合は、デコード実行前に `RSACPD_SetDSE0` 関数を実行し(「3.2.13 `RSACPD_SetDSE`」参照)、本引数 `mixdown_mode` を 3、4 または 5 に設定し、かつ 第 2 引数 `sel_std` を 0 に設定してください。

(注) DSE にダウンミックス情報が格納されていない場合、または、第 2 引数 `sel_std` に 0 以外の値が設定された場合は、`mixdown_mode = 0`、第 2 引数 `sel_std` に従ったダウンミックス処理を行います。

- DVB-T 規格対応には、次の 2 つの計算式があります。
 - 音量正規化係数を使う: `mixdown_mode = 3`
 - 音量正規化係数を使わない: `mixdown_mode = 4`、または 5
(オーバーフロー対策を実施しない、または実施する)
- オーバーフロー対策を実施しない(`mixdown_mode = 4`)が指定された場合、オーバーフロー発生時、本関数は、オーバーフローした PCM を出力し、戻り値 `RSACPD_RTb_CHECK`(ステータスコード: `RSACPD_WARN_MIXDOWN_OVF`)で終了します。
- オーバーフロー対策を実施(`mixdown_mode = 5`)が指定された場合、オーバーフロー発生時、オーバーフロー発生時のサンプル以降、ダウンミックスモードを 3 に変更し音量正規化係数を乗算した PCM を出力します。
- DVB-T 規格対応したダウンミックス実行時において、一度 DSE 情報を取得すると次に新しい DSE 情報を取得するまで、その DSE 情報を参照し続けます。
- * `mixdown_mode` の値を確認することで、実施されたダウンミックスモードを確認することが可能です。

<注意事項>

- (1) 本関数が正常終了(または、ワーニング終了)した場合、ダウンミックス前の出力 PCM データはダウンミックス後の出力 PCM データで上書きされます。同様に `RSACPD_OUT_INFO` 型構造体のその他のメンバもダウンミックス後に上書きされます。
- (2) ステレオへのダウンミックスで出力される PCM データは非インタリーブ PCM データ形式のため、2ch インタリーブ PCM データ形式に変換する場合は、本関数の実行後、`RSACPD_InterleavePCM0`関数を実行して下さい。
- (3) オーバーフローの発生状況によって 適用する準拠規格(ダウンミックスモード)を変更する方式(第 2 引数 `sel_std = 3`、`mixdown_mod = 5`) は、ルネサス独自の実装方法です。
- (4) <注意事項>(3)記載の方式において、Left チャンネルでオーバーフローを検出した場合、Right チャンネルにおいても変更後の準拠規格が適用されます (Left チャンネルの処理を先に実施するため)。Right チャンネルでのみオーバーフローを検出した場合は、Left チャンネルに適用する準拠規格は変更されません。
- (5) 1 つの `raw_data_block` 内に、複数の DSE エlementが存在する場合は、最初に検出したダウンミックス情報に従います。
- (6) 準拠規格を ARIB STD-B21 5.3 版設定時 (`sel_std=3`)、準拠規格 ARIB STD-B21 5.2 版(`sel_std = 1`)に変更後の計算において、オーバーフローが発生した場合は、オーバーフローした PCM を出力し、戻り値 `RSACPD_RTb_CHECK`(ステータスコード: `RSACPD_WARN_MIXDOWN_OVF`)で終了します。

3.2.16 RSACPD_SetSAC

構文	int RSACPD_SetSAC(RSACPD_AAC *aac, RSACPD_SAC *sac)	
機能	MPEG サラウンド機能を実行するために必要な SAC(スペーシャル情報)をビットストリームから取得するための領域の設定を行う。	
引数	I/O	意味
RSACPD_AAC *aac	I/O	RSACPD_AAC 型構造体へのポインタ
RSACPD_SAC *sac	I	RSACPD_SAC 型構造体へのポインタ
戻り値	マクロ名	意味
0	RSACPD_RTN_GOOD	正常終了
-1	RSACPD_RTN_ERROR	異常終了
説明	<p><本関数の実行について></p> <p>MPEG サラウンド情報を含むビットストリームデコード時、スペーシャル情報を取得する必要がある場合は、RSACPD_Open0関数または RSACPD_SetDecOpt0関数実行後、はじめて RSACPD_Decode0関数または RSACPD_Skip0関数を実行する前に、必ず本関数を実行して下さい</p> <p><機能詳細></p> <p>RSACPD_Decode0関数、RSACPD_Skip0関数実行時にスペーシャル情報を検出した場合、RSACPD_SAC 型構造体 sac にスペーシャル情報が出力されます。</p> <p><注意事項></p> <ol style="list-style-type: none"> (1) 本関数を実行しても、MPEG サラウンド機能は実行されません。MPEG サラウンド機能につきましては、国際標準規格 ISO/IEC 23003-1 を参照してください。 (2) RSACPD_SAC 型構造体は、アプリケーションプログラムで領域確保することになります。詳細については「7.8 RSACPD_SAC type structure」を参照して下さい。 (3) 本関数がコールされない場合、入力ビットストリームにスペーシャル情報が含まれていても情報を取得することができませんが、デコード機能は継続して実行可能です。 (4) スペーシャル情報を取得したブロック(raw_data_block)の次のブロックにスペーシャル情報がなかった場合は、RSACPD_SAC 型構造体メンバの present は 0 に設定されますが、その他のメンバはクリアされません。 (5) 1 ブロック(raw_data_block)中に 1 つ以上のスペーシャル情報が検出された場合、最後に検出された情報が RSACPD_SAC 型構造体出力されます。 	

3.2.17 RSACPD_SetDRC

構文	int RSACPD_SetDRC(RSACPD_AAC *aac, RSACPD_DRC *drc, int hi, int lo, int ref_level)		
機能	ダイナミックレンジコントロール情報を設定する。		
引数	I/O	意味	
RSACPD_AAC *aac	I/O	RSACPD_AAC 型構造体へのポインタ	
RSACPD_DRC *drc	O	Pointer to the RSACPD_DRC type structure	
int hi	I	DRC カットスケール (設定可能値: 0 ～ 100)	
int lo	I	DRC ブーストスケール (設定可能値: 0 ～ 100)	
int ref_level	I	DRC 出力レベル (設定可能値: 0 ～127, -1)	
戻り値	マクロ名	意味	
0	RSACPD_RTN_GOOD	正常終了	
-1	RSACPD_RTN_ERROR	異常終了	
説明	<p><本関数の実行について></p> <p>ダイナミックレンジ(DRC)機能を実行する場合は、RSACPD_Open0関数が正常終了後に必ず本関数を実行し DRC 情報を設定して下さい。</p> <p><機能詳細></p> <p>本関数はビットストリームに格納された DRC 情報を参照し、数フレームまたは、全フレームの出力 PCM のダイナミックレンジカット(コンプレス)またはブーストします。</p> <p>第 3、4 引数 hi、lo に値 (0～100) を設定することにより、ビットストリーム内のカット(コンプレス)係数、ブースト係数をスケーリングすることが可能です。</p> <p>本引数に 100 を設定した場合、ビットストリーム内の係数はスケーリングせずにフルスケールで参照され、出力 PCM のレベルは最大カット (または、最小ブースト)されます。</p> <p>本引数に 0 が設定された場合は、カット (または、ブースト) 処理は実施されません。</p> <p>また、第 5 引数 ref_level に値 (0～127) を設定することによりビットストリーム内の DRC プログラムリファレンスレベル(prog_level)を調整(出力レベル = 第 5 引数 ref_level – prog_level)することが可能です。本引数に -1 が設定された場合は、本ミドルウェアはビットストリーム内の DRC プログラムリファレンスレベルを参照せず、カット、ブースト係数のみが参照します。</p> <p><注意事項></p> <p>(1) 第 3、第 4 引数 hi、lo に、設定可能値以外が設定された場合は、100 を設定して継続動作します。</p> <p>(2) 第 5 引数 ref_level に設定可能値以外が指定された場合の動作は、以下となります。</p> <ul style="list-style-type: none">・ 128 以上が指定された場合、第 5 引数 ref_level に 127 を設定して処理を継続します。・ -2 以下が指定された場合、第 5 引数 ref_level に-1 を設定して処理を継続します。 <p>(3) 1 raw_data_block 内に含まれる DRC スレッドの個数は最大 3 個までサポートしています。4 番目以降の DRC スレッドの情報は無視します。</p>		

4 ユーザ記述関数

4.1 ユーザ記述関数

本ミドルウェアでは、下記の関数をユーザが記述する必要があります。

ユーザ記述関数は、入力バッファ内のビットストリームデータを本ミドルウェアに渡すために、本ミドルウェアから呼び出されます。

構文	unsigned int RSACPD_GetData(unsigned char *wpt, int size)		
機能	入力バッファにビットストリームの補充を行う。		
引数		I/O	意味
unsigned char *wpt		I	入力バッファへのポインタ
int size		I	入力バッファに入力可能なサイズ(バイト)
戻り値	入力バッファに入力したバイト数		
説明	<p><本関数の実行について> 本関数は入力バッファ内のビットストリームデータが空になった場合、本ミドルウェアから呼び出されます。</p> <p><機能詳細> ビットストリームデータを引数 wpt で指定されたアドレス以降に引数 size で示されたデータ数以下のデータを入力します。入力するデータが無い場合は、戻り値に 0 を設定して下さい。</p> <p>本関数の戻り値が 0 の場合は、本ミドルウェアは入力ビットストリームの終了と判断し、ステータス : RSACPD_ERR_DATA_EMPTY で異常終了します。この機能を利用して、ユーザはデコード処理を強制停止させることも可能です。</p> <p>本関数の第 1 引数は、RSACPD_Open0関数の第 2 引数(入力バッファ先頭アドレス)が渡されます。 本関数の第 2 引数は、RSACPD_Open0関数の第 3 引数(入力バッファサイズ)が渡されます。</p> <p><注意事項> ユーザは本関数の戻り値が、第 2 引数(入力バッファに入力可能なサイズ)以下であることを必ず保障して下さい。それ以外は、動作保証外となります。</p>		

(注) ユーザ記述関数の名前は、ユーザが自由に設定可能です。本ドキュメントでは、RSACPD_GetData としています。

4.2 動作概要

ユーザ記述関数の動作概要を図 4.1に示します。

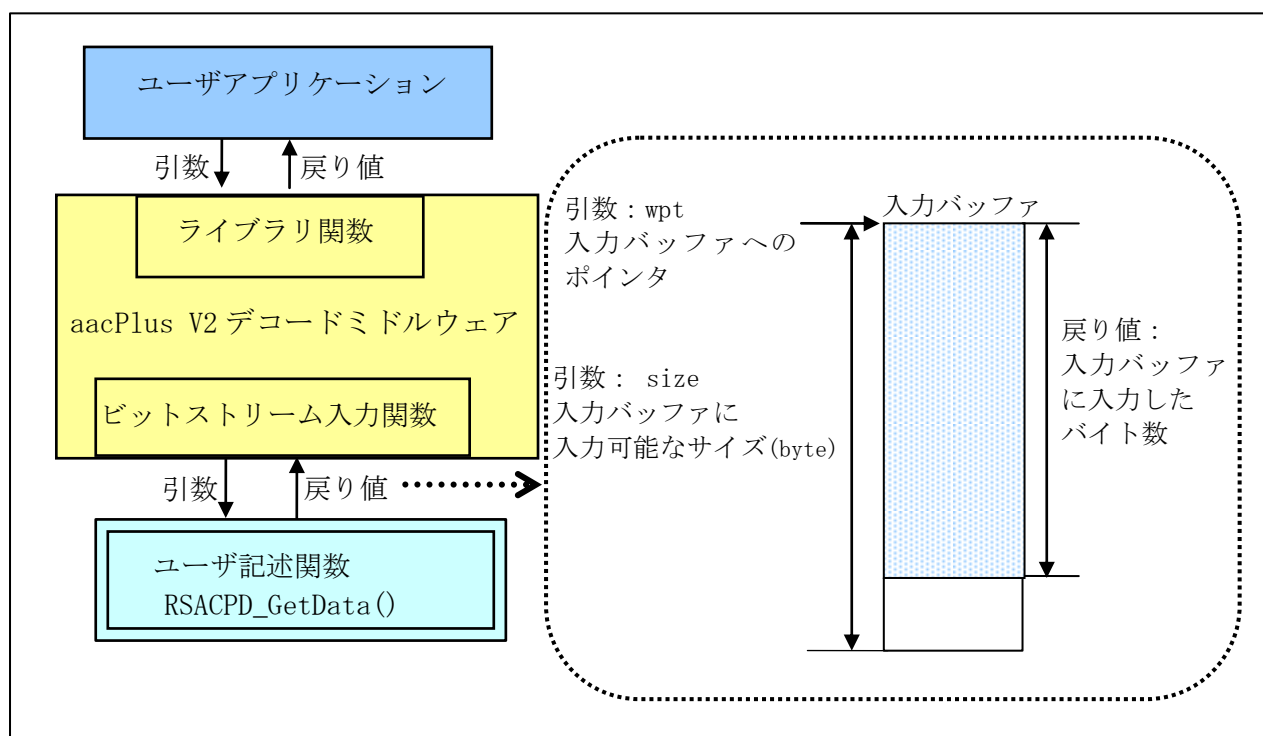


図 4.1 動作概要

5 チャンネル構成

5.1 チャンネル数定義

本ミドルウェアがサポートする音声モードに応じた、PCM データの出力先 (RSACPD_OUT_INFO 型構造体メンバ) を表 5.1、表 5.2に示します。デコード処理の実行後、本構造体に設定される情報を利用して、チャンネル構成にあった PCM データの取り出しを行って下さい。

表 5.1 チャンネル構成と PCM データの出力先(1)

音声モード		pcm_cf	pcm_lf	pcm_rf	pcm_ls	pcm_rs	pcm_lfe	エレメント出現順				channel Mode
								1	2	3	4	
モノラル	(1)	M						SCE				0
ステレオ	(2)		L	R				CPE				1
3/0	(3)	C	L	R				SCE	CPE			4
3/1	(4)	C	L	R	MS			SCE	CPE	SCE		5
3/2	(5)	C	L	R	LS	RS		SCE	CPE	CPE		6
3/2 + LFE (5.1)	(6)	C	L	R	LS	RS	LFE	SCE	CPE	CPE	LFE	7
デュアルモノラル	(0)		M	M				SCE	SCE			2
2/1	(0)		L	R	MS			CPE	SCE			8
2/2	(0)		L	R	LS	RS		CPE	CPE			9

「ARIB STD-B32 2.1 版」標準規格を参考に定義

(注 1)…()内の数字は、ADTS ヘッダの channel_configuration の値

(注 2)…音声モードの「/」は前方/後方のスピーカのチャンネル数を示す。

例) 3/1 = 前方 3ch + 後方 1ch

C:Center L:Left R:Right M:Monaural

MS:Monaural Surround Rear(Rear Surround) LS:Left Surround Rear

RS:Right Surround Rear LFE:Low Frequency Effects

(注 3)…channelMode は、RSACPD_OUT_INFO 型構造体のメンバを示す。

(注 4)…PCM データが割り当てられていないメンバ(RSACPD_OUT_INFO 型構造体)には、NULL が設定されます。

表 5.1で定義した以外の音声モードの場合、合計チャンネル数が本ミドルウェアのサポート範囲であれば、RSACPD_Decode()関数はワーニング (ステータス : RSACPD_RTN_UNSUPPORTED_CH_CFG) を返し、正常に PCM データを出力します。この時、PCM データの出力先は表 5.2のようになります。

但し、RSACPD_SetPCEArea()関数を実行し PCE を取得した場合は、PCE 情報を使用します。「3.2.2 RSACPD_SetPCEArea」を参照してください。

デコード中にチャンネル情報(PCE 情報、ADTS ヘッダの channel configuration、raw_data_block のエレメント構成)が変更された場合、デコーダは最新の情報に従います。

ADTS ヘッダの channel configuration と PCE 情報が同じフレームに存在した場合は、デコーダは PCE 情報に従います。

PCE に含まれるチャンネル数情報が、実際の raw_data_block に含まれる各エレメントで構成されるチャンネル数と異なる場合は、PCE 情報は無視されます。ADTS ヘッダの channel configuration と raw_data_block に含まれるチャンネル数が異なる場合は、デコーダはエラー(ステータス : RSACPD_WARN_ILLEGAL_CHAN_CONFIG)を返します。

表 5.2 チャンネル構成と PCM データの出力先(2)

音声モード	エレメント出現順 チャンネルマッピング					channelMode
3/2	SCE	SCE	SCE	SCE	SCE	-1
	pcm_cf	pcm_lf	pcm_rf	pcm_ls	pcm_rs	
3/2	SCE	SCE	SCE	CPE		-1
	pcm_cf	pcm_lf	pcm_rf	pcm_ls	pcm_rs	
3/2	SCE	SCE	CPE		SCE	-1
	pcm_cf	pcm_lf	pcm_ls	pcm_rs	pcm_rf	
3/2	SCE	CPE		SCE	SCE	-1
	pcm_cf	pcm_lf	pcm_rf	pcm_ls	pcm_rs	
3/2	CPE		SCE	SCE	SCE	-1
	pcm_lf	pcm_rf	pcm_cf	pcm_ls	pcm_rs	
3/2	CPE		SCE	CPE		-1
	pcm_lf	pcm_rf	pcm_cf	pcm_ls	pcm_rs	
3/2	CPE		CPE		SCE	-1
	pcm_lf	pcm_rf	pcm_ls	pcm_rs	pcm_cf	
3/1	SCE	SCE	SCE	SCE	-	-1
	pcm_lf	pcm_rf	pcm_ls	pcm_rs	-	
3/1	SCE	SCE	CPE		-	-1
	pcm_lf	pcm_rf	pcm_ls	pcm_rs	-	
3/1	CPE		SCE	SCE	-	-1
	pcm_lf	pcm_rf	pcm_ls	pcm_rs	-	
3/0	SCE	SCE	SCE	-	-	-1
	pcm_cf	pcm_lf	pcm_rf	-	-	

(注 1)...LFE チャンネルの PCM データは、常に pcm_lfe で指定される領域に出力されます。

(本表では省略しています)。

(注 2)...CPE は必ず pcm_lf, pcm_rf のペア、または pcm_ls, pcm_rs のペアとして出力されます。

(注 3)...SCE が 3 つ、CPE が 1 つの場合(SCE,SCE,SCE,CPE の順)は、CPE は pcm_ls, pcm_rs のペアで出力されます。

6 処理フロー

入力ビットストリームのデータフォーマットには、RawDataStream 形式、ADTS 形式、ADIF 形式、LOAS 形式があります。

以下にそれぞれのデータフォーマットごとの、デコード時の処理フローを示します。

6.1 RawDataStream 形式デコード

RawDataStream 形式のビットストリームのデコード手順を 図 6.1 に示します。デコーダの初期化処理を行った後、1 フレームごとにデコード処理を繰り返します。

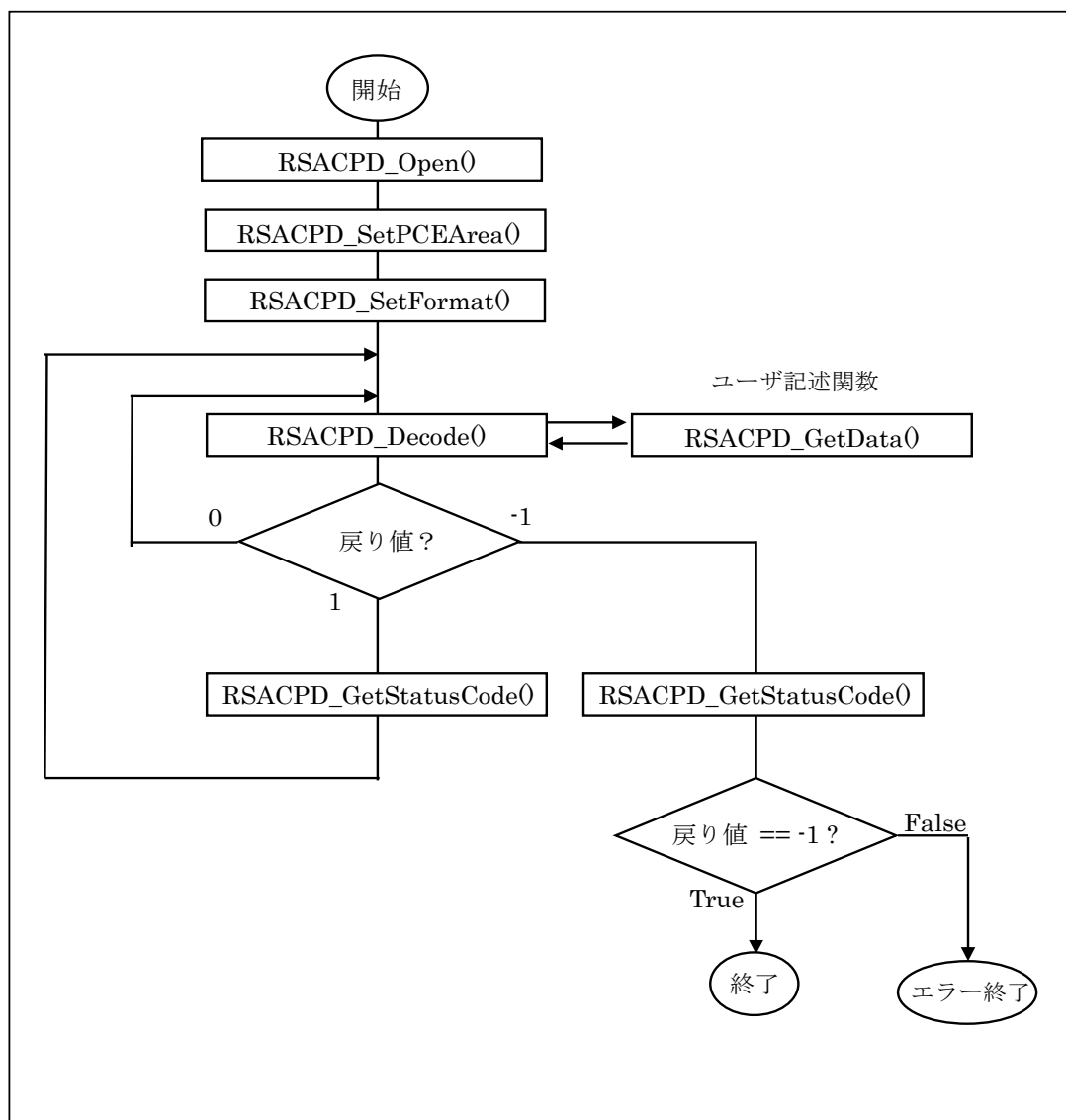


図 6.1 RawDataStream 形式デコード処理の流れ（例）

6.2 ADTS 形式デコード

図 6.2は、ADTS 形式のビットストリームのデコード手順を示すフローチャートです。ADTS 形式のビットストリームの各ADTSフレームの先頭には同期語から始まるヘッダ(ADTSヘッダ)が存在します。各 ADTS フレームはいくつかの raw_data_block で構成されています。デコーダの初期化を行った後、ADTS ヘッダ情報を取得し、raw_data_block ごとにデコード処理を繰り返します。

ADTS フレーム内に未デコードの raw_data_block が存在する場合、RSACPD_DecodeStatus()関数により取得するデコードステータスの第 4 ビットに 1 が設定されています。

ADTS フレームに含まれる raw_data_block の数は、ADTS ヘッダ情報の "number_of_raw_data_blocks_in_frame" で取得することができます。

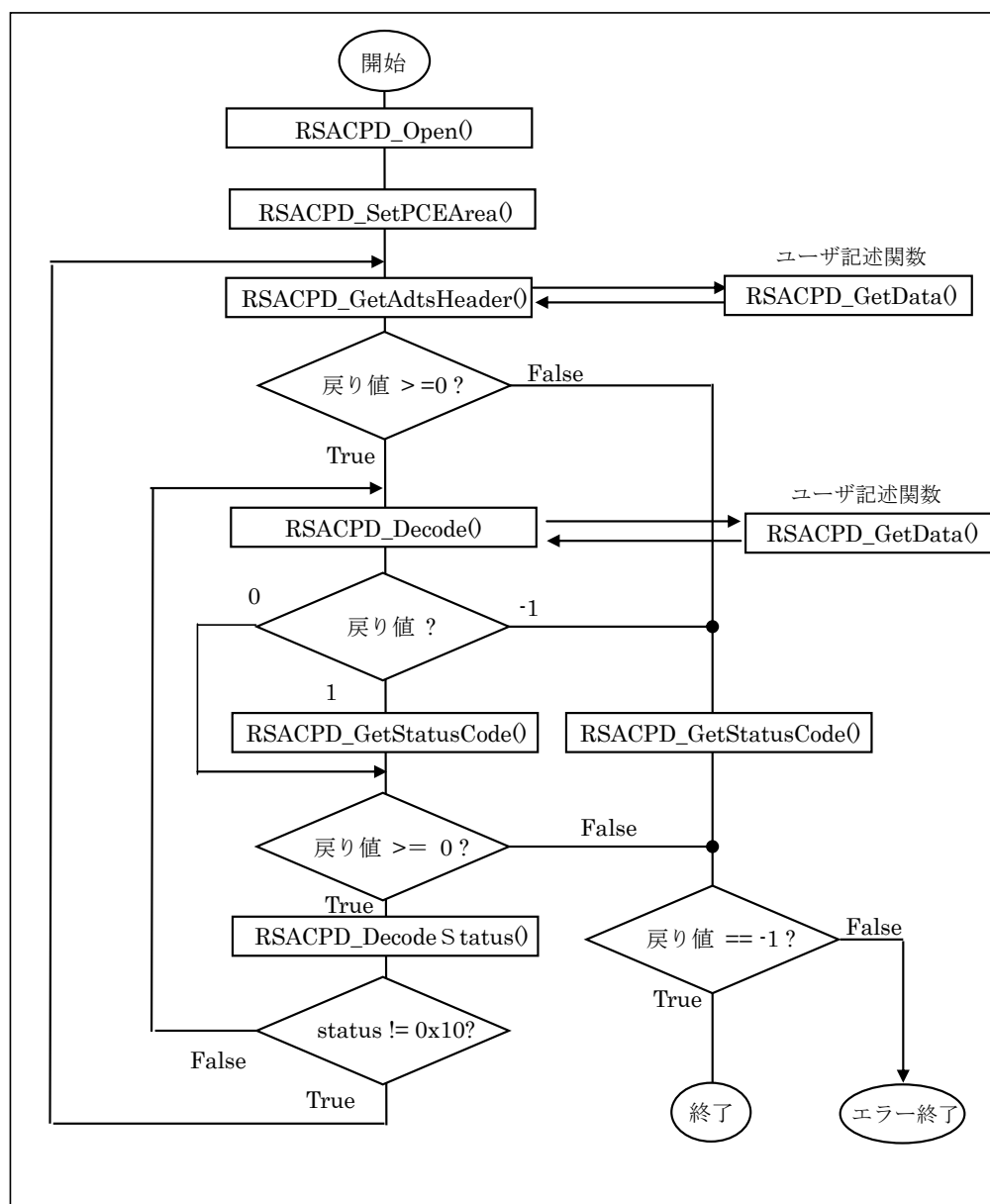


図 6.2 ADTS 形式デコード処理の流れ(例)

6.3 ADIF 形式デコード

図 6.3は、ADIF 形式のビットストリームのデコード手順を示すフローチャートです。ADIF 形式のビットストリームは、ビットストリームの先頭に1つのヘッダ情報を持ちます。デコーダの初期化を行った後、ADIF ヘッダ情報を取得し、1 フレームごとにデコード処理を繰り返します。

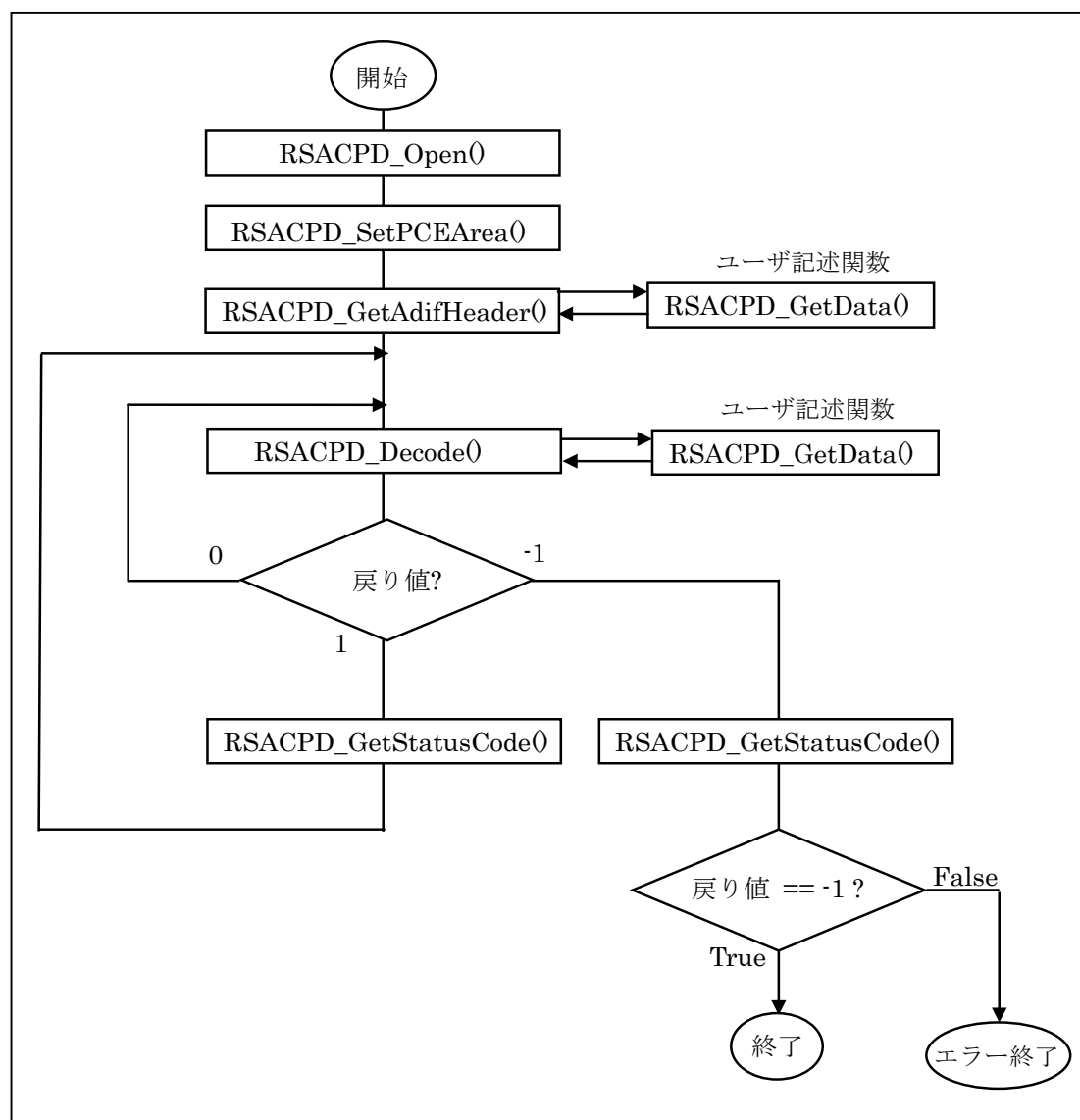


図 6.3 ADIF 形式デコード処理の流れ(例)

6.4 LOAS 形式デコード

LOAS 形式のビットストリームのデコード手順を示すフローチャートを以下に示します。

LOAS 形式のビットストリームの各フレームの先頭には同期語から始まるヘッダ(LOAS ヘッダ)が存在します。デコーダの初期化を行った後、LOAS ヘッダ情報を取得し、デコード処理を行います。

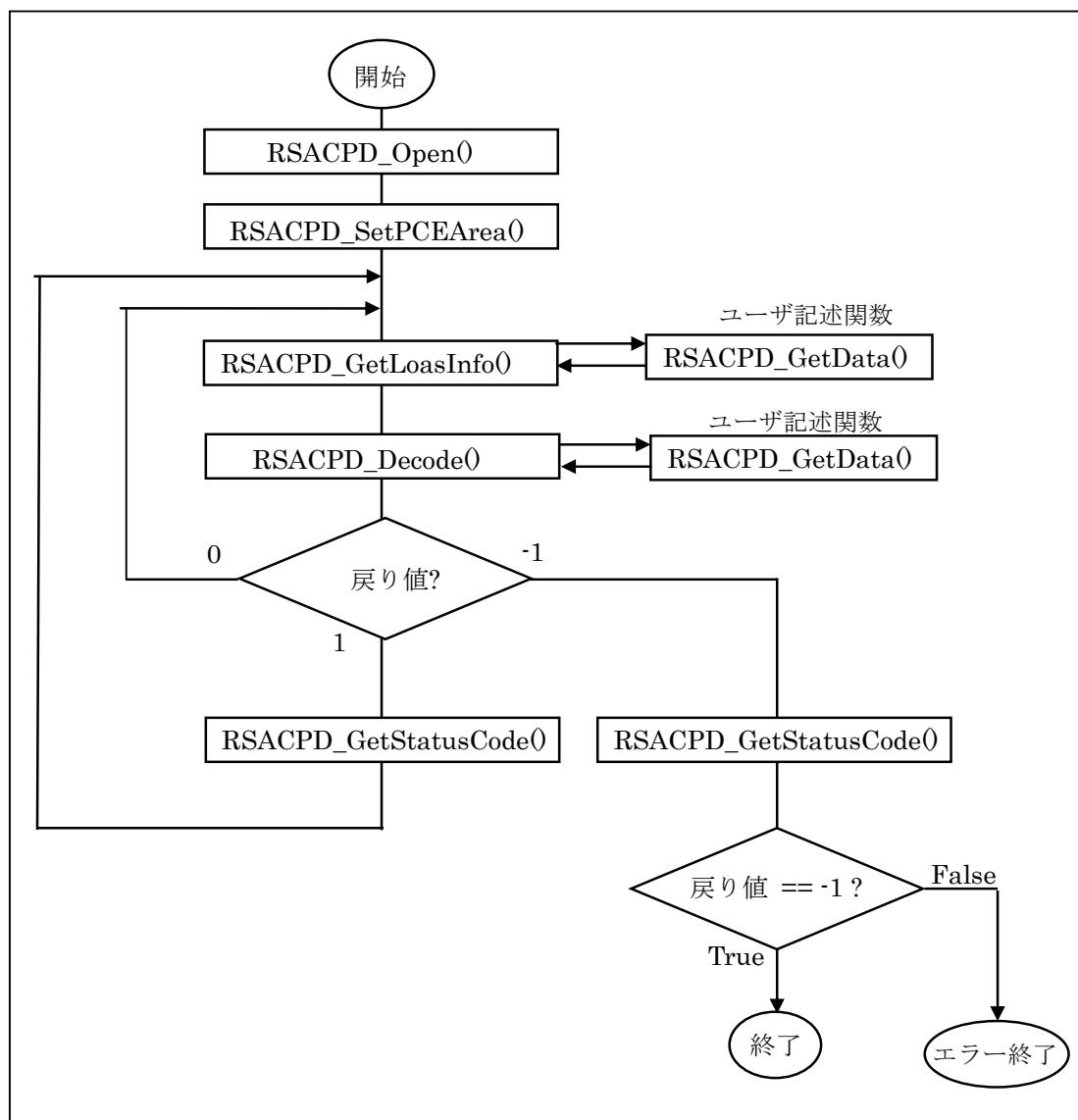


図 6.4 LOAS 形式デコード処理の流れ(例)

7 構造体

本ミドルウェアを組み込む際に、アプリケーションプログラムで領域を確保する必要がある構造体を表 7.1に示します。それぞれの構造体の型はライブラリヘッダファイル(RSACPD_ADL.h)に定義されています。

これらの構造体は、本ミドルウェアが使用するワークエリアですので、構造体の内容をアプリケーションプログラムで変更しないで下さい。

表 7.1 構造体一覧

No	構造体定義	サイズ	備考
1	RSACPD_AAC	約 225K バイト	必須
2	RSACPD_PCE	413 バイト	必須
3	RSACPD_AdifHeader	28 バイト	ADIF 形式のデコード時に必須
4	RSACPD_AdtsHeader	18 バイト	ADTS 形式のデコード時に必須
5	RSACPD_LoasInfo	36 バイト	LOAS 形式のデコード時に必須
7	RSACPD_OUT_INFO	52 バイト	必須
7	RSACPD_DSE	24 バイト	任意
8	RSACPD_SAC	1096 バイト	任意
9	RSACPD_DRC	2108 バイト	任意

7.1 RSACPD_AAC 型構造体

RSACPD_AAC 型構造体は、本ミドルウェアが使用するワーク領域です。本ミドルウェアを組み込む際に、アプリケーションプログラムで領域を確保して下さい。本構造体には、ミドルウェアが使用する内部変数、ワークバッファが格納されていますので、アプリケーションが本構造体の内容を参照する必要はありません。データ構造は、本資料の付録に記載しています。

アプリケーションプログラムで本領域の値を変更しないように注意して下さい。

7.2 RSACPD_PCE 型構造体

RSACPD_PCE 型構造体は、raw_data_block 内の PCE エlement、ADIF ヘッダ内に存在する PCE 情報をアプリケーションプログラムが取得するための領域です。本ミドルウェアを組み込む際に、アプリケーションプログラムで領域を確保して下さい。また、必要に応じて、RSACPD_PCE 型構造体を配列化し、複数の PCE 情報を利用することが可能です。RSACPD_PCE 型構造体のデータ構造を以下に示します。

表 7.2 RSACPD_PCE 型構造体情報

メンバ名		内容
unsigned char	element_instance_tag	エレメントインスタンスタグ
unsigned char	profile	00:Main 01:LC (必須) 10:SSR 11:予約 (MPEG2)、LTP(MPEG4)
unsigned char	sampling_frequency_index	表 3.3を参照して下さい
unsigned char	num_front_channel_elements	フロントオーディオ構成エレメント数
unsigned char	num_side_channel_elements	サイドオーディオ構成エレメント数
unsigned char	num_back_channel_elements	リアオーディオ構成エレメント数
unsigned char	num_lfe_channel_elements	LFE オーディオ構成エレメント数
unsigned char	num_assoc_data_elements	関連データエレメント数
unsigned char	num_valid_cc_elements	オーディオデータに付加される CCE エレメント数
unsigned char	mono_mixdown_present	モノラルダウンミックスフラグ
unsigned char	mono_mixdown_element_number	モノラルダウンミックスエレメント数
unsigned char	stereo_mixdown_present	ステレオダウンミックスフラグ
unsigned char	stereo_mixdown_element_number	ステレオダウンミックスする CPE のエレメント数
unsigned char	matrix_mixdown_idx_present	マトリックスミックスダウン情報フラグ
unsigned char	matrix_mixdown_idx	サラウンドダウンミックス係数インデックス
unsigned char	pseudo_surround_enable	擬似サラウンドダウンミックスフラグ
unsigned char	front_element_is_cpe[16]	SCE/CPE のフロント指定フラグ
unsigned char	front_element_tag_select[16]	フロントエレメントとして取り扱われる SCE/CPE のインスタンスタグ
unsigned char	side_element_is_cpe[16]	SCE/CPE のサイドエレメント指定フラグ
unsigned char	side_element_tag_select[16]	サイドエレメントとして取り扱われる SCE/CPE のインスタンスタグ
unsigned char	back_element_is_cpe[16]	SCE、CPE のリアエレメント指定フラグ
unsigned char	back_element_tag_select[16]	リアエレメントとして取り扱われる SCE/CPE のインスタンスタグ
unsigned char	lfe_element_tag_select[4]	LFE のインスタンスタグ
unsigned char	assoc_data_element_tag_select[8]	DSE のインスタンスタグ
unsigned char	cc_element_is_ind_sw[16]	独立 CCE フラグ
unsigned char	valid_cc_element_tag_select[16]	CCE のインスタンスタグ
unsigned char	comment_field_bytes	後に続くコメントフィールドのバイト数
unsigned char	comment_field_data[256]	コメントフィールドデータ

7.3 RSACPD_AdifHeader 型構造体

RSACPD_AdifHeader 型構造体は、ADIF 形式のビットストリームから、ADIF ヘッダ情報を取得するための領域です。本ミドルウェアを組み込む際に、アプリケーションプログラムで領域を確保して下さい。データ構造を以下に示します。

表 7.3 RSACPD_AdifHeader 構造体情報

メンバ名	内容
unsigned char adif_id[4]	“ADIF” (0x41,0x44,0x49,0x46)
unsigned char copyright_id_present	0: コピーライト ID を付加 1: 同情報を付加しない
unsigned char copyright_id[9]	コピーライト ID
unsigned char original_copy	0: コピー 1: オリジナル
unsigned char home	予約
unsigned char bitstream_type	0: 固定ビットレート 1: 可変ビットレート
unsigned long bitrate	固定ビットレート時:ビットレート 可変ビットレート時:最大ビットレート
unsigned char num_program_config_element	num_program_config_element + 1 が program_config_element 数 (値が 0 の時、program_config_element は 1 個)

7.4 RSACPD_AdtsHeader 型構造体

RSACPD_AdtsHeader 型構造体は、ADTS 形式のビットストリームから、ADTS フレームごとに ADTS ヘッダ情報を取得するための領域です。本ミドルウェアを組み込む際に、アプリケーションプログラムで領域を確保して下さい。データ構造を以下に示します。

表 7.4 RSACPD_AdtsHeader 型構造体情報

メンバ名	内容
unsigned char ID	0: MPEG-4 1: MPEG-2
unsigned char layer	00: 予約 01: レイヤ 3 10: レイヤ 2 11: レイヤ 1 (AAC/aacPlus 符号化ビットストリームの場合:00)
unsigned char protection_absent	0: 誤り検出訂正情報付加、 1: 誤り検出訂正情報を付加しない
unsigned char profile	00: Main

		01: LC (必須) 10: SSR 11: 予約(MPEG2)、LTP(MPEG4)
unsigned char	sampling_frequency_index	表 3.3を参照して下さい
unsigned char	private_bit	プライベートビット
unsigned char	channel_configuration	チャンネル構成
unsigned char	original_copy	0: コピー 1: オリジナル
unsigned char	home	予約
unsigned char	copyright_identification_bit	0: コピーライトを付加、 1: 同情報を付加しない
unsigned char	copyright_identification_start	0: コピーライト情報なし、 1: コピーライト情報あり
unsigned short	frame_length	ADTS フレーム長 (1ADTS フレームのバイト数)
unsigned short	adts_buffer_fullness	ビット貯蓄の状態
unsigned char	number_of_rawdata_blocks_in_frame	ブロック(raw_data_block)数 注: 値が 0 の場合は 1 ブロック (値を+1 したブロック数が存在する)

7.5 RSACPD_LoasInfo 型構造体

RSACPD_LoasInfo 型構造体は、LOAS ヘッド情報を格納するための構造体です。本ミドルウェアを組み込む際に、アプリケーションプログラムで領域を確保して下さい。

RSACPD_LoasInfo 型構造体のデータ構造を以下に示します。

表 7.5 RSACPD_LoasInfo 型構造体情報

メンバ名		内容
unsigned short	audioMuxLengthBytes	後に続く AUdioMuxElement()のバイトアライメントを考慮したバイト数
unsigned char	useSameStreamMux	前フレームのマルチプレックスコンフィギュレーションデータを現在のフレームで参照するかどうかのフラグ
unsigned char	audioMuxVersion	使用するマルチプレックスシンタックスデータの種類 0: デフォルト 1: taraBufferFullness の転送及び各 AudioSpecificConfig()のデータ長の転送をサポートする
unsigned char	audioMuxVersionA	使用するシンタックスデータの種類 0: デフォルト 1: 予約(非サポート)
unsigned char	allStreamsSameTimeFraming	PayloadMux()に多重化されたすべてのペイロードが共通の time base を共有するかどうかのフラグ 0: 共有しない (非サポート) 1: 共有する

unsigned char	numSubFrames	“numSubFrames + 1”が多重化された PayloadMux() フレーム数 を示す。 最小値(0)の時、多重化されたサブフレーム数は 1
unsigned char	numSubFramesIndex	現在の PayLoadMux() フレームの値
unsigned char	numProgram	“numProgram + 1”が多重化されたプログラム数 を示す。 最小値(0)の時、多重化されたプログラム数は 1 最小値(0)以外は、非サポート
unsigned char	numLayer	“numLayer + 1”が多重化されたスケーラブルレイヤ 数 を示す。 最小値(0)の時、多重化されたスケーラブルレイヤは 1 最小値(0)以外は、非サポート
unsigned char	otherDataPresent	オーディオペイロード以外のデータ検出フラグ 0: オーディオペイロード以外のデータが多重化 されていない 1: オーディオペイロード以外のデータが多重化 されている
unsigned char	audioObjectType	ビットストリームの audioObjectType 2: AAC - LC 5: AAC – SBR 29: AAC - PS その他: 非サポート
Unsigned long	otherDataLenBits	otherData のビット長
unsigned char	crcCheckPresent	StreamMuxConfig() データの誤り検出訂正情報(CRC)付加 フラグ 0: 誤り検出訂正情報(CRC)が付加されていない 1: 誤り検出訂正情報(CRC)が付加されている
unsigned char	crcChecksum	誤り検出訂正情報(CRC) crcCheckPresent ビットを除く StreamMuxConfig()全体を 対象とした CRC8 多項式により生成された CRC 情報。 (注) LOAS 形式の場合は CRC チェックを行いません。
unsigned char	frameLengthType	ペイロードのフレーム長タイプ 0: 可変フレーム長のペイロード PayLoadLengthInfo()に 8 ビットコードで格納される 1: 固定フレーム長のペイロード StreamMuxConfig()にフレーム長(ビット)が格納される その他: 非サポート
unsigned char	frameLengthFlag	フレーム長、およびスペクトルデータ長の指定 0: 1024/ 128 の IMDCT を使用する。フレーム長は、1024。 1: 960/ 120 の IMDCT を使用する。フレーム長は、960。
unsigned short	frameLength	frameLengthType が 1 の場合のペイロードのフレーム長 ペイロード長は、8 x (frameLength + 20)ビット
unsigned char	channelConfiguration	出力チャンネルコンフィギュレーション (4 ビットフィールド)

unsigned char	samplingFrequencyIndex	サンプリング周波数(Sampling_frequency_index) (4 ビットフィールド) 表 3.3を参照してください
unsigned long	samplingFrequency	samplingFrequencyIndex に 0xF が指定された場合のサンプリング周波数
unsigned long	extensionSamplingFrequency	extensionSamplingFrequencyIndex に 0xF が指定された場合の extensionAudioObjectType()に対応する extension ツールの出力サンプリング周波数
unsigned char	extensionFlag	extension ツール検出フラグ
unsigned char	extensionSamplingFrequencyIndex	extensionAudioObjectType()に対応する extension ツールの出力サンプリング周波数(Sampling_frequency_index) (4 ビットフィールド) 表 3.3を参照してください

「非サポート」に関しては、本ミドルウェアはステータスコード RSACPD_ERR_LOAS_INFO のエラーを返します。
詳細は、「8.2.5 RSACPD_GetLoasInfo」を参照してください。

7.6 RSACPD_OUT_INFO 型構造体

RSACPD_OUT_INFO 型構造体は、チャンネルモード、出力 PCM データのチャンネル数、PCM データへのポインタを本ミドルウェアが格納するための構造体です。

本ミドルウェアを組み込む際に、アプリケーションプログラムで領域を確保して下さい。

RSACPD_OUT_INFO 型構造体のデータ構造を以下に示します。

表 7.6 RSACPD_OUT_INFO 型構造体情報

メンバ名	内容
int channelMode	0: モノラル
	1: ステレオ
	2: デュアルモノラル
	3: パラメトリックステレオ
	4: 3/0
	5: 3/1
	6: 3/2
	7: 3/2 + LFE (5.1ch)
	8: 2/1
	9: 2/2
	-1: その他 (デコードが異常終了、または表 5.1で定義した以外の音声モードの場合。例: 2.1 ch など)
int ChannelNumber	全チャンネル数
int nfch	フロントチャンネル数
int nsch	サイドチャンネル数 (必ず 0 が設定されます。)
int nbch	リアチャンネル数
int nlch	LFE チャンネル数

short *pcm_buf	PCM データ領域(アプリケーションにより確保) へのポインタ
short *pcm_cf	nfch = 1、3 の場合 : ・ Center チャンネルの出力 PCM データへのポインタ
short *pcm_lf	nfch = 2、3 の場合 : 次のいずれかのチャンネルの出力 PCM データへのポインタ ・ Left チャンネル ・ デュアルモノラル第 1 チャンネル(channelMode = 2 の場合)
short *pcm_rf	nfch = 2、3 の場合 : 次のいずれかのチャンネルの出力 PCM データへのポインタ ・ Right チャンネル ・ デュアルモノラル第 2 チャンネル (channelMode = 2 の場合)
short *pcm_ls	nbch = 1、2 の場合 : 次のいずれかのチャンネルの出力 PCM データへのポインタ ・ Left Surround Rear チャンネル ・ Monaural Surround Rear チャンネル (channelMode = 5、8 の場合)
short *pcm_rs	nbch = 2 の場合 : ・ Right Surround Rear の出力 PCM データへのポインタ
short *pcm_lfe	nlch = 1 の場合 : ・ LFE チャンネルの出力 PCM データへのポインタ

7.7 RSACPD_DSE type structure

RSACPD_DSE 型構造体は、DSE エlementに格納されて data_stream_element 情報を取得するための領域です。data_stream_element 情報を取得する必要がある場合は、本ミドルウェアを組み込む際に、アプリケーションプログラムで領域を確保して下さい。また、必要に応じて RSACPD_DSE 型構造体を配列化し、複数の DSE を取得することが可能です。領域が確保されなかった場合は、DSE 情報は読み飛ばされます。詳細は、「3.2.13 RSACPD_SetDSE」を参照して下さい。

RSACPD_DSE 型構造体のデータ構造を以下に示します。

表 7.7 RSACPD_DSE 型構造体情報

メンバ変数名	備考
unsigned int count	DSE に含まれるデータのバイト数
unsigned char present	1: DSE(data_stream_element) を取得した 0: DSE(data_stream_element) を取得せず
unsigned char element_instance_tag	インスタンスタグ
unsigned char ancillary_data_sync	0xBC : アンシラリデータが存在する others: アンシラリデータが存在しない
unsigned char mpeg_audio_type	MPEG オーディオタイプ

unsigned char	dolby_surround_mode	ドルビーサラウンドモード
unsigned char	downmixing_levels_MPEG4_status	1 : MPEG4 ダウンミックスレベルが存在する 0 : MPEG4 ダウンミックスレベルが存在しない
unsigned char	center_mix_level_on	1 : center_mix_level_value 情報を使用する 0 : center_mix_level_value 情報を使用しない
unsigned char	center_mix_level_value	Center チャンネルのマトリックスミックスダウンレベル
unsigned char	surround_mix_level_on	1 : surround_mix_level_value 情報を使用する 0 : surround_mix_level_value 情報を使用しない
unsigned char	surround_mix_level_value	サラウンドチャンネルのマトリックスミックスダウンレベル
unsigned char	audio_coding_mode_and_compression_status	1 : audio_coding_mode が存在する 0 : audio_coding_mode が存在しない
unsigned char	compression_on	1: compression_value 情報を使用する 0: compression_value 情報を使用しない
unsigned char	compression_value	モノラルダウンミックス実施時の重圧縮係数
unsigned char	coarse_grain_timecode_status	1: coarse_grain_timecode が存在する 0: coarse_grain_timecode が存在しない
unsigned short	coarse_grain_timecode	Coarse grain timecode 値
unsigned short	fine_grain_timecode	1: fine_grain_timecode が存在する 0: fine_grain_timecode が存在しない
unsigned char	fine_grain_timecode_status	Fine grain timecode 値

7.8 RSACPD_SAC type structure

RSACPD_SAC 型構造体は、extension_type エLEMENTの EXT_SAC_DATA に格納されたスペシャル情報を取得するための領域です。スペシャル情報を取得する必要がある場合は、本ミドルウェアを組み込む際に、アプリケーションプログラムで領域を確保して下さい。RSACPD_SAC 型構造体のデータ構造を以下に示します。

表 7.8 RSACPD_SAC 型構造体情報

メンバ変数名	備考
int present	1: SAC 情報(スペシャル情報)を取得 0: SAC 情報(スペシャル情報)を未取得
int ancType	アンシラリデータタイプ 0: MPEG Surround frame 1: MPEG Surround header 及び MPEG Surround frame その他: 予約
int ancStart	データブロックの開始を示す

int	ancStop	データブロックの終了を示す
int	ancDataSegmentByte[269]	スペシャル情報 ※下位 8 ビットが有効
int	count	ancDataSegmentByte[269]に取得したバイト数 (=有効ワード数)

7.9 RSACPD_DRC type structure

RSACPD_DRC 型構造体は、DRC 情報を取得するための領域です。本ミドルウェアを組み込む際に、アプリケーションプログラムで領域を確保して下さい。

表 7.9 RSACPD_DRC 型構造体情報

メンバ変数名		備考
int	Enable	1: DRC 処理を実施 0: DRC 処理を実施しない
int	hi	DRC カットスケール
int	lo	DRC ブーストスケール
int	digital_norm	1: プログラムリファレンスレベルを参照して出力レベルの正規化を実施 0: 出力レベルの正規化を実施しない
int	target_ref_level	DRC 出力レベル
int	prog_ref_level	ビットストリーム内のプログラム出力レベル
int	thread	DRC スレッドの数
RSACPD_DRC_Bitstream	drc_bit[3]	DRC ビットストリーム構造体
RSACPD_DRC_Info	drc_info[6]	DRC 情報格納構造体

(注) RSACPD_DRC_Bitstream、RSACPD_DRC_Info 構造体のデータ構造は、本資料の付録に記載しています。

8 ステータスコード一覧

本ミドルウェアの API 関数は、正常終了、ワーニング、または異常終了のエラーコードを返します。ワーニング、異常終了の場合は、RSACPD_GetStatusCode()関数で詳細なステータスを確認することが可能です。ステータスコードの一覧については、表 8.1を参照して下さい。

各 API 関数でのエラー復帰を含むアプリケーションプログラムの処理は、「8.2 API 関数とステータスコード」を参照して下さい。

複数のステータスを検出した場合は、ステータスコードは以下のように設定されます。

(1) すでにワーニングのステータスコードが設定されている場合

異常ステータスでのみ上書きされます。ワーニングステータスでは上書きされません。但し、以下の場合は例外となります。

- ・ワーニングステータスが RSACPD_WARN_DISAGREE_INPUT_FS 検出後、RSACPD_WARN_FI_MIX_AAC または、RSACPD_WARN_AUTO_DS_SBR が検出された場合上書きされます。
- ・ワーニングステータスが RSACPD_WARN_FI_MIX_AAC 検出後、RSACPD_WARN_AUTO_DS_SBR が検出された場合上書きされます。
- ・ワーニングステータスが RSACPD_WARN_FI_MULT_CCE または、RSACPD_WARN_ILLEGAL_CHAN_CONFIG 検出後、他のワーニングが検出された場合上書きされます。

(2) すでに異常のステータスコードが設定されている場合(RSACPD_SetDecOpt()関数でエラーコンシールモードを有効に設定されている場合のみ)、

ステータスコードの上書きはされず、最初に検出した異常のステータスコードが設定されたままとなります。

8.1 ステータスコード一覧

本ミドルウェアのステータスコード一覧を表 8.1に示します。

表 8.1 ステータスコード一覧

分類	値	コード名	内容
正常	0	RSACPD_RTN_GOOD	正常終了
ワーニング	5	RSACPD_WARN_PCE	1 ブロックに PCE を検出 (オーディオデータ含まず)
	7	RSACPD_WARN_NO_AUDIO_DATA	フレーム内にオーディオデータが存在しない (PCE のみの場合は、コード 5 を返す)
	10	RSACPD_WARN_ERR_ADTS_LEN	ADTS ヘッダの frame_length の値と実際にデコードした ADTS フレームサイズが不一致
	12	RSACPD_WARN_UNSUPPORTED_CH_CFG	未サポートのチャンネル構成を検出
	13	RSACPD_WARN_FI_MULTI_CCE	複数の CCE を検出
	21	RSACPD_WARN_FI_GARBAGE	ADTS ヘッダ取得時に同期語の前にごみデータを検出
	22	RSACPD_WARN_FI_MIX_AAC	入力ビットストリームの切り替わりを検出 (AAC ⇄ aacPlus)
	23	RSACPD_WARN_MIXDOWN_OVF	ダウンミックス処理中にオーバーフローが発生
	25	RSACPD_WARN_DISAGREE_SBR_DATA	SCE/CPE の数と SBR_DATA の数が不一致
	26	RSACPD_WARN_ERR_SBR_CRC_CHECK	SBR-CRC エラーを検出
	27	RSACPD_WARN_AUTO_DS_SBR	自動ダウンサンプリングを実施
	28	RSACPD_WARN_SEQUENCE	API 関数の実行順序が未サポート
	29	RSACPD_WARN_NOT_SUPPORT_SBR_FS	SBR のサンプリング周波数が未サポート
	30	RSACPD_WARN_LFE_RESTRICT_ERR	LFE の制約違反を検出
	31	RSACPD_WARN_SBR_HEADER_ERR	SBR ヘッダエラーを検出
	33	RSACPD_WARN_DISAGREE_INPUT_FS	入力サンプリング周波数の変化を検出
異常	39	RSACPD_WARN_AVG_PERFORMANCE	予約
	40	RSACPD_WARN_ILLEGAL_CHAN_CONFIG	PCE エレメントで指定されたチャンネル構成がデコード結果と不一致
	100	RSACPD_WARN_INVALID_DECODE_TYPE	設定されたデコードタイプが不正
	-1	RSACPD_ERR_DATA_EMPTY	入力データの終了を検出
	-2	RSACPD_ERR_NO_RAW_DATA_BLOCK	デコード可能な raw_data_block が存在しない

-3	RSACPD_ERR_PARAM	入力パラメータが不正
-10	RSACPD_ERR_NO_ADIF	ADIF ヘッダ取得時に ADIF ヘッダ ID ("ADIF")が検出できない
-60	RSACPD_ERR_ADTS_DATA	ADTS ヘッダ取得時、ADTS ヘッダフォーマット異常
-90	RSACPD_ERR_SAMPLE_INDEX	RSACPD_SetFormat()関数で無効なサンプリング周波数を指定
-100	RSACPD_ERR_NOT_READY	デコードに必要な情報が未指定
-110	RSACPD_ERR_STREAM_DATA	フォーマット異常を検出
-111	RSACPD_ERR_SFB_TBL	SFB(スケールファクタ)テーブルデータ異常
-112	RSACPD_ERR_HUFFMAN	ハフマンコードが異常
-200	RSACPD_ERR_AAC_LC	AAC LC プロファイル以外の未サポート機能検出
-201	RSACPD_ERR_NOT_SUPPORT	ADTS ヘッダ取得時、AAC 以外のフォーマット検出
-202	RSACPD_ERR_CRC_CHECK	ADTSCRC エラーを検出
-212	RSACPD_ERR_FI_GAIN	未サポート機能(ゲインコントロール)検出
-213	RSACPD_ERR_FI_PREDI	未サポート機能(Prediction)検出
-220	RSACPD_ERR_SEQUENCE	API 関数の実行順序が不適切
-403	RSACPD_ERR_AUDIO_MODE	未サポートの音声モード
-404	RSACPD_ERR_PCECNT	利用可能な PCE 情報の数の設定値が不正
-405	RSACPD_ERR_PCE_LOC	PCE が他のエレメントの後に検出
-410	RSACPD_ERR_DRC_THREAD	不正な DRC データを検出
-501	RSACPD_ERR_DISAGREE_OUTPUT_FS	直前フレームのサンプリング周波数が現在のフレームのサンプリング周波数と異なる
-510	RSACPD_ERR_MIXDOWN_PARAM	ダウンミックスのパラメータが不正
-511	RSACPD_ERR_MIXDOWN_CH	ダウンミックスできないチャンネル数
-512	RSACPD_ERR_DSECNT	取得可能な DSE 情報の数の設定値が不正
-700	RSACPD_ERR_LOAS_INFO	異常な LOAS ヘッダまたは、未サポートの LOAS ヘッダを検出

8.2 API 関数とステータスコード

8.2.1 RSACPD_Open

表 8.2 RSACPD_Open()関数のステータスコード

分類	コード名	出力	説明	アプリケーションプログラムの 処理
		RSACPD_AAC 構造体		
正常	RSACPD_RTN_GOOD	確定	RSACPD_AAC 構造体、 ワーク領域の初期化処理 が正常終了しました。	デコードシーケンスに従い、次の 関数を実行して下さい。
異常	RSACPD_ERR_PARAM	不定	引数に NULL ポインタが 渡された、または、入力 バッファ長に 0 以下が指 定されました。	引数を正しく設定して再実行し て下さい。

8.2.2 RSACPD_SetPCEArea

表 8.3 RSACPD_SetPCEArea()関数のステータスコード

分類	コード名	出力		説明	アプリケーション プログラムの処理
		RSACPD_AAC 構造体	RSACPD_PCE 構造体		
正常	RSACPD_RTN_GOOD	確定	不定 (注 1)	RSACPD_PCE 構造体の領域が 正しく設定され ました。	デコードシーケ ンスに従い、次の関 数を実行して下さい。
異常	RSACPD_ERR_PARAM	不定	不定	引数に NULL ポイ ンタが渡されま した。	引数を正しく設定 して再実行して下 さい。
	RSACPD_ERR_SEQUENCE	不定	不定	RSACPD_Open() 関数が実行され ていません。	RSACPD_Open() 関数を実行して下 さい。
	RSACPD_ERR_PCECNT	不定	不定	利用可能な PCE 情報の数が不正 な値です。	利用可能な PCE 情 報の数は 1~16 で す。正しい値の設 定と領域の確保を 行って下さい。

(注 1) 領域のみ確保しました。内容は PCE エレメントをデコードするまで不定です。

8.2.3 RSACPD_GetAdifHeader

表 8.4 RSACPD_GetAdifHeader()関数のステータスコード

分類	コード名	出力			説明	アプリケーション プログラムの処理
		RSACPD_AAC 構造体	RSACPD_Adif Header 構造体	bcnt		
正常	RSACPD_RTN_GOOD	確定	確定	ADIF ヘッダ のバイト数	ADIF ヘッダ情報が正常に構造体に格納されました。	デコードシーケンスに従い、次の関数を実行して下さい。
異常	RSACPD_ERR_DATA_EMPTY	不定	不定	0	入力データの終了を検出しました。	なし。
	RSACPD_ERR_PARAMETER	不定	不定	不定	引数に NULL ポインタが渡されました。	引数を正しく設定して再実行して下さい。
	RSACPD_ERR_NO_ADIF	不定	不定	0	ADIF ヘッダ ID ("ADIF")が検出されませんでした。	ビットストリームのデータフォーマットを確認して下さい。
	RSACPD_ERR_STREAM_DATA	不定	不定	0	ビットストリームのフォーマットが異常です。	異常ビットストリームのためデコードできません。引き続きデコードする場合は、ビットストリームを確認し、RSACPD_Open()関数を実行して下さい。
	RSACPD_ERR_SEQUENCE	不定	不定	不定	RSACPD_Open()関数が正しく実行されていません。	RSACPD_Open()関数を実行して下さい。
	RSACPD_ERR_AUDIO_MODE	不定	不定	不定	未サポートの音声モードです。	未サポートビットストリームのため、デコード出来ません。引き続きデコードする場合は、ビットストリームを確認し、RSACPD_Open()関数を実行して下さい。
	RSACPD_ERR_PCECNT	不定	不定	不定	RSACPD_SetPCEArea()関数が実行されていない可能性があります。または、RSACPD_SetPCEArea()関数で設定した利用可能なPCE情報の数以上のPCEを検出しました。	RSACPD_SetPCEArea()関数を実行し、利用可能なPCE情報の数を正しく設定して下さい。

8.2.4 RSACPD_GetAdtsHeader

表 8.5 RSACPD_GetAdtsHeader()関数のステータスコード

分類	コード名	出力			説明	アプリケーション プログラムの処理
		RSACPD_AAC 構造体	RSACPD_Adts Header 構造体	bcnt		
正常	RSACPD_RTN_GO OD	確定	確定	ADTS ヘッダの バイト数	ADTSヘッダ情報が正 常に構造体に格納さ れました。	デコードシーケンスに 従い、次の関数を実行し て下さい。
ワー ーニング	RSACPD_WARN_F I_GARBAGE	確定	確定	ADTS ヘッダのバ イト数 + ごみデータ のサイズ	同期語の前に不要(ご み)データを検出しま した。	デコードシーケンスに 従い、次の関数を実行し て下さい。
異常	RSACPD_ERR_DA TA_EMPTY	不定	不定	0	入力データの終了を 検出しました。	なし。
	RSACPD_ERR_PA RAM	不定	不定	不定	引数に NULL ポイン タが渡されました。	引数を正しく設定して 再実行して下さい。
	RSACPD_ERR_AD TS_DATA	不定	不定	直前まで の読み ヘッダ バイト数	ADTSヘッダのメンバ sampling_frequency_i ndexの値が不正です。 (0xb より大きい)	未サポートビットスト リームのため、デコード 出来ません。
	RSACPD_ERR_ST REAM_DATA	不定	不定	0	ビットストリームの フォーマットが異常 です。	異常ビットストリーム のためデコードできま せん。引き続きデコード する場合は、ビットスト リームを確認し、 RSACPD_Open()関数 を実行して下さい。
	RSACPD_ERR_AA C_LC	不定	不定	直前まで の読み ヘッダ バイト数	ADTSヘッダのメンバ profileの値が不正で す。 (AAC LC ではない)	未サポートビットスト リームのため、デコード 出来ません。
	RSACPD_ERR_NO T_SUPPORT	不定	不定	直前まで の読み ヘッダ バイト数	ADTSヘッダのメンバ layerの値が不正です。 (0x00 ではない)	未サポートビットスト リームのため、デコード 出来ません。
	RSACPD_ERR_CR C_CHECK	不定	不定	0	CRC エラーを検出し ました。	異常ビットストリーム のため、デコード出来 ません。
	RSACPD_ERR_SE QUENCE	不定	不定	不定	RSACPD_Open()関 数が正しく実行され ていません。	RSACPD_Open()関数 を実行して下さい

8.2.5 RSACPD_GetLoasInfo

表 8.6 RSACPD_GetLoasInfo()関数のステータスコード

分類	ステータスコード名	出力			説明	アプリケーション プログラムの処理
		RSACPD_AAC 構造体	RSACPD_LoasInfo 構造体	bcnt		
正常	RSACPD_RTN_GOOD	確定	確定	LOAS ヘッダの バイト数	LOAS ヘッダ情報が正常に構造体に格納されました。	デコードシーケンスに従い、次の関数を実行してください。
ワーニング	RSACPD_WARN_FI_GARBAGE	確定	確定	LOAS ヘッダの バイト数 +ごみ データの サイズ	同期語の前にごみデータを検出しました。	デコードシーケンスに従い、次の関数を実行してください。
異常	RSACPD_ERR_DATA_EMPTY	不定	不定	0	入力データの終了を検出しました。	RSACPD_Close()関数を実行し、終了処理を行ってください。
	RSACPD_ERR_PARAM	不定	不定	不定	引数に NULL ポインタが渡されました。	引数を正しく設定して再実行して下さい。
	RSACPD_ERR_LOAS_INFO	不定	不定	0	異常、または未サポートの LOAS ヘッダです。	異常、または未サポートビットストリームのため、デコードできません。
	RSACPD_ERR_STREAM_DATA	不定	不定	0	ビットストリームのフォーマットが異常です。	異常ビットストリームのためデコードできません。引き続きデコードする場合は、ビットストリームを確認し、RSACPD_Open()関数を実行してください。
	RSACPD_ERR_AAC_LC	不定	不定	0	PCE エlementのプロファイルが不正です。(AAC LC ではない)	未サポートビットストリームのため、デコードできません。
	RSACPD_ERR_AUDIO_MODE	不定	不定	0	未サポートの音声モードです。	異常ビットストリームのためデコードできません。引き続きデコードする場合は、ビットストリームを確認し、RSACPD_Open()関数を実行してください。
	RSACPD_ERR_SEQUENCE	不定	不定	不定	RSACPD_Open()関数が正しく実行されていません。	RSACPD_Open()関数を実行してください。

8.2.6 RSACPD_SetFormat

表 8.7 RSACPD_SetFormat()関数のステータスコード

分類	コード名	出力	説明	アプリケーション プログラムの処理
		RSACPD_AAC 構造体		
正常	RSACPD_RTN_GOOD	確定	RawDataStream 形式のビットストリームをデコードするための情報が正しく設定されました。	デコードシーケンスに従い、次の関数を実行して下さい。
ワーニング	RSACPD_WARN_INVALID_DECODE_TYPE	確定	不正なデコードタイプが指定されました。	デコードシーケンスに従い、次の関数を実行して下さい。デコードタイプに 0 が設定されたとしデコードを継続します。
異常	RSACPD_ERR_SAMPLE_INDEX	不定	サポート外のサンプリングレートインデックスが指定されました。	「表 3.3 サンプリング周波数一覧(Sampling_frequency_index)」を参照して正しいサンプリングレートインデックスを指定して下さい。
	RSACPD_ERR_SEQUENCE	不定	RSACPD_Open()関数が正しく実行されていません。	RSACPD_Open()関数を実行して下さい。本関数は、RSACPD_Decode()関数、RSACPD_Skip()関数実行前に実行する必要があります。

8.2.7 RSACPD_Decode

表 8.8 RSACPD_Decode()関数のステータスコード

分類	コード名	出力			説明	アプリケーション プログラムの処理
		RSACPD_AAC / RSACPD_OUT_INFO structure	bcnt	pnum		
正常	RSACPD_RTN_GOOD	確定	1 ブロックのバイト数	1024/2048 or 960/1920	正常にデコード処理を完了しました。	デコードシーケンスに従い、次の関数を実行して下さい。
ワーニング	RSACPD_WARN_PCE	確定	1 ブロックのバイト数	0	PCE のみを含むブロックのデコードを完了しました。	PCE のみを含むブロックは規格外です。引き続きデコードする場合は、ビットストリームを確認して下さい。(注 1)

RSACPD_WARN_NO_AUDIO_DATA	確定	1 ブロックのバイト数	0	ブロック内にオーディオデータが存在しません。	規格外のビットストリームです。引き続きデコードする場合は、ビットストリームを確認して下さい。(注 1)
RSACPD_WARN_ERR_ADTS_LEN	確定	1 ブロックのバイト数	1024/2048 or 960/1920	RSACPD_AdtsHeader 型構造体メンバ frame_length と実際に読み取った ADTS フレーム長が異なります。	デコード処理は正常に終了していますが、frame_length が異常なため出力 PCM データが不正な可能性があります。
RSACPD_WARN_UNSUPPORTED_CH_CFG	確定	1 ブロックのバイト数	1024/2048 or 960/1920	未サポートのチャンネル構成を検出しました。	未サポートのチャンネル構成です。詳細は「5.1チャンネル数定義」を参照して下さい。
RSACPD_WARN_FI_MULTI_CCE	確定	1 ブロックのバイト数	1024/2048 or 960/1920	複数の CCE を検出したため、最初に出現した CCE のみカップリングデコードしました。	カップリングチャンネルは 1 チャンネルのみサポートしています。
RSACPD_WARN_FI_MIX_AAC	確定	1 ブロックのバイト数	1024/2048 or 960/1920	入力ビットストリームが切り替わりしました。 (AAC <-> aacPlus)	デコードシーケンスに従い、次の関数を実行して下さい。
RSACPD_WARN_DISAGREE_SBR_DATA	確定	1 ブロックのバイト数	1024/2048 or 960/1920	SCE/CPE の数と SBR_DATA の数が一致していません。	SBR_DATA が充足時は、通常にデコード、 SBR_DATA が不足時は、アップサンプリング処理された PCM データが出力されます。 引き続きデコードを続行する場合は、デコードシーケンスに従い、次の関数を実行して下さい。
RSACPD_WARN_ERR_SBR_CRC_CHECK	確定	1 ブロックのバイト数	1024/2048 or 960/1920	SBR-CRC エラーを検出しました。	アップサンプリング処理された PCM データが出力されます。 引き続きデコードを続行する場合は、デコードシーケンスに従い、次の関数を実行して下さい。
RSACPD_WARN_AUTO_DS_SBR	確定	1 ブロックのバイト数	1024 or 960	自動ダウンサンプリングを実施しました。	デコードシーケンスに従い、次の関数を実行して下さい。

	RSACPD_WARN_NOT_SUPPORT_SBR_FS	確定	1 ブロックのバイト数	1024 or 960	未サポートの SBR のサンプリング周波数です。	SBR デコードは行わず、AAC のみの PCM データが出力されます。 引き続きデコードを続行する場合は、デコードシーケンスに従い、次の関数を実行して下さい。
	RSACPD_WARN_LFE_RESTRICT_ERR	確定	1 ブロックのバイト数	1024/2048 or 960/1920	LFE の制約違反を検出しました。	出力 PCM が不正な可能性があります。 引き続きデコードを続行する場合は、デコードシーケンスに従い、次の関数を実行して下さい。
	RSACPD_WARN_SBR_HEADER_ERR	確定	1 ブロックのバイト数	1024/2048 or 960/1920	SBR ヘッダエラーを検出しました。	アップサンプリング処理された PCM データが出力されます。 引き続きデコードを続行する場合は、デコードシーケンスに従い、次の関数を実行して下さい。
	RSACPD_WARN_DISAGREE_INPUT_FS	確定	1 ブロックのバイト数	1024/2048 or 960/1920	入力サンプリング周波数の変化を検出しました。	引き続きデコードを続行する場合は、デコードシーケンスに従い、次の関数を実行して下さい。
	RSACPD_WARN_ILLEGAL_CHAN_CONFIG	確定	1 ブロックのバイト数	1024/2048 or 960/1920	ADTS ヘッダの channel_configuration と実際のチャンネル構成が一致しません。	未サポートのチャンネル構成です。詳細は「5.1チャンネル数定義」を参照して下さい。
異常	RSACPD_ERR_DATA_EMPTY	不定	0 or 直前までの読込バイト数	0	入力データの終了を検出しました。	なし。
	RSACPD_ERR_NO_RAW_DATA_BLOCK	不定	0	0	フレーム内にデコード可能な raw_data_block が存在しません。	なし。
	RSACPD_ERR_PARAMETER	不定	不定	不定	引数に NULL ポインタが渡されました。	引数を正しく設定して再実行して下さい。
	RSACPD_ERR_NOT_READY	不定	直前までの読込バイト数	0	デコードに必要な情報が指定されていません。	RawDataStream 形式の場合は、RSACPD_SetFormat()関数によりデコードに必要な情報を与えて下さい。
	RSACPD_ERR_STREAM_DATA	不定	直前までの読込バイト数	0	ビットストリームのフォーマットが異常です。	異常ビットストリームのため、デコードできません。

RSACPD_ERR_SFB_TBL	不定	直前までの読込バイト数	0	スケールファクタ値が異常です。	異常ビットストリームのため、デコードできません。
RSACPD_ERR_HUFFMAN	不定	直前までの読込バイト数	0	ハフマンコードエラーを検出しました。	異常ビットストリームのため、デコードできません。
RSACPD_ERR_AAC_LC	不定	直前までの読込バイト数	0	PCE エLEMENTのプロファイルが不正です。(AAC LC ではない)	異常ビットストリームのため、デコードできません。
RSACPD_ERR_CRC_CHECK	不定	直前までの読込バイト数	0	CRC エラーを検出しました。	異常ビットストリームのため、デコード出来ません。
RSACPD_ERR_FI_GAIN	不定	直前までの読込バイト数	0	未サポート機能(ゲインコントロール)を検出しました。	未サポートビットストリームのため、デコード出来ません。ゲインコントロール機能は、AAC LC プロファイルではサポート外です。
RSACPD_ERR_FI_PREDI	不定	直前までの読込バイト数	0	未サポート機能(PREDICTION)を検出しました。	未サポートビットストリームのためデコード出来ません。PREDICTION 機能は、AAC LC プロファイルではサポート外です。
RSACPD_ERR_SEQUENCE	不定	不定	不定	RSACPD_Open()関数が正しく実行されていません。	RSACPD_Open()関数を実行して下さい。
RSACPD_ERR_AUDIO_MODE	不定	0 or 直前までの読込バイト数	0	未サポートの音声モードです。	未サポートビットストリームのため、デコード出来ません。
RSACPD_ERR_PCECNT	不定	0 or 直前までの読込バイト数	0	PCE エLEMENTの数が不正です。(16 より大きい)	異常ビットストリームのため、デコードできません。
RSACPD_ERR_PCELOC	不定	直前までの読込バイト数	0	PCE を他のELEMENTの後に検出しました。	異常ビットストリームのため、デコードできません。
RSACPD_ERR_DRC_THREAD	不定	直前までの読込バイト数	0	不正な DRC スレッド情報を検出しました。	異常ビットストリームのため、デコードできません。

RSACPD_ERR_DISAGREE_OUTPUT_FS	不定	1 ブロックのバイト数	0	直前フレームの出力サンプリング周波数が現在のフレームの出力サンプリング周波数と異なります。	異常ビットストリームのため、デコードできません。
-------------------------------	----	-------------	---	---	--------------------------

(注 1) ビットストリームは規格外ですが、続くフレームのビットストリームが正常であれば継続デコード可能です。

8.2.8 RSACPD_Skip

表 8.9 RSACPD_Skip()関数のステータスコード

分類	コード名	出力		説明	アプリケーションプログラムの処理
		RSACPD_AAC 構造体	bcnt		
正常	RSACPD_RTN_GOOD	確定	1 ブロックのバイト数	正常にスキップ処理を完了しました。	デコードシーケンスに従い、次の関数を実行して下さい。
ワーニング	RSACPD_WARN_PCE	確定	1 ブロックのバイト数	PCE のみを含むブロックのデコードを完了しました。	PCE のみを含むブロックは規格外です。引き続きデコードする場合は、ビットストリームを確認して下さい。(注 1)
	RSACPD_WARN_NO_AUDIO_DATA	確定	1 ブロックのバイト数	ブロック内にオーディオデータが存在しません。	規格外のビットストリームです。引き続きデコードする場合は、ビットストリームを確認して下さい。(注 1)
	RSACPD_WARN_ERR_ADTS_LEN	確定	1 ブロックのバイト数	RSACPD_AdtsHeader 型構造体メンバー frame_length と実際に読み取った ADTS フレーム長が異なります。	デコード処理は正常に終了していますが、frame_length が異常なため出力 PCM データが不正な可能性があります。
	RSACPD_WARN_UNSUPPORTED_CH_CFG	確定	1 ブロックのバイト数	未サポートのチャンネル構成を検出しました。	未サポートのチャンネル構成です。詳細は「5.1 チャンネル数定義」を参照して下さい。
	RSACPD_WARN_FI_MULTIPLE_CCE	確定	1 ブロックのバイト数	複数の CCE を検出したため、最初に出現した CCE のみカップリングデコードしました。	カップリングチャンネルは 1 チャンネルのみサポートしています。

	RSACPD_WARN_ILLEGAL_CHAN_CONFIG	確定	1 ブロックのバイト数	ADTS ヘッダの channel_configuration と実際のチャンネル構成が一致しません。	未サポートのチャンネル構成です。詳細は「5.1 チャンネル数定義」を参照して下さい。
異常	RSACPD_ERR_DATA_EMPTY	不定	0 or 直前までの読込バイト数	入力データの終了を検出しました。	なし。
	RSACPD_ERR_NO_RAW_DATA_BLOCK	不定	0	フレーム内にデコード可能な raw_data_block が存在しません。	なし。
	RSACPD_ERR_PARAM	不定	不定	引数に NULL ポインタが渡されました。	引数を正しく設定して再実行して下さい。
	RSACPD_ERR_NOT_READY	不定	直前までの読込バイト数	デコードに必要な情報が指定されていません。	RawDataStream 形式の場合は、RSACPD_SetFormat() 関数によりデコードに必要な情報を与えて下さい。
	RSACPD_ERR_STREAM_DATA	不定	直前までの読込バイト数	ビットストリームのフォーマットが異常です。	異常ビットストリームのため、デコードできません。
	RSACPD_ERR_SFB_TBL	不定	直前までの読込バイト数	スケールファクタ値が異常です。	異常ビットストリームのため、デコードできません。
	RSACPD_ERR_HUFFMAN	不定	直前までの読込バイト数	ハフマンデコードエラーを検出しました。	異常ビットストリームのため、デコードできません。
	RSACPD_ERR_AAC_LC	不定	直前までの読込バイト数	PCE エLEMENTのプロファイルが不正です。(AAC LC ではない)	異常ビットストリームのため、デコードできません。
	RSACPD_ERR_CRC_CHECK	不定	直前までの読込バイト数	CRC エラーを検出しました。	異常ビットストリームのため、デコード出来ません。
	RSACPD_ERR_FI_GAIN	不定	直前までの読込バイト数	未サポート機能(ゲインコントロール)を検出しました。	未サポートビットストリームのため、デコード出来ません。ゲインコントロール機能は、AAC LC プロファイルではサポート外です。
	RSACPD_ERR_FI_PREDI	不定	直前までの読込バイト数	未サポート機能(PREDICTION)を検出しました。	未サポートビットストリームのためデコード出来ません。PREDICTION 機能は、AAC LC プロファイルではサポート外です。

RSACPD_ERR_SEQUENCE	不定	不定	RSACPD_Open()関数が正しく実行されていません。	RSACPD_Open()関数を実行して下さい。
RSACPD_ERR_AUDIO_MODE	不定	0 or 直前までの読込バイト数	未サポートの音声モードです。	未サポートビットストリームのため、デコード出来ません。
RSACPD_ERR_PCECNT	不定	0 or 直前までの読込バイト数	PCE エLEMENTの数が不正です。 (16 より大きい)	異常ビットストリームのため、デコードできません。
RSACPD_ERR_PCE_LOC	不定	直前までの読込バイト数	PCE を他のエレメントの後に検出しました。	異常ビットストリームのため、デコードできません。
RSACPD_ERR_DRC_THREAD	不定	直前までの読込バイト数	不正な DRC スレッド情報を検出しました。	異常ビットストリームのため、デコードできません。

(注 1) ビットストリームは規格外ですが、続くフレームのビットストリームが正常であれば継続デコード可能です。

8.2.9 RSACPD_DecodeStatus

表 8.10 RSACPD_DecodeStatus()関数のステータスコード

分類	コード名	出力	説明	アプリケーション プログラムの処理
		ステータス *status		
正常	RSACPD_RTN_GOOD	確定	デコードステータスの確認が正常に実行されました。	デコードシーケンスに従い、次の関数を実行して下さい。
異常	RSACPD_ERR_PARAM	不定	引数に NULL ポインタが渡されました。	引数を正しく設定して再実行して下さい。
	RSACPD_ERR_SEQUENCE	不定	デコードシーケンスが異常です。	RSACPD_Open()関数を実行して下さい。 本関数は、 RSACPD_Decode()、 RSACPD_Skip() 関数実行後に実行してください。

8.2.10 RSACPD_SetDecOpt

表 8.11 RSACPD_SetDecOpt()関数のステータスコード

分類	コード名	デコード オプション	説明	アプリケーション プログラムの処理
正常	RSACPD_RTN_GOOD	確定	デコードオプションが 正しく設定されました。	デコードシーケンスに従い、 次の関数を実行して下さい。
ワー ニング	RSACPD_WARN_SEQUENCE	確定	未サポートの API 関数 の実行順序です。	デコードごとのデコードオ プションの切り替えは、ユー ザ責任となります。
異常	RSACPD_ERR_SEQUENCE	不定	RSACPD_Open()関数 が正しく実行されてい ません。	RSACPD_Open()関数を実 行して下さい。

8.2.11 RSACPD_SetDSE

表 8.12 RSACPD_SetDSE()関数のステータスコード

分類	コード名	出力		説明	アプリケーション プログラムの処理
		RSACPD_ AAC 構造体	RSACPD_ DSE 構造		
正常	RSACPD_RTN_GOOD	確定	不定 (注 1)	RSACPD_DSE 構造 体の領域が正しく設 定されました。	デコードシーケンスに 従い、次の関数を実行 して下さい。
異常	RSACPD_ERR_SEQUENCE	不定	不定	RSACPD_Open()関 数が、正しく実行さ れていません。	RSACPD_Open()関数 を実行して下さい。
	RSACPD_ERR_DSECNT	不定	不定	取得可能な DSE の 数が不正な値です。	利用可能な DSE 情報 の数は 1~16 です。正 しい値の設定と領域の 確保を行って下さい。
	RSACPD_ERR_PARAM	不定	不定	引数に NULL ポイン タが渡されました。	引数を正しく設定して 再実行して下さい。

(注 1) 領域のみ確保しました。内容は DSE エlement をデコードするまで不定です。

8.2.12 RSACPD_MatrixMixdown

表 8.13 RSACPD_MatrisMixdown()関数のステータスコード

分類	コード名	出力		説明	アプリケーション プログラムの処理
		RSACPD_AAC 構造体/ RSACPD_OUT_ INFO_structure	mixdown_ mode		

正常	RSACPD_RTN_GOOD	確定	確定	ダウンミックス処理が正常に完了しました。	デコードシーケンスに従い、次の関数を実行して下さい。
ワーニング	RSACPD_WARN_MIXDOWN_OVF	確定	確定	ダウンミックス後の出力がオーバーフローしました。	ダウンミックス時の準拠規格、モード設定（適用する規格の変更）を確認してください。
異常	RSACPD_ERR_SEQUENCE	不定	不定	API 関数の実行順序が不正です。	RSACPD_Decode() 関数の実行後、本関数を実行して下さい。
	RSACPD_ERR_MIXDOWN_PARAM	不定	不定	パラメータの値の設定が不正です。	正しいパラメータの値を設定して下さい。
	RSACPD_ERR_PARAM	不定	不定	引数に NULL ポインタが渡されました。	引数を正しく設定して再実行して下さい。
	RSACPD_ERR_MIXDOWN_CH	不定	不定	ダウンミックス対象外のチャンネル数です。	3/2、3/2+LFE および 3/1、2/2、3/0、2/1 以外のチャンネル構成に対するダウンミックス処理には対応していません。

8.2.13 RSACPD_SetSAC

表 8.14 RSACPD_SetSAC()関数のステータスコード

分類	コード名	出力		説明	アプリケーションプログラムの処理
		RSACPD_AAC 構造体	RSACPD_SAC 構造体		
正常	RSACPD_RTN_GOOD	確定	不定 (注 1)	RSACPD_SAC 構造体の領域が正しく設定されました。	デコードシーケンスに従い、次の関数を実行して下さい。
異常	RSACPD_ERR_SEQUENCE	不定	不定	RSACPD_Open()関数が正しく実行されていません。	RSACPD_Open()関数を実行して下さい。
	RSACPD_ERR_PARAM	不定	不定	引数に NULL ポインタが渡されました。	引数を正しく設定して再実行して下さい。

(注 1) 領域のみ確保しました。内容は SAC エlement をデコードするまで不定です。

8.2.14 RSACPD_SetDRC

表 8.15 RSACPD_SetDRC()関数のステータスコード

分類	コード名	出力		説明	アプリケーション プログラムの処理
		RSACPD_ AAC 構造体	RSACPD_ DRC 構造		
正常	RSACPD_RTN_GOOD	確定	不定 (注 1)	RSACPD_DRC 構造体の領域が正しく設定されました。	デコードシーケンスに従い、次の関数を実行して下さい。
異常	RSACPD_ERR_SEQUENCE	不定	不定	RSACPD_Open()関数が正しく実行されていません。	RSACPD_Open()関数を実行して下さい。
	RSACPD_ERR_PARAM	不定	不定	引数に NULL ポインタが渡されました。	引数を正しく設定して再実行して下さい。

(注 1) 領域のみ確保しました。内容は DRC 情報をデコードするまで不定です。

9 組み込み手順

9.1 システム構成

本ミドルウェアのシステム構成例を図 9.1に示します。点線部分が本ミドルウェアに該当します。

ビットストリームのメモリへの読み込み、および PCM データのメモリからの読み出しはアプリケーションプログラムで行う必要があります。

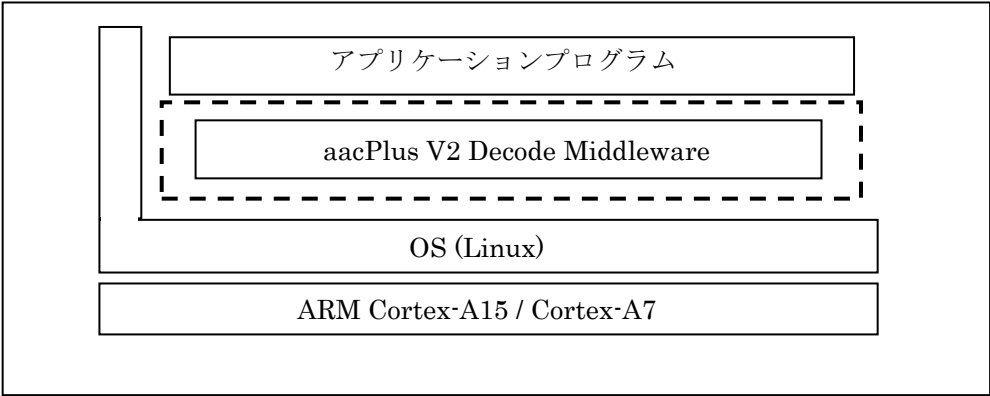


図 9.1 システム構成例

9.2 開発環境

本ミドルウェアは、表 9.1の開発環境を用いて開発しています。アプリケーションプログラムに組み込む際には、表 9.1の開発ツール、あるいは互換性のある開発環境を使用して下さい。

表 9.1 開発環境

項目	説明	備考
Target OS	Linux kernel release 3.10	関連マニュアル[1]参照
Board Support Package	Linux Interface Specification Yocto recipe	
ツールチェーン	arm-poky-linux-gnueabi-gcc (Linaro GCC 4.8-2014.04) 4.8.3 20140401 (prerelease)	

9.3 ミドルウェア構成

本ミドルウェアは、以下のファイルで構成されます。

表 9.2 ミドルウェア構成

	ファイル名	内容
1	libRSACPDLA_L.so.1.1	ダイナミックリンクライブラリ
2	RSACPD_ADL.h	ヘッダファイル
3	RTM0AC0000ADAAPMZ1SL32E-02.pdf	ユーザーズマニュアル (英文)
4	RTM0AC0000ADAAPMZ1SL32J-02.pdf	ユーザーズマニュアル (和文)
5	sample_main.c	サンプルプログラム(C 言語) (注)

(注) 本サンプルプログラムは、お客様の環境で動作を保証するものではありません。

9.4 ユーザアプリケーションプログラムの作成

本ミドルウェアの各 API 関数を呼び出すユーザアプリケーションプログラム、及びユーザ記述関数(「4 ユーザ記述関数」)を作成して下さい。

ユーザアプリケーションプログラムは、次のインクルードファイルをインクルードして下さい。

表 9.3 インクルードファイル

	インクルードファイル	内容
1	RSACPD_ADL.h	本ミドルウェアが提供するヘッダファイル
2	string.h	標準ライブラリヘッダファイル

9.5 コンパイルオプションの設定

本ミドルウェアビルド時に設定したオプションを表 9.4に示します(その他はデフォルトの設定)。本ミドルウェアをリンクする際は、表 9.4の内容を確認の上リンクしてください。

表 9.4 コンパイルオプション

	コンパイルオプション	設定値	内容
1	Optimization	-O2	最適化レベル 2 を指定する
2	Character signed	-fsigned-char	char 型変数を符号付 char とする

10 注意事項

本章では、本ミドルウェア使用時の注意事項を記載しています。各デコーダ API 関数に関する注意事項は、3.2節の API 関数の説明<注意事項>に記載してありますので、本章とあわせて参照してください。

10.1 予約語

本ミドルウェアでは、他のアプリケーションプログラムと区別するため、関数名、マクロ名の先頭に“RSACPD_”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、“RSACPD_”ではじまる関数・変数を使用しないで下さい。

10.2 エラー終了後の復帰処理

RSACPD_Decode0関数、RSACPD_Skip0関数がエラー終了した後や、異なるビットストリームをデコードする場合は、必ず RSACPD_Open0関数による初期化処理を行って下さい。

10.3 デコードミドルウェアの監視

本ミドルウェア組み込み時は、システムがハングアップしないよう本ミドルウェアのデコード処理時間をタイマなどで監視し、上位プログラムにタイムアウト処理を実装して下さい。

10.4 aacPlus/AAC 混在符号化ビットストリームのデコード処理

本ミドルウェアは、デコード中に符号化ビットストリーム種別の遷移(AAC->aacPlus)を検出した場合、内部処理をリセットします。このため、最大 1/2 フレームのミュート区間が発生する場合があります。

aacPlus/AAC 混在符号化ビットストリームデコード時、aacPlus フレームと AAC フレームでは、出力ワード数が異なります(AAC フレームは aacPlus フレームのハーフレートのため)。混在ビットストリームデコード時は、出力 PCM データのワード数を必ず確認して下さい(「3.2.7 RSACPD_Decode」を参照して下さい)。

10.5 SBR ヘッダ情報未取得時の動作

aacPlus 符号化ビットストリームデコード時、ビットストリームから SBR ヘッダ情報を取得するまでは、aacPlus デコード処理を実施せずアップサンプリング処理を実施します。従って、SBR ヘッダ取得前後で出力サンプリングレートは変化しません(aacPlus データを含まないビットストリームのデコード時は除きます)。

10.6 デコード途中のチャネル構成の変化について

10.6.1 出力チャネル数の変化

デコード途中で入力ビットストリームのチャネル数が変化した場合、デコードは続行可能ですが、出力 PCM データのチャネル数は該当フレームの入力ビットストリームの内容に準じます。したがって、モノラルからステレオに変化した場合などは、上位アプリケーションプログラムでチャネル数の変化を監視し、必要な対応を行って下さい。

パラメトリックステレオ(PS)データを含むビットストリームをデコード時、出力 PCM データのチャネル数は、SBR ヘッダや PS データの有無によりフレームごとに出力チャネル数が変化する可能性があります。上位アプリケーションプログラムで出力チャネル数を監視し必要な対応を行って下さい。

10.6.2 入力チャネル構成の変化

本ミドルウェアは、入力ビットストリームのチャネル構成の変化を検出した場合、内部処理をリセットします。このため、最大 1/2 フレームのミュート区間が発生する場合があります。

10.7 デコード途中でのサンプリング周波数の変化について

デコード途中で入力サンプリング周波数の変化や、ビットストリーム種別の変化(AAC/aacPlus V1/V2)、または、デコードオプションによって、出力サンプリング周波数に変化があった場合の動作は以下ようになります。

- ・ エラーコンシールオプション有効時

変化前のサンプリング周波数で、エラーコンシール処理を行った PCM を出力します。その後も、出力サンプリング周波数が正常デコード時と異なる場合は、エラーコンシール処理を行いますが、出力 PCM データは無音となります。

- ・ エラーコンシールオプション無効時

エラー終了します。

ただし、デコード動作の間に RSACPD_Skip0関数により 1 フレーム以上のスキップ処理を行い、スキップ処理の前後のデコード区間で、出力サンプリング周波数に変化があった場合は、エラーコンシールオプションに関係なくデコード処理を続行します。この場合は、上位アプリケーションプログラムで出力サンプリング周波数の変化を監視し、必要な対応をとって下さい。

付録

RSACPD_AAC 型構造体のデータ構造

Member name	Description
unsigned char *BsBuf	Pointer to the beginning of input buffer
unsigned char *BsBufIdx	Indicate the address to read current raw data block
unsigned long UseBitCount	The number of bit are used
unsigned long nNoUseBit	The number of remained bit in Bs4Byte
unsigned long Bs4Byte	Buffer to read the necessary information
unsigned long BsBufSize	Size of the read data in input buffer
RSACPD_PCE *pcebuf	Pointer to the PCE buffer
RSACPD_SAC *sacbuf	Pointer to the SAC buffer
RSACPD_DRC *drcbuf	Pointer to the DRC buffer
RSACPD_PCE *ppcebuf[RSACPD_MAX_ELE_TAG]	Pointer to each PCE element buffer
RASCPD_DSE *pdsebuf[RSACPD_MAX_ELE_TAG]	Pointer to each DSE element buffer
char AudioObjectType	The audio object type of the input bit stream
char sampling_frequency_index	The index of sampling frequency of the input bit stream
unsigned char id	ID of the current element
unsigned char common_window	The flag indicating two individual_channel_stream share a common ics_info or not
struct RSACPD_ChannelData1 *pCD1	Pointer to the channel data CD1
struct RSACPD_ChannelData2 *pCD2	Pointer to the channel data CD2
struct RSACPD_ChannelData1 *pCDCW1	Pointer to the saved channel data
struct RSACPD_CCE CCE	CCE working data
SPEC *spec	MDCT spectral working data
int sequence_number	Sequence number
int ahcod[MAX_CHANNEL_NUM][1024]	Help element: decoder working data
int x_quant[1024]	Help element: decoder working data
int is_position[120]	Help element: decoder working data
int *hcod	Pointer to the ahcod when in decoding process
int default_config	Default configuration
int implicit_ch_cfg	Flag indicating the implicit channel configure
int current_program	The current program
int adts_channel_config	The ADTS header channel configuration
int first_block	Help element: temporary working data
int max_pce_cnt	Max PCE count in the current bit stream
int element_skip[RSACPD_Chans]	Flag mark the element is skipped in decoding process
SPEC *channel_spec[MAX_CHANNEL_NUM]	Pointer to the spectral buffer of each channel
struct RSACPD_ChannelData1 CD1[MAX_CHANNEL_NUM]	Channel data information 1 of each channel
struct RSACPD_ChannelData2 CD2[MAX_CHANNEL_NUM];	Channel data information 2 of each channel
char ID	ADTS header ID data

char header_type	Header type
unsigned int (*UserFunc)(unsigned char *, int)	Function pointer to user defined read data function
int ApiSeqNum	API sequence number
long RandomSeed[8][64]	The random vector seed use for decoding process
long CurrentSeed	The current seed number
void *crc	Pointer to the CRC data (RSACPD_CRCDATA)
unsigned char CrcFlag	CRC control flag 0 : Disable CRC reading 1 : Enable CRC reading
unsigned short FileCRC	The bit stream CRC value, checking with CalcCRC
unsigned short CalcCRC	The calculated CRC value, checking with FileCRC
unsigned short frame_length	ADTS frame length
RSACPD_CRC RSACPD_CRCDATA	The CRC working data
int audioCountSCE	SCE counter
int audioCountCPE	CPE counter
int audioCountLFE	LFE counter
int audioCountCCE	CCE counter
int audioCountPCE	PCE counter
int audioCountDSE	DSE counter
int audioCountSAC	SAC counter
int audioCountDRC	DRC counter
int flag_AAC_reset	Flags indicating the reset step is required
int curr_AAC_DecodeMode	Current AAC decode mode
unsigned int all_spectraldata_bit	The number of spectral data bit is used
int get_pce_cnt	The PCE count number acquired from ADIF/LOAS header
int crc_check_flag	The CRC checking flag 0 : No CRC checking 1 : CRC checking
int frame_len_cnt	ADTS frame length counter
int sac_enable	SAC status flag 0 : SAC disable 1 : SAC enable
int max_dse_cnt	Max DSE element count
int drc_scale	DRC scale
int drc_enable	DRC enable flag 0 : DRC disable 1 : DRC enable
int SkipFlag	Flag indicating skip is effective
short *PcmBuf[MAX_CHANNEL_NUM]	Output PCM buffer
short ChannelNumber	The number channel of output data
int PcmLen	The number of PCM samples per channel
int ch_Index	Number of current channel index is processed
int next_ch_Index	Number of the next channel index will be processed
int DerrorCode	The decode error code
RSACPD_OUT_INFO *pOutInfo	Pointer to the RSACPD_OUT_INFO data
RSACPD_OUT_INFO prevOutInfo	The value of the previous RSACPD_OUT_INFO data

RSACPD_MC_Info	MC_Info	Multichannel information
RSACPD_MC_Info	prev_mc_info	Previous frame channel information
RSACPD_MC_Info	save_mc_info	Saved multichannel information
SPEC overLapBuffer[1024*MAX_CHANNEL_NUM]		Overlapping spectral buffer
unsigned long buf_len		Size of input buffer
char RSACPD_flag		SBR decoding flag
int channelMode		Channel mode value
REENTRANCY1 entrancy1		Reentrancy data
int dec_mode		Decoding mode
int flag_error_conceal		Flag indicating error conceal is executing
int sbr_behavior		SBR decoding behavior
int prev_SBR_DecodeMode		Previous SBR decode mode
int curr_SBR_DecodeMode		Current SBR decode mode
int prev_sbr_behavior		Previous SBR decoding behavior
int prev_PcmLen		Previous frame PCM output samples number
int curr_sbrDataNum		The number of current SBR data number
int prev_sbrDataNum		The number of previous SBR data number
int prev_FS		The previous sampling frequency
int curr_FS		The current sampling frequency
int cc_enable		Coupling channel enable flag
int decode_cce		The decoding CCE status
int indicate_pce_flag		Flag indicating PCE is defined
int prev_cc_enable		Previous coupling channel enable flag
int curr_pce_detected		Flag indicating PCE is detected
unsigned char bUpSample;		Upsample SBR flag 0 : Upsample AAC off 1 : Upsample AAC on
int bDownSample		Down-sampling SBR flag 0 : Down-sampling SBR off 1 : Down-sampling SBR on
int ForceSbrOff		Force AAC flag 0 : SBR process is off 1 : SBR process is on
unsigned char useHqSbr		The HQ-SBR usage data
SBRBITSTREAM streamSBR		SBR stream data
SBRDECODER RSACPD_sbrDecoderInfo		SBR decoding information
struct SBR_DECODER_INSTANCE SBR_DECODER_INSTANCE_DATA		SBR decoding working data
int sbr_header_err_detect		SBR header error detected flag
int sbr_crc_err_detect		SBR CRC error detected flag
int curr_ps_detected		PS data detected flag
int flag_force_SBR_Up		SBR force up-sampling flag
int force_SBR_Up_enable		SBR force up-sampling enable flag
RSACPD_LoasInfo prev_header		The previous LOAS frame header data
unsigned long UseBitCount_LOAS		The number of LOAS header data is used
unsigned short frame_size		The number output samples per frames (1024/960)
LONG32 sbr_drcFactorVector[6][26][32]		The DRC Factor vector using for calculation DRC

RSACPD_DRC_Bitstream 型構造体のデータ構造

Member name	Description
int excl_chn_present	One bit indicating that excluded channels are present
int excl_chn_mask[RSACPD_MAX_CHAN];	Boolean array indicating the audio channels of a program that are excluded from DRC processing using this DRC information
int num_bands;	The number of bands greater than one if there is multi-band DRC information
int band_top[RSACPD_MAX_DRC_BANDS];	Indicates top of i-th DRC band in units of 4 spectral lines.
int prog_ref_level_present;	One bit indicating that reference level is present
int prog_ref_level;	Reference level. A measure of long-term program audio level for all channels combined.
int drc_sgn[RSACPD_MAX_DRC_BANDS];	Dynamic range control sign information. One bit indicating the sign of drc_mag (0 if positive, 1 if negative)
int drc_mag[RSACPD_MAX_DRC_BANDS];	Dynamic range control magnitude information
int drc_interp_scheme;	Indicates which interpolation scheme is used for the DRC data in the SBR QMF domain

RSACPD_DRC_Info 型構造体のデータ構造

Member name	Description
int num_bands;	The number of bands greater than one if there is multi-band DRC information
int drc_interp_scheme;	Indicates which interpolation scheme is used for the DRC data in the SBR QMF domain
int band_top[RSACPD_MAX_DRC_BANDS];	Indicates top of i-th DRC band in units of 4 spectral lines.
int drc_sgn[RSACPD_MAX_DRC_BANDS];	Dynamic range control sign information. One bit indicating the sign of drc_mag (0 if positive, 1 if negative)
int drc_mag[RSACPD_MAX_DRC_BANDS];	Dynamic range control magnitude information

改訂記録	ARM 5.1ch aacPlus V2 Decode 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
1.00	2014.09.30	-	初版発行
1.01	2014.12.05	30	表 0.1 末尾のビットフィールドにおいて、PCE のビットが予約領域になっていた問題を修正。
		33	RSACPD_get_version 関数の戻り値を更新
		46	ADTS ヘッダの channel configuration と raw_data_block に含まれるチャネル数が異なる場合に返却するワーニングをエラーに修正。
		63-76	8.2 API 関数とステータスコードのステータスコードを修正。
		78	ミドルウェア構成のファイル名を更新。
		83	構造体のメンバに「prev_mc_info」を追加。

ARM 5.1ch aacPlus V2 Decode Middleware for Linux

RTM0AC0000ADAAPMZ1SL32C

ユーザーズマニュアル

発行年月日 2014 年 12 月 05 日 Rev.1.01

発行 ルネサス エレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部 1753



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 5.1ch aacPlus V2 Decode Middleware for Linux

RTM0AC0000ADAAPMZ1SL32C



ルネサス エレクトロニクス株式会社

RTM0AC0000ADAAPMZ1SL32J-02