



客户端音频编解码库

API 参考

| | |
|-------|------------|
| 文档版本 | 06 |
| 发布日期 | 2008-09-05 |
| BOM编码 | N/A |

秘密

版权所有 © 深圳市海思半导体有限公司

深圳市海思半导体有限公司为客户提供全方位的技术支持，用户可与就近的海思办事处联系，也可直接与公司总部联系。

深圳市海思半导体有限公司

地址：深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址：<http://www.hisilicon.com>

客户服务电话：0755-28788858

客户服务传真：0755-28357515

客户服务邮箱：support@hisilicon.com.

版权所有 © 深圳市海思半导体有限公司 2006-2008。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思，均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

秘密

版权所有 © 深圳市海思半导体有限公司



目 录

| | |
|---------------------------------------|-----|
| 前 言..... | 1 |
| 1 概述..... | 1-1 |
| 1.1 描述范围 | 1-1 |
| 1.2 接口格式 | 1-1 |
| 1.3 函数列表 | 1-1 |
| 1.4 函数描述方式 | 1-2 |
| 1.5 结构体描述方式 | 1-2 |
| 1.6 海思语音帧结构 | 1-3 |
| 2 API参考 | 2-1 |
| 2.1 海思语音编码器初始化 | 2-1 |
| 2.2 海思语音解码器初始化 | 2-2 |
| 2.3 海思语音编解码器 | 2-4 |
| 2.4 海思语音转码器初始化 | 2-7 |
| 2.5 海思语音转码器 | 2-8 |
| 3 其它信息..... | 3-1 |
| 3.1 数据类型 | 3-1 |
| 3.1.1 常数定义 | 3-1 |
| 3.1.2 hiVOICE_G711_STATE_S | 3-3 |
| 3.1.3 hiVOICE_ADPCM_STATE_S | 3-3 |
| 3.1.4 hiVOICE_G726_STATE_S | 3-4 |
| 3.1.5 hiVOICE_TRANSCODE_STATE_S | 3-4 |
| 3.2 错误码 | 3-4 |
| A 缩略语 | A-1 |



前 言

概述

本节介绍本文档的内容、对应的产品版本、适用的读者对象、行文表达约定、历史修订记录等。

产品版本

与本文档相对应的产品版本如下所示。

| 产品名称 | 产品版本 |
|---------------------|------|
| Hi3510 通信媒体处理器 | V100 |
| Hi3511 H.264 编解码处理器 | V100 |
| Hi3512 H.264 编解码处理器 | V100 |

读者对象

本参考适用于程序员初学者阅读，描述了快速了解基于媒体处理音频编解码库的开发。本参考适合下列人员阅读：

- 熟练使用 C 语言的程序员

内容简介

本参考首先概述了媒体处理客户端音频编解码库 API 参考种类及其间关联，然后分别详细介绍了各种参考信息。全书共分为 3 章。

| 章节 | 内容 |
|------|---|
| 1 概述 | 介绍媒体处理客户端音频编解码库 API 参考有哪几类参考信息、各类参考信息的作用和关联。 阅读本主题后，您将对媒体处理音频编解码库的参考信息有一个整体了解。 |






| 章节 | 内容 |
|----------|---|
| 2 API 参考 | 首先根据功能分类列出全部 API，然后详细介绍每一个 API 接口函数。 本主题供您查阅媒体处理音频编解码库的 API。 |
| 3 其他信息 | 介绍除上述参考项外的其他信息参考（如数据结构、错误码等）。 |
| A 缩略语 | 列出本文档中出现的缩略语。 |

约定

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

| 符号 | 说明 |
|--|--|
|  危险 | 以本标志开始的文本表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害。 |
|  警告 | 以本标志开始的文本表示有中度或低度潜在危险，如果不能避免，可能导致人员轻微或中等伤害。 |
|  注意 | 以本标志开始的文本表示有潜在风险，如果忽视这些文本，可能导致设备或器件损坏、数据丢失、设备性能降低或不可预知的结果。 |
|  窍门 | 以本标志开始的文本能帮助您解决某个问题或节省您的时间。 |
|  说明 | 以本标志开始的文本是正文的附加信息，是对正文的强调和补充。 |

通用格式约定

| 格式 | 说明 |
|----|---------------------------------|
| 宋体 | 正文采用宋体表示。 |
| 黑体 | 一级、二级、三级标题采用黑体。 |
| 楷体 | 警告、提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。 |



| 格式 | 说明 |
|-----------------------|--|
| “Terminal Display” 格式 | “Terminal Display” 格式表示屏幕输出信息。此外，屏幕输出信息中夹杂的用户从终端输入的信息采用加粗字体表示。 |

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

| 修订日期 | 版本 | 修订说明 |
|------------|----|--|
| 2008-09-05 | 06 | 增加本文档适用的 Hi3512 产品版本。 |
| 2008-04-21 | 05 | 增加本文档适用的 Hi3511 产品版本。 |
| 2007-12-18 | 04 | <ul style="list-style-type: none">修改“1.3 函数列表”，新增函数 HI_VOICE_TransCodeReset、HI_VOICE_TransCodeFrame。修改 HI_VOICE_EncReset 函数中的注意的内容。修改 HI_VOICE_DecReset 函数中的注意的内容。新增“2.4 海思语音转码器初始化”。新增“2.5 海思语音转码器”。修改“3.1.1 常数定义”，新增 DVS1.3.1.0 版本编解码器类型、转码器类型、2 个返回错误码定义。新增“3.1.5 VOICE_TransState_S”。修改“3.2 错误码”，新增错误码 HI_ERR_VOICE_TRANS_TYPE、HI_ERR_VOICE_TRANS_DEVICE。 |
| 2007-04-20 | 03 | <ul style="list-style-type: none">修改 HI_VOICE_EncodeFrame 函数中的注意的内容。修改“3.1.1 常数定义”中的代码。修改 3.1.2、3.1.3、3.1.4 定义中的代码。修改各头文件。增加 HI_VOICE_EncodeFrame 注意。修改 HI_VOICE_DecodeFrame 注意。 |



| 修订日期 | 版本 | 修订说明 |
|------------|----|--|
| 2006-12-29 | 02 | <ul style="list-style-type: none">• 修改 API 接口为 API 参考。• 修改 HI_VOICE_EncReset 函数的描述。• 删除 HI_VOICE_DecReset 函数前的一句 typedef unsigned char HI_U8 孤立代码。• 修改 HI_VOICE_EncodeFrame 函数的目的、描述，增加注意内容。• 修改 HI_VOICE_DecodeFrame 函数的描述、参数，增加注意内容。• 将数据类型和错误码合并在一章，为其他信息。• 修改错误代码为错误码。• 修改缩略语表为缩略语。 |
| 2006-05-16 | 01 | 第 1 次发布。 |



1 概述

1.1 描述范围

本音频解码库支持 8kHz 采样率 G.711、ADPCM、G.726 编解码。其中，G.711 编解码器提供 A 律与 μ 律压缩编码，采用非线性量化把 128kbit/s 采样数据压缩为 64kbit/s 数据。G.711 标准适用于综合业务网和大多数数字电话链路。北美与日本通常采用 μ 律编码，欧洲和其他地区大都采用 A 律编码。ADPCM（Adaptive Differential Pulse Code Modulation）提供自适应差分脉冲编码调制编码，把 128kbit/s 线性数据压缩为 32kbit/s 数据。G.726 编解码器把 128kbit/s 线性数据压缩为 16kbit/s、24kbit/s、32kbit/s、40kbit/s 数据。

1.2 接口格式

无。

1.3 函数列表

| 函数 | 功能 | 页码 |
|---|--------------------------|---------------------|
| HI_VOICE_EncReset | 编码器初始化。 | 2-1 |
| HI_VOICE_DecReset | 解码器初始化。 | 2-2 |
| HI_VOICE_EncodeFrame | 编码器，负责对语音的压缩编码，一次编码一帧数据。 | 2-4 |
| HI_VOICE_DecodeFrame | 解码器，负责对压缩语音的解码，一次解码一帧数据。 | 2-5 |
| HI_VOICE_TransCodeReset | 转码器初始化。 | 2-7 |



| 函数 | 功能 | 页码 |
|---|---|---------------------|
| HI_VOICE_TransCodeFrame | 转码器（仅适用于 Hi3510_VSSDK_V1.3.1.0 版本）。负责将 Hi3510_VSSDK_V1.3.1.0 版本的 G711A、G711U、ADPCM_DVI4、G726_16KBPS 编码码流转为之后版本的编码码流。一次转换一帧数据。 | 2-8 |

1.4 函数描述方式

本章用 10 个域对 API 参考信息进行描述。

| 参数域 | 作用 |
|------|--------------------------------|
| 目的 | 简要描述 API 的主要功能。 |
| 语法 | 列出 API 的语法样式。 |
| 描述 | 简要描述 API 的工作过程。 |
| 参数 | 列出 API 的参数、参数说明及参数属性。 |
| 返回值 | 列出 API 的返回值及返回值说明。 |
| 错误码 | 列出 API 的错误码及错误码说明。 |
| 需求 | 列出 API 包含的头文件和 API 编译时要链接的库文件。 |
| 注意 | 使用 API 时应注意的事项。 |
| 举例 | 使用 API 的实例。 |
| 相关主题 | API 的其他相关信息。 |

1.5 结构体描述方式

| 参数域 | 作用 |
|------|----------------|
| 说明 | 简要描述结构体所实现的功能。 |
| 定义 | 列出结构体的定义。 |
| 注意事项 | 列出结构体的注意事项。 |



1.6 海思语音帧结构

| 参数位置(单位: HI_S16) | 参数比特位说明 | 参数含义 |
|------------------|---------|----------------------------------|
| 0 | [15:8] | 数据帧类型标志位。 01: 语音帧; 其他: 保留。 |
| | [7:0] | 保留。 |
| 1 | [15:8] | 帧循环计数器: 0~255。 |
| | [7:0] | 数据净荷长度(单位: HI_S16)。 |
| 2 | [15:0] | 净荷数据。 |
| 3 | [15:0] | 净荷数据。 |
| | [15:0] | 净荷数据。 |
| 2+n-1 | [15:0] | 净荷数据。 |
| 2+n | [15:0] | 净荷数据。 |

注: n 的取值范围请参见“2.3 海思语音编解码器”中“HI_VOICE_EncodeFrame”的描述。



2 API 参考

2.1 海思语音编码器初始化

HI_VOICE_EncReset

【目的】

开始编码任务时，初始化编码设备。

【语法】

```
#include "hi_voice_api.h"
HI_VOICE_API HI_RESULT HI_VOICE_EncReset(HI_VOID *pEncState,
HI_S16 nCoder);
```

【描述】

根据上层传入的编码设备指针，初始化编码设备。之后可以进入编码流程。

【参数】

| 参数名称 | 描述 | 输入/输出 | 全局/局部 |
|-----------|---|-------|-------|
| pEncState | 用户指定的编码设备，设备定义请参见“3.1.2 hiVOICE_G711_STATE_S”、“3.1.3 hiVOICE_ADPCM_STATE_S”、“3.1.4 hiVOICE_G726_STATE_S”。 | 输入/输出 | 全局 |
| nCoder | 用户指定的编码设备类型，类型定义请参见“3.1.1 常数定义”。 | 输入 | 局部 |

【返回值】

| 返回值 | 描述 |
|-----|------------|
| 0 | 成功。 |
| 非 0 | 失败，其值为错误码。 |



【错误码】

| 错误码 | 描述 |
|---|--------------|
| HI_ERR_VOICE_ENC_TYPE | 编码器类型非法。 |
| HI_ERR_VOICE_INVALID_DEVICE | 输入编解码设备指针非法。 |

【需求】

头文件：/include/hi_voice_api.h

【注意】

- G711_ORG_A、G711_ORG_U、ADPCM_ORG_DV14、G726_ORG_16KBPS 四种编码设备类型仅针对 Hi3510_VSSDK_V1.3.1.0 版本编码，之后版本不推荐使用。

【举例】

```
HI_VOID VoiceDemo()  
{  
    HI_RESULT Ret;  
    hiVOICE_G711_STATE_S g_sG711EncState;  
    Ret =HI_VOICE_EncReset (&g_sG711EncState, G711_A);  
    if (Ret != 0)  
        MyErrorExit("HISI_VOICE_enc_reset ");  
    else  
        printf("HISI_VOICE_enc_reset SUCCESS.\n");  
}
```

【相关主题】

- [HI_VOICE_EncodeFrame](#)

2.2 海思语音解码器初始化

HI_VOICE_DecReset

【目的】

开始解码任务时，初始化解码设备。

【语法】

```
#include "hi_voice_api.h"  
HI_VOICE_API HI_RESULT HI_VOICE_DecReset(HI_VOID *pDecState,  
HI_S16 nCoder);
```

【描述】



初始化解码设备。

【参数】

| 参数名称 | 描述 | 输入/输出 | 全局/局部 |
|-----------|---|-------|-------|
| pDecState | 用户指定的解码设备，设备定义请参见“3.1.2 hiVOICE_G711_STATE_S”、“3.1.3 hiVOICE_ADPCM_STATE_S”、“3.1.4 hiVOICE_G726_STATE_S”。 | 输入/输出 | 全局 |
| nCoder | 用户指定的解码设备类型，类型定义请参见“3.1.1 常数定义”。 | 输入 | 局部 |

【返回值】

| 返回值 | 描述 |
|-----|------------|
| 0 | 成功。 |
| 非 0 | 失败，其值为错误码。 |

【错误码】

| 错误码 | 描述 |
|-----------------------------|--------------|
| HI_ERR_VOICE_DEC_TYPE | 解码器类型非法。 |
| HI_ERR_VOICE_INVALID_DEVICE | 输入编解码设备指针非法。 |

【需求】

头文件：/include/hi_voice_api.h

【注意】

- G711_ORG_A、G711_ORG_U、ADPCM_ORG_DVI4、G726_ORG_16KBPS 四种解码设备类型仅针对 Hi3510_VSSDK_V1.3.1.0 版本解码，之后版本不推荐使用。

【举例】

```
HI_VOID VoiceDemo()  
{  
    HI_RESULT Ret;  
    hiVOICE_G711_STATE_S g_sG711DecState;  
    Ret=HI_VOICE_DecReset (&g_sG711DecState, G711_A);  
    if (Ret != 0)  
        MyErrorExit("HISI_VOICE_dec_reset");  
    else
```



```
printf("HISI_VOICE_dec_reset SUCCESS.\n");  
}
```

【相关主题】

- [HI_VOICE_DecodeFrame](#)

2.3 海思语音编解码器

编码器对原始语音数据进行编码，打包成海思语音帧输出；解码器对海思语音帧解码，输出解码语音数据。

HI_VOICE_EncodeFrame

【目的】

进行语音编码。

【语法】

```
#include "hi_voice_api.h"  
HI_VOICE_API HI_RESULT HI_VOICE_EncodeFrame(HI_VOID *pEncState,  
HI_S16 *pInputBuf,  
HI_S16 *pOutputBuf,  
HI_S16 len);
```

【描述】

用户传入待编码语音数据，编码器编码后，把已编码数据按照海思语音帧结构存放在用户指定的输出缓冲区。注意数据单位为 Word16，最大帧长为 **HI_VOICE_MAX_FRAME_SIZE**，除 ADPCM_IMA 编解码器之外，必须保证帧长为 80 的整数倍，即 80、160、240、320、480。ADPCM_IMA 编解码器需要多输入一个样点作为参考每帧编码的参考电平，即帧长为 81、161、241、321、481。

【参数】

| 参数名称 | 描述 | 输入/输出 | 全局/局部 |
|------------|------------|-------|-------|
| pEncState | 用户指定的编码设备。 | 输入/输出 | 全局 |
| pInputBuf | 待编码数据输入缓冲。 | 输入 | 局部 |
| pOutputBuf | 已编码数据输出缓冲。 | 输出 | 局部 |
| len | 数据帧长度。 | 输入 | 局部 |

【返回值】



| 返回值 | 描述 |
|-----|------------|
| 0 | 成功。 |
| 非 0 | 失败，其值为错误码。 |

【错误码】

| 错误码 | 描述 |
|---|--------------|
| HI_ERR_VOICE_ENC_TYPE | 编码器类型非法。 |
| HI_ERR_VOICE_INVALID_DEVICE | 输入编解码设备指针非法。 |
| HI_ERR_VOICE_INVALID_INBUF | 输入缓冲指针非法。 |
| HI_ERR_VOICE_INVALID_OUTBUF | 输出缓冲指针非法。 |
| HI_ERR_VOICE_ENC_FRAME_SIZE | 编码长度非法。 |

【需求】

头文件：/include/hi_voice_api.h

【注意】

- 编码前，必须保证编码器已经正确初始化。

【举例】

```
HI_VOID VoiceDemo()  
{  
    Ret=HI_VOICE_EncodeFrame(&g_sG711EncState,in_pcm_buf,ou_enc_buf,MAX_FRAME  
_SIZE);  
}
```

【相关主题】

- [HI_VOICE_EncReset](#)
- [海思语音帧结构](#)

HI_VOICE_DecodeFrame

【目的】

解码器对输入的 1 帧海思语音帧数据解码，输出解码语音数据。

【语法】

```
#include "hi_voice_api.h"  
HI_VOICE_API HI_RESULT HI_VOICE_DecodeFrame (HI_VOID *pDecState,  
HI_S16 * pInputBuf,
```



```
HI_S16 * pOutputBuf,  
HI_S16 * pLen);
```

【描述】

用户传入编码压缩帧，解码器解码后，把已解码的声音数据存放在用户指定的输出缓冲区。

【参数】

| 参数名称 | 描述 | 输入/输出 | 全局/局部 |
|------------|-------------|-------|-------|
| pDecState | 用户指定的解码设备。 | 输入/输出 | 全局 |
| pInputBuf | 压缩码流的输入缓冲区。 | 输入 | 局部 |
| pOutputBuf | 解码数据的输出缓冲区。 | 输出 | 局部 |
| pLen | 解码输出数据个数。 | 输出 | 局部 |

【返回值】

| 返回值 | 描述 |
|-----|------------|
| 0 | 成功。 |
| 非 0 | 失败，其值为错误码。 |

【错误码】

| 错误码 | 描述 |
|-----------------------------|--------------|
| HI_ERR_VOICE_DEC_TYPE | 解码器类型非法。 |
| HI_ERR_VOICE_INVALID_DEVICE | 输入编解码设备指针非法。 |
| HI_ERR_VOICE_INVALID_INBUF | 输入缓冲指针非法。 |
| HI_ERR_VOICE_INVALID_OUTBUF | 输出缓冲指针非法。 |
| HI_ERR_VOICE_DEC_FRAMETYPE | 语音帧类型非法。 |
| HI_ERR_VOICE_DEC_FRAME_SIZE | 净荷长度非法。 |

【需求】

头文件：/include/hi_voice_api.h

【注意】



- 解码前，必须保证解码器已经正确初始化，并确保输出缓冲区足以容纳一帧解码数据。压缩码流格式采用海思语音帧结构。

【举例】

```
HI_VOID VoiceDemo()  
{  
    Ret = HI_VOICE_DecodeFrame(&g_sG711DecState, in_unpacked, ou_dec_pcmbuf,  
                                &len);  
}
```

【相关主题】

- [HI_VOICE_DecReset](#)
- [海思语音帧结构](#)

2.4 海思语音转码器初始化

HI_VOICE_TransCodeReset

【目的】

开始转码任务时，初始化转码设备。

【语法】

```
#include "hi_voice_api.h"  
HI_VOICE_API HI_RESULT HI_VOICE_TransCodeReset (HI_VOID *pTransState,  
HI_S16 nCoder);
```

【描述】

初始化转码设备。

【参数】

| 参数名称 | 描述 | 输入/输出 | 全局/局部 |
|-------------|--|-------|-------|
| pTransState | 用户指定的转码设备，设备定义请参见“ 3.1.5 hiVOICE_TRANSCODE_STATE_S ”。 | 输入 | 全局 |
| nCoder | 用户指定的转码设备类型。转码设备类型由 Hi3510_VSSDK_V1.3.1.0 版本编码器类型确定，类型定义请参见“ 3.1.1 常数定义 ”。 | 输入 | 局部 |

【返回值】



| 返回值 | 描述 |
|-----|------------|
| 0 | 成功。 |
| 非 0 | 失败，其值为错误码。 |

【错误码】

| 错误码 | 描述 |
|---|-------------|
| HI_ERR_VOICE_TRANS_TYPE | 转码器类型非法。 |
| HI_ERR_VOICE_TRANS_DEVICE | 输入转码设备指针非法。 |

【需求】

头文件：/include/hi_voice_api.h

【注意】

- 仅针对 Hi3510_VSSDK_V1.3.1.0 版本编码码流进行转码。

【举例】

```
HI_VOID VoiceDemo()  
{  
    HI_RESULT Ret;  
    hiVOICE_TRANSCODE_STATE_S g_TransState;  
    Ret= HI_VOICE_TransCodeReset (&g_TransState, ORG_G711A_TRANSF2_G711A);  
    if (Ret != 0)  
        MyErrorExit("HISI_VOICE_trans_reset");  
    else  
        printf("HISI_VOICE_trans_reset SUCCESS.\n");  
}
```

【相关主题】

- [HI_VOICE_TransCodeFrame](#)

2.5 海思语音转码器

将 Hi3510_VSSDK_V1.3.1.0 版本 G711A、G711U、ADPCM_DVI4、G726_16KBPS 编码码流转为之后版本编码码流，打包成海思语音帧输出。

HI_VOICE_TransCodeFrame

【目的】



进行语音编码转换。

【语法】

```
#include "hi_voice_api.h"

HI_VOICE_API HI_RESULT HI_VOICE_TransCodeFrame (HI_VOID *pTransState,
HI_S16 *pInputBuf);
```

【描述】

用户传入待转码压缩帧，转码完成后，把已转码数据按照海思语音帧结构存放在输入缓冲区。

【参数】

| 参数名称 | 描述 | 输入/输出 | 全局/局部 |
|-------------|--------------------------|-------|-------|
| pTransState | 用户指定的转码设备。 | 输入/输出 | 全局 |
| pInputBuf | 待转码数据输入缓冲。 转码后数据输出缓冲。 | 输入/输出 | 局部 |

【返回值】

| 返回值 | 描述 |
|-----|------------|
| 0 | 成功。 |
| 非 0 | 失败，其值为错误码。 |

【错误码】

| 错误码 | 描述 |
|----------------------------|-------------|
| HI_ERR_VOICE_TRANS_TYPE | 转码器类型非法。 |
| HI_ERR_VOICE_TRANS_DEVICE | 输入转码设备指针非法。 |
| HI_ERR_VOICE_INVALID_INBUF | 输入缓冲指针非法。 |

【需求】

头文件：/include/hi_voice_api.h

【注意】

- 转码前，必须保证转码器已经正确初始化。
- 需保证输入编码码流为 Hi3510_VSSDK_V1.3.1.0 版本编码码流。

【举例】



```
HI_VOID VoiceDemo()  
{  
    Ret=HI_VOICE_TransCodeFrame (&g_TransState, in_enc_buf);  
}
```

【相关主题】

- [HI_VOICE_TransCodeReset](#)
- [海思语音帧结构](#)



3 其它信息

3.1 数据类型

3.1.1 常数定义

```
/* 编码器最大帧长，即数据个数 */
#define HI_VOICE_MAX_FRAME_SIZE      481

/* 海思编解码器类型 */
/* Real-time transport protocol (RTP) */

/* 64kbit/s G.711 A, see RFC3551.txt 4.5.14 PCMA */
#define G711_A                        0x01

/* 64kbit/s G.711 U, see RFC3551.txt 4.5.14 PCMU */
#define G711_U                        0x02

/* 32kbit/s ADPCM (DVI4) for RTP; see DVI4 differs in three respects from
the IMA ADPCM at RFC3551.txt 4.5.1 DVI4 */
#define ADPCM_DVI4                    0x03

/* 16kbit/s G.726, see RFC3551.txt 4.5.4 G726-16 */
#define G726_16KBPS                    0x04

/* 24kbit/s G.726, see RFC3551.txt 4.5.4 G726-24 */
#define G726_24KBPS                    0x05

/* 32kbit/s G.726, see RFC3551.txt 4.5.4 G726-32 */
#define G726_32KBPS                    0x06

/* 40kbit/s G.726, see RFC3551.txt 4.5.4 G726-40 */
#define G726_40KBPS                    0x07

/* IMA ADPCM wave type, 32kbit/s ADPCM (IMA) */
#define ADPCM_IMA                      0x23
```



3 其它信息

```

/*
An example of the packing scheme for G726-32 codewords is as shown and
bit A3 is the least significant bit of the first codeword:
RTP G726-32:
0          1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+--+--+--+--+--+--+--+--+--+--+
|B B B B|A A A A|D D D D|C C C C| ...
|0 1 2 3|0 1 2 3|0 1 2 3|0 1 2 3|
+-+--+--+--+--+--+--+--+--+--+--+

MEDIA G726-32:
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+--+--+--+--+--+--+--+--+--+--+
|A A A A|B B B B|C C C C|D D D D| ...
|3 2 1 0|3 2 1 0|3 2 1 0|3 2 1 0|
+-+--+--+--+--+--+--+--+--+--+--+
*/

#define MEDIA_G726_16KBPS      0x24 /* G726 16kbit/s for ASF */
#define MEDIA_G726_24KBPS      0x25 /* G726 24kbit/s for ASF */
#define MEDIA_G726_32KBPS      0x26 /* G726 32kbit/s for ASF */
#define MEDIA_G726_40KBPS      0x27 /* G726 40kbit/s for ASF */

/* Hi3510_VSSDK_V1.3.1.0版本编解码器类型 */
#define G711_ORG_A              0x41 /* 64kbit/s G.711 A */
#define G711_ORG_U              0x42 /* 64kbit/s G.711 U */
#define ADPCM_ORG_DVI4          0x43 /* 32kbit/s ADPCM */
#define G726_ORG_16KBPS         0x44 /* 16kbit/s G.726 */

/* 转码器类型 */
/* transform Hi3510_VSSDK_V1.3.1.0 64kbit/s G.711 A to current G711_A */
#define ORG_G711A_TRANSF2_G711A 0x61

/* transform Hi3510_VSSDK_V1.3.1.0 64kbit/s G.711 U to current G711_U */
#define ORG_G711U_TRANSF2_G711U 0x62

/* transform Hi3510_VSSDK_V1.3.1.0 32kbit/s ADPCM to current
ADPCM_DVI4 */
#define ORG_ADPCM_DVI4_TRANSF2_ADPCM_DVI4 0x63

/* transform Hi3510_VSSDK_V1.3.1.0 16kbit/s G.726 to current
G726_16KBPS */
#define ORG_G726_16KBPS_TRANSF2_G726_16KBPS 0x64

```



```
/* 海思语音帧类型 */  
  
#define HI_NORMAL_FRAME          0x01    /* 语音帧 */  
  
/* 返回码定义 */  
  
#define HI_SUCCESS                0x00    /* 成功 */  
  
#define HI_ERR_VOICE_PREFIX      0xA1220000 /* 错误码前缀定义 */  
  
/* 返回错误码定义 */  
  
#define HI_ERR_VOICE_ENC_TYPE      (HI_ERR_VOICE_PREFIX | 0x0001)  
#define HI_ERR_VOICE_ENC_FRAME_SIZE (HI_ERR_VOICE_PREFIX | 0x0002)  
#define HI_ERR_VOICE_DEC_TYPE      (HI_ERR_VOICE_PREFIX | 0x0011)  
#define HI_ERR_VOICE_DEC_FRAME_SIZE (HI_ERR_VOICE_PREFIX | 0x0012)  
#define HI_ERR_VOICE_DEC_FRAME_TYPE (HI_ERR_VOICE_PREFIX | 0x0013)  
#define HI_ERR_VOICE_INVALID_DEVICE (HI_ERR_VOICE_PREFIX | 0x0101)  
#define HI_ERR_VOICE_INVALID_INBUF  (HI_ERR_VOICE_PREFIX | 0x0102)  
#define HI_ERR_VOICE_INVALID_OUTBUF (HI_ERR_VOICE_PREFIX | 0x0103)  
#define HI_ERR_VOICE_TRANS_DEVICE   (HI_ERR_VOICE_PREFIX | 0x1001)  
#define HI_ERR_VOICE_TRANS_TYPE     (HI_ERR_VOICE_PREFIX | 0x1002)
```

3.1.2 hiVOICE_G711_STATE_S

【说明】

针对 G711 标准的编码器、解码器的通道变量。

【定义】

```
/* State for G711 encoder and decoder */  
typedef struct hiVOICE_G711_STATE  
{  
    HI_S32 G711StateBuf[2];  
} hiVOICE_G711_STATE_S;
```

【注意事项】

编码器、解码器需分别创建通道变量。

3.1.3 hiVOICE_ADPCM_STATE_S

【说明】

针对 ADPCM 标准的编码器、解码器的通道变量。

【定义】

```
/* State for ADPCM encoder and decoder */  
typedef struct hiVOICE_ADPCM_STATE
```



```
{
    HI_S32 AdpcmStateBuf[4];
} hiVOICE_ADPCM_STATE_S;
```

【注意事项】

编码器、解码器需分别创建通道变量。

3.1.4 hiVOICE_G726_STATE_S

【说明】

针对 G.726 标准的编码器、解码器的通道变量。

【定义】

```
/* State for G726 encoder and decoder */
typedef struct hiVOICE_G726_STATE
{
    HI_S32 G726StateBuf[32];
} hiVOICE_G726_STATE_S;
```

【注意事项】

编码器、解码器需分别创建通道变量。

3.1.5 hiVOICE_TRANSCODE_STATE_S

【说明】

将 Hi3510_VSSDK_V1.3.1.0 版本 G711A、G711U、ADPCM_DVI4、G726_16KBPS 编码转为之后版本编码的通道变量。

【定义】

```
/* State for TransCode */
typedef struct hiVOICE_TRANSCODE_STATE
{
    HI_S32 TransStateBuf[0x400];
} hiVOICE_TRANSCODE_STATE_S;
```

【注意事项】

无。

3.2 错误码

| 错误码 | 描述 |
|-----------------------|----------|
| HI_ERR_VOICE_ENC_TYPE | 编码器类型非法。 |



| 错误码 | 描述 |
|-----------------------------|-----------|
| HI_ERR_VOICE_ENC_FRAME_SIZE | 编码数据帧长非法。 |
| HI_ERR_VOICE_DEC_TYPE | 解码器类型非法。 |
| HI_ERR_VOICE_DEC_FRAME_SIZE | 解码数据帧长非法。 |
| HI_ERR_VOICE_DEC_FRAME_TYPE | 语音帧类型非法。 |
| HI_ERR_VOICE_INVALID_DEVICE | 编解码设备非法。 |
| HI_ERR_VOICE_INVALID_INBUF | 输入缓冲区非法。 |
| HI_ERR_VOICE_INVALID_OUTBUF | 输出缓冲区非法。 |
| HI_ERR_VOICE_TRANS_TYPE | 转码器类型非法。 |
| HI_ERR_VOICE_TRANS_DEVICE | 转码设备非法。 |