1. 初始化

可配置的参数：（红色表示必须，黄色为初始化后反馈的参数）

AVCodecContext

{

channels // [1, 6]

sample\_rate // 可在ff\_libfdk\_aac\_encoder.supported\_samplerates里查看有哪些值

sample\_fmt // 可在ff\_libfdk\_aac\_encoder.sample\_fmts里查看

profile // 默认值FF\_PROFILE\_AAC\_LOW

flags

global\_quality

bit\_rate

cutoff // 设置带宽的，默认值0，[(sample\_rate + 255) >> 8, 20000]

frame\_size

initial\_padding

extradata\_size

extradata

}

typedef struct AACContext {

const AVClass \*class;

HANDLE\_AACENCODER handle;

int afterburner; //取值0或1，默认值1

int eld\_sbr; //取值0或1，默认值0，profile=FF\_PROFILE\_AAC\_ELD时开启

int signaling; //取值[-1, 2]，默认值-1，default=-1/implicit=0/explicit\_sbr=1/explicit\_hierarchical=2

int latm; //取值0或1，默认值0

int header\_period; //取值[0, 0xFFFF]，默认值0

int vbr; //取值[0, 5]，默认值0，表示恒定码率

AudioFrameQueue afq;

} AACContext;

初始化编码器libfdk-aac.c::aac\_encode\_init()

|  |  |  |
| --- | --- | --- |
| Ffmpeg profile | AACENC\_AOT | 描述 |
| FF\_PROFILE\_AAC\_LOW = 1 | LC = 2 | MPEG4 Low Complexity |
| FF\_PROFILE\_AAC\_HE = 4 | HE-AAC = 5 | MPEG4 High Efficiency(LC with SBR) |
| FF\_PROFILE\_AAC\_HE\_V2 = 28 | HE-AAC v2 = 29 | MPEG4 High Efficiency v2(LC with SBR & PS) |
| FF\_PROFILE\_AAC\_LD = 22 | LD = 23 | MPEG4 Low Delay |
| FF\_PROFILE\_AAC\_ELD = 38 | ELD = 39 | MPEG4 Enhanced Low Delay(should enable SBR) |

当profile为FF\_PROFILE\_AAC\_ELD时，eld\_sbr应配置为1

采样率avctx->sample\_rate取值：{8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100, 48000, 64000, 88200, 96000}

通道avctx->channels取值：{1, 2, 3, 4, 5, 6, 8}

|  |  |
| --- | --- |
| channels | Channel\_layout |
| 1 | MODE\_1 |
| 2 | MODE\_2 |
| 3 | MODE\_1\_2 |
| 4 | MODE\_1\_2\_1 |
| 5 | MODE\_1\_2\_2 |
| 6 | MODE\_1\_2\_2\_1 |
| 8 | avctx->channel\_layout == AV\_CH\_LAYOUT\_7POINT1, MODE\_7\_1\_REAR\_SURROUND;  否则，MODE\_7\_1\_FRONT\_CENTER |

Ffmpeg中固定使用了WAVE格式的通道序，即AACENC\_CHANNELORDER = 1；可修改该值为0，使用MPEG的通道序

通道个数必须与输入匹配

码率配置：

avctx->flags = AV\_CODEC\_FLAG\_QSCALE 或者 vbr != 0，开启可变码率模式；vbr优先于avctx->flags，如果只使用avctx->flags = AV\_CODEC\_FLAG\_QSCALE开启可变码率模式，那么将通过avctx->global\_quality决定级别，取值也是[1,5]，可以随便指定，会被截断的。

avctx->bit\_rate > 0，开启恒定码率模式，并且设置固定码率为avctx->bit\_rate。

如果没有开启可变码率模式，并且没有配置avctx->bit\_rate，那么ffmpeg根据profile推算码率，并配置为恒定码率。

可变码率优先于恒定码率，即如果配置了avctx->bit\_rate，也开启了可变码率模式，使用可变码率模式。

选择比特流帧格式：

|  |  |
| --- | --- |
| AACENC\_TRANSMUX | 编码器输出的帧格式 |
| 0 | Raw access unit |
| 1 | ADIF，FFMPEG不支持 |
| 2 | ADTS |
| 6 | Audio Mux Elements (LATM) with muxConfifigPresent = 1，ffmpeg不支持 |
| 7 | Audio Mux Elements (LATM) with muxConfifigPresent = 0，ffmpeg不支持 |
| 10 | Audio Sync Stream (LOAS) |

avctx->flags = AV\_CODEC\_FLAG\_GLOBAL\_HEADER时，配置为0，编码器输出裸流帧，可用adts的muxer打包成ADTS格式。

当avctx->flags != AV\_CODEC\_FLAG\_GLOBAL\_HEADER时，由latm指定，0表示2，1表示10

header\_period选项：

Fdk-aac文档描述如下，不清楚具体作用。

AACENC\_HEADER\_PERIOD Frame count period for sending in-band confifiguration buffers within

LATM/LOAS transport layer. Additionally this parameter confifigures the PCE repetition period

in raw\_data\_block(). See encPCE.

• 0xFF: auto-mode default 10 for TT\_MP4\_ADTS, TT\_MP4\_LOAS and TT\_MP4\_LATM\_-

MCP1, otherwise 0.

• n: Frame count period

Signaling选项：

具体参考fdk-aac文档。

|  |  |
| --- | --- |
| AACENC\_SIGNALING\_MODE | 描述 |
| 0 | Implicit backward compatible signaling，可被任何aac解码器解码 |
| 1 | Implicit backward compatible signaling，可被任何aac解码器解码 |
| 2 | Explicit hierarchical signaling，只能被HE-AAC解码器解码 |

默认值为-1，可主动配置该选项为0,1,2

当为默认值时，ffmepg根据avctx->flags == AV\_CODEC\_FLAG\_GLOBAL\_HEADER，配置该选项：

当avctx->flags == AV\_CODEC\_FLAG\_GLOBAL\_HEADER时，配置为2；否则，配置为0

Afterburner选项：

这是一个增强音质的算法的开关，它会消耗内存和CPU资源。Ffmpeg中默认是开启的，当计算机资源不充足时，可考虑关闭该算法。

avctx->cutoff参数：

配置编码器带宽AACENC\_BANDWIDTH。

AACENC\_BANDWIDTH的默认值为0，由编码器内部推断带宽，最好不要动这个选项。

当编码器配置完成后，ffmpeg会从编码器中取得一些返回信息。

avctx->frame\_size

avctx->initial\_padding

如果avctx->flags == AV\_CODEC\_FLAG\_GLOBAL\_HEADER，那么avctx->extradata\_size，avctx->extradata也会更新，注意这两个值是adts muxer需要使用的，adts muxer根据avctx->extradata\_size是否为0，决定要不要组装ADTS格式。如果为0，那么ffmpeg使编码器直接输出ADTS的帧，就不需要再组装了；如果不为0，那么ffmpeg使编码器输出裸流帧，需要组装ADTS格式。

1. 编码

编码时，最重要的两个参数是：输出AVPacket和输入AVFrame

AVFrame必须要有的参数：

{

data

nb\_samples // 与AVCodecContext.frame\_size相等，单通道采样数

pts

}

AVPacket编码后的输出

{

data

size

pts

duration

}

1. 编译

--enable-encoder=libfdk\_aac --enable-gpl --enable-nonfree --enable-libfdk-aac --extra-cflags=”-Ipath/to/fdk-aac headers” --extra-ldflags=”-Lpath/to/fdk-aac library directory” --extra-libs=”-lfdk-aac”

注意：ffmpeg在配置时，试探性的运行一些测试，所以需要把--extra-ldflags=”-Lpath/to/fdk-aac library directory”这个路径，配置到LD\_LIBRARY\_PATH环境变量里

编译ffmpeg时，最好在新建的目录里执行configure，不污染源代码；如果在源码目录里执行了configure，那么在新建目录里配置、编译时，会跟预期不一样。