全 志 科 技
Allwinner Technology

音频编码库 API 说明

文档履历

版本号	日期	制/修订人	内容描述
V0.1	2016-11-2	程衍	建立初稿

目 录

1. Aencoder 主要 API 及其说明

API 列表

音频编码库 APIs 列		
1	CreateAudioEncoder	创建一个编码库实作句柄
2	DestroyAudioEncoder	销毁一个编码库实作句柄
3	InitializeAudioEncoder	初始化编码库实作句柄
4	ResetAudioEncoder	重置编码库
5	EncodeAudioStream	编码一帧码流
6	WriteAudioStreamBuffer	App 向 PBM 申请上传 Pcm 数据
7	RequestAudioFrameBuffer	Muxer 向 OBM 申请占用并调走一帧编码好的数据
8	ReturnAudioFrameBuffer	Muxer 已经调走 OBM 一帧数据，此时 OBM 释放这一帧

1.1 CreateAudioEncoder

函数原型	AudioEncoder* CreateAudioEncoder()
功能	创建一个音频编码库
参数	无
返回值	成功：音频编码库句柄； 失败：返回 NULL；
调用说明	First called

1.2 DestroyAudioEncoder

函数原型	void DestroyAudioEncoder(AudioEncoder* pEncoder)
功能	销毁一个音频编码库
参数	编码库句柄
返回值	无
调用说明	Last called

1.3 InitializeAudioEncoder

函数原型	void InitializeAudioEncoder(AudioEncoder* pEncoder, AudioEncConfig *pConfig)
功能	传入编码配置，初始化编码库
参数	pEncoder：编码库句柄 pConfig：编码配置内容
返回值	0：Success -1：Fail
调用说明	After CreateAudioEncoder, before encFrame

1.4 ResetAudioEncoder

函数原型	int ResetAudioEncoder (AudioEncoder* pEncoder)
功能	重置编码库
参数	pEncoder：编码库句柄
返回值	0
调用说明	无

1.5 EncodeAudioStream

函数原型	int EncodeAudioStream (AudioEncoder* pEncoder)
功能	编码一帧 pcm
参数	pEncoder：编码库句柄
返回值	见 audio enc result t
调用说明	无

1.6 WriteAudioStreamBuffer

函数原型	int WriteAudioStreamBuffer(AudioEncoder *pEncoder, char* pBuf, int len)
功能	申请 PBM 空间，若成功则将 pBuf 中的 Pcm 数据上报 PBM.
参数	pEncoder：编码库句柄 pBuf：Pcm 数据首地址 len：Pcm 数据空间长度
返回值	0：Success -1：Fail
调用说明	Audio Eecoding Component 或者 APP 专用

	do{
	ret = WriteAudioStreamBuffer
	if(ret == -1)
	sleep;
	if(ret == 0)
	break;
	}while(1)

1.7 RequestAudioFrameBuffer

函数原型	int RequestAudioFrameBuffer(AudioEncoder *pEncoder, char **pOutBuf, unsigned int *size, long long *pts, int *bufId)
功能	MUXER 申请 OBM 数据进行 MUXER
参数	pEncoder：编码库句柄 pOutBuf： OBM 中数据的首地址填给它 size: OBM 中数据的长度给它 pts: OBM 中数据的 pts 给它 bufId: 本次从 OBM 中第 bufId 个拿走数据
返回值	0：Success -1: Fail
调用说明	同 WriteAudioStreamBuffer，调用者从 App 变成 Muxer

1.8 ReturnAudioFrameBuffer

函数原型	int ReturnAudioFrameBuffer(AudioEncoder *pEncoder, char *pOutBuf, unsigned int size, long long pts, int bufId)
功能	Muxer 已经调走 OBM 一帧数据，此时 OBM 释放这一帧
参数	pEncoder：编码库句柄 pOutBuf： OBM 中数据的首地址填 size: OBM 中数据的长度 pts: OBM 中数据的 pts bufId: OBM 中第 bufId 个数据
返回值	0：Success -1: Fail
调用说明	Must be called after RequestAudioFrameBuffer

2 Aencoder 主要数据结构及其说明

2.1 __pcm_buf_manager_t

__pcm_buf_manager_t **(PBM)**		
成员类型	**成员名称**	**说明**
unsigned char *	pBufStart	环形 buffer 起始地址
int	uBufTotalLen	环形 buffer pcm 数据吞吐总长
unsigned char *	pBufReadPtr	当前 Aencoder 已消费 pcm 数据的偏移地址
int	uDataLen	当前 PBM 有多少 Pcm 数据可供 Aecoder 消费
unsigned char *	pBufWritPtr	当前应用线程已填充的 pcm 数据的偏移地址
int	uFreeBufSize	当前有多少空间可让应用线程填入 Pcm 数据
int	uDataFlowflag	编码库内部使用，目前 aac 编码库专用。编码速度过慢时，丢弃待编码 pcm 数据，并补 aac 空帧，此位置 1。速度回归时，复位 0。此变量合理与否待考究，不关注
Void *	parent	Aencoder 自身句柄，供编码库回调数据使用

2.2 OutBufManager_t

OutBufferManager_t(OBM)		
成员类型	**成员名称**	**说明**
AudioOutBuf_t []	out_buf	AudioOutBuf_t 结构体数组，目前数组有 FIFO_LEVEL = 32 个。AudioOutBuf_t 保存输出数据的首地址，长度，及其 PTS
int	write_id	EncEngine 数据输出到 OBM 的位置
int	read_id	同 prefetch_id，只是 prefetch_id 在 Muxer Get buffer 成功时，自加更新。而 read_id 只有在 Muxer free buffer 成功时才自加更新。一般正常情况下，两者相等
int	prefetch_id	Muxer 在 OBM 中数据消费的位置
int	buf_unused	FIFO_LEVEL = 32 个 Buffer 中，

		有多少个没有被 encengine 填数据

2.3 AudioOutBuf_t

AudioOutBuf_t		
注	OBM 中对数据的封装结构体	
成员类型	成员名称	说明
void *	buf	数据首地址
int	size	数据长度
unsigned int	timeStamp	数据时间戳

2.4 __audio_enc_inf_t

__audio_enc_inf_t		
注	输入的 pcm 数据规格	
成员类型	成员名称	说明
int	InSamplerate	输入采样率
int	InChan	输入通道数
int	bitrate	输入的比特率
int	SamplerBits	输入的 Pcm 数据位宽，只支持 16bit
int	OutSamplerate	输出的压缩数据的采样率。和输入的采样率必须相等
int	frame_style	对 aac 0 - 加 adts 头 1 - 不加头 对 pcm 2 - 大段 pcm，否则小段

2.5 __com_internal_prameter_t

__com_internal_prameter_t		
注	编码 engine 和 aencoder 共享信息结构体	
成员类型	成员名称	说明
unsigned int	ulNowTimeMS	当前累计编码总时间长度
unsigned int	ulPCMgainSet	输入 Pcm 数据增益调制接口，目前没用

unsigned int	framecount	当前编了多少笔码流
unsigned int	ulEncCom	外部控制录制状态接口，告知编码库当前状况 0：continue 5：stop
unsigned char [BS_HEADER_SIZE]	BsHeaderBuf	特种格式头信息预生成 Buffer. 比如 Wav：44byte Aac：7byte
unsigned int	ValidHeaderLen	头信息有效长度
unsigned int *	pEncInfoSet	编码 engine 内部句柄，内部生成，内部访问，内部销毁

2.6 AudioEnc_AC320

AudioEnc_AC320		
注	编码库具体实作句柄	
成员类型	成员名称	说明
__pcm_buf_manager_t *	pPcmBufManager	PBM
__audio_enc_inf_t *	AudioBsEncInf	编码库输入 Pcm 数据规格结构体
__com_internal_prameter_t *	EncoderCom	编码 engine 和 aencoder 共享信息结构体
int	Encinitedflag	编码 engine 启动标志，只启动一次
int (*Func)(struct __AudioENC_AC320 *p);	EncInit	EncEngine 初始化 hook
int (*Func)(struct __AudioENC_AC320 *p,char *OutBuffer,int *OutBuffLen);	EncFrame	EncEngine 编码 hook
int (*Func)(struct __AudioENC_AC320 *p);	EncExit	EncEngine 退出 hook

3 Aencoder 主要枚举变量及其说明

3.1 __audio_enc_result_t

__audio_enc_result_t	
注	编码返回结果

成员名称	说明
ERR_AUDIO_ENC_ABSEND = -2	编码结束，没有任何 pcm 数据输入并且外部已退出
ERR_AUDIO_ENC_UNKNOWN = -1	未知编码错误
ERR_AUDIO_ENC_NONE = 0	正确编码
ERR_AUDIO_ENC_PCMUNDERFLOW = 1	PBM 没有足够空间供 app 写入 Pcm 数据
ERR_AUDIO_ENC_OUTFRAME_UNDERFLOW =2	OBM 没有足够空间供 EncEngine 写入编码数据
ERR_AUDIO_ENC_	略

3.1 AUDIO_ENCODER_TYPE

AUDIO_ENCODER_TYPE	
注	编码数据类型
成员名称	说明
AUDIO_ENCODER_AAC_TYPE	AAC
AUDIO_ENCODER_LPCM_TYPE	LPCM, for blue and mpeg2ts
AUDIO_ENCODER_PCM_TYPE	Wav or adpcm,目前 adpcm 没有启用，默认 wav
AUDIO_ENCODER_MP3_TYPE	MP3