杰理iOS音频编码库开发说明

声明

- 本项目所参考、使用技术必须全部来源于公知技术信息，或自主创新设计。
- 本项目不得使用任何未经授权的第三方知识产权的技术信息。
- 如个人使用未经授权的第三方知识产权的技术信息，造成的经济损失和法律后果由个人承担

历史版本

版本	更新概述	修改人	备注
1.0.1	1.增加OpusUnit的重要接口说明 2.增加OpusUnit的使用示例	凌煊峰	2021.07.19
1.0.2	1.修复编解码时较低概率发生的崩溃问题	凌煊峰	2021.07.30

一、概述

本文档是为了方便用户接入杰理音频编解码库而创建，用户可以通过该文档快速接入杰理音频编解码库功能。
杰理音频编解码库已支持功能：
1.Opus编解码使用说明
  1.1 Opus文件 解码成 PCM文件
  1.2 Opus数据流 解码成 PCM数据流
  1.3 PCM文件 编码成 Opus文件
  1.4 PCM数据流 编码成 Opus数据流
2.Speex编解码
  2.1 Speex解码文件
  2.2 Speex解码数据流
  2.3 Speex编码文件

二、接口说明

2.1 Opus编解码接口说明

OpusUnit

```
#import <Foundation/Foundation.h>

NS_ASSUME_NONNULL_BEGIN

/**
 *  通知：流式编码回调Opus数据
 *  通知数据类型：NSData（Opus）
 */
extern NSString *kOPUS_ENCODE_DATA;
/**
 *  通知：流式解码回调PCM数据
 *  通知数据类型：NSData（PCM）
 */
extern NSString *kOPUS_DECODE_DATA;


@interface OpusUnit : NSObject

+ (void)opusIsLog:(BOOL)log;

#pragma mark - Opus解码

/**
 *  直接【opus文件】转换成【pcm文件】
 *  @param path_opus    opus文件路径
 *  @param path_pcm    pcm文件路径
```

```
 */
+ (int)opusDecodeOPUS:(NSString *)path_opus PCM:(NSString *)path_pcm;

/**
 *  流式解码【开启】
 */
+ (int)opusDecoderRun;

/**
 *  输入Opus的数据
 *  通知监听“kOPUS_DECODE_DATA”获得解码后数据
 *  通知数据类型：NSNotification.object => NSData（PCM格式）
 *  @param data    opus数据流（长度：1024）
 */
+ (void)opusWriteData:(NSData*)data;

/**
 *  流式解码【关闭】
 */
+ (int)opusDecoderStop;

#pragma mark - Opus编码

/**
 *  【pcm文件】转换成【opus文件】
 */
+ (int)opusEncodePCM:(NSString *)path_pcm OPUS:(NSString *)path_opus;

/**
 *  流式编码【开启】
 */
+ (int)opusEecoderRun;

/**
 *  输入pcm的数据
 *  通知监听“kOPUS_DECODE_DATA”获得解码后数据
 *  通知数据类型：NSNotification.object =>  NSData（opus格式）
 *  @param data    opus数据流（长度：1024）
 */
+ (void)pcmWriteData:(NSData*)data;

/**
 *  流式编码【关闭】
 */
+ (int)opusEncoderStop;

@end

NS_ASSUME_NONNULL_END
```

2.2 Speex编解码接口说明

SpeexUnit

```
#import <Foundation/Foundation.h>

/**
 *  流式解码回调PCM数据
 */
extern NSString *kSPEEX_DECODE_DATA;        //speex decode

@interface SpeexUnit : NSObject

+(void)speexIsLog:(BOOL)log;

/**
 *  直接【speex文件】转换成【pcm文件】
 */
+(int)speexDecodeSPX:(NSString *)path_spx PCM:(NSString *)path_pcm;

/**
 *  流式解码【开启】
 */
+(int)speexDecoderRun;

/**
 *  输入Speex的数据
 */
+(void)speexWriteData:(NSData*)data;

/**
 *  流式解码【关闭】
 */
+(int)speexDecoderStop;

@end
```

三、使用说明

3.1 Opus编解码使用说明

3.1.1 Opus文件 解码成 PCM文件

```
    // 创建PCM文件
    NSString *docsdir = [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES) firstObject];
    NSString *dataFilePath = [docsdir stringByAppendingPathComponent:@"pcm"];
    BOOL isDirExist = NO;
    [[NSFileManager defaultManager] fileExistsAtPath:dataFilePath
isDirectory:&isDirExist];
    if (!isDirExist) {
        [[NSFileManager defaultManager] createDirectoryAtPath:dataFilePath
withIntermediateDirectories:YES attributes:nil error:nil];
    }
    NSString *filePath = [dataFilePath stringByAppendingPathComponent:@"1.pcm"];
    if ([[NSFileManager defaultManager] fileExistsAtPath:filePath]) {
        [[NSFileManager defaultManager] removeItemAtPath:filePath error:nil];
        NSLog(@"remove file : %@", filePath);
    }
    NSString *txt = @"";
    [txt writeToFile:filePath atomically:YES encoding:NSUTF8StringEncoding error:nil];

    // 调用解码方法，输入 Opus文件路径 与 PCM文件路径
    int result = [OpusUnit opusDecodeOPUS:[[NSBundle mainBundle] pathForResource:@"o3o"
ofType:@"opus"] PCM:filePath];

    if (result == 0) {
        // 解码成功
        NSLog(@"opusFileToPCMFile OK 了");
    }
```

3.1.2 Opus数据流 解码成 PCM数据流

```
//Step0.运行Opus解码库，必须异步调用
// 注意：运行编码库与解码库必须使用gcd分包调用
dispatch_async(dispatch_get_global_queue(0, 0), ^{
    [OpusUnit opusIsLog:YES];//Opus解码库的LOG
    [OpusUnit opusDecoderRun];//运行Opus解码库
});

//Step1.监听解码后PCM数据回调
[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(notePCMData:)
name:kOPUS_DECODE_DATA object:nil];

//Step2.实现解码后PCM数据回调通知监听
- (void)notePCMData:(NSNotification*)note {
```

```
    NSData *data = [note object];
    NSLog(@"--->pcm buffer : %lu",(unsigned long)data.length);

    ...
}

//Step3.当不需要解码时，可以停止解码并释放资源
dispatch_async(dispatch_get_global_queue(0, 0), ^{
    [OpusUnit opusDecoderStop];
});

//StepN.传入OPUS数据到OpusUnit
dispatch_async(dispatch_get_global_queue(0, 0), ^{
    NSString *opusPath = [DFFile find:@"MyAuido.pcm"];
    NSData *opusBuffer = [NSData dataWithContentsOfFile:opusPath];
    unsigned long seek = 0;
    while (1) {
        if (!self.isPlayPCM) {
            break;
        }
        NSData *tmp = [DFTools data:opusBuffer R:seek L:1*1024];
        NSLog(@"---> pcm input data: %lu",(unsigned long)tmp.length);
        if (tmp.length > 0) {
            /*---- 传入PCM数据  ----*/
            [OpusUnit pcmWriteData:tmp];
            sleep(0.1);
            seek = seek + tmp.length;
        } else {
            self.isPlayPCM = NO;
            break;
        }
    }
});
```

3.1.3 PCM文件 编码成 Opus文件

```
    // 创建Opus文件
    NSString *docsdir = [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES) firstObject];
    NSString *dataFilePath = [docsdir stringByAppendingPathComponent:@"opus"];
    BOOL isDirExist = NO;
    [[NSFileManager defaultManager] fileExistsAtPath:dataFilePath
isDirectory:&isDirExist];
    if (!isDirExist) {
        [[NSFileManager defaultManager] createDirectoryAtPath:dataFilePath
withIntermediateDirectories:YES attributes:nil error:nil];
    }
```

```
    NSString *filePath = [dataFilePath stringByAppendingPathComponent:@"1.opus"];
    if ([[NSFileManager defaultManager] fileExistsAtPath:filePath]) {
        [[NSFileManager defaultManager] removeItemAtPath:filePath error:nil];
        NSLog(@"remove file : %@", filePath);
    }
    NSString *txt = @"";
    [txt writeToFile:filePath atomically:YES encoding:NSUTF8StringEncoding error:nil];

    // 调用编码方法，输入 PCM文件路径 与 Opus文件路径
    int result = [OpusUnit opusEncodePCM:[[NSBundle mainBundle]
pathForResource:@"MyAuido" ofType:@"pcm"] OPUS:filePath];

    if (result == 0) {
        // 编码成功
        NSLog(@"pcmFileToOpusBtnFunc OK 了");
    }
```

3.1.4 Pcm数据流 编码成 Opus数据流

```
//Step0.运行Opus编码库，必须异步调用
// 注意：运行编码库与解码库必须使用gcd分包调用
dispatch_async(dispatch_get_global_queue(0, 0), ^{
    [OpusUnit opusIsLog:YES];//Opus解码库的LOG
    [OpusUnit opusEecoderRun];//运行Opus编码库
});

//Step1.监听编码后Opus数据回调
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(noteOpusData:) name:kOPUS_ENCODE_DATA object:nil];

//Step2.实现编码后Opus数据回调通知监听
- (void)noteOpusData:(NSNotification*)note {
    NSData *data = [note object];
    NSLog(@"--->opus buffer : %lu",(unsigned long)data.length);

    ...
}

//Step3.当不需要编码时，可以停止编码并释放资源
dispatch_async(dispatch_get_global_queue(0, 0), ^{
    [OpusUnit opusEncoderStop];
});

//StepN.传入PCM数据到OpusUnit中进行编码
dispatch_async(dispatch_get_global_queue(0, 0), ^{
    NSString *opusPath = [DFFile find:@"MyAuido.pcm"];
```

```
    NSData *opusBuffer = [NSData dataWithContentsOfFile:opusPath];
    unsigned long seek = 0;
    while (1) {
        if (!self.isPlayPCM) {
            break;
        }
        NSData *tmp = [DFTools data:opusBuffer R:seek L:1*1024];
        NSLog(@"---> pcm input data: %lu",(unsigned long)tmp.length);
        if (tmp.length > 0) {
            /*---- 传入PCM数据  ----*/
            [OpusUnit pcmWriteData:tmp];
            sleep(0.1);
            seek = seek + tmp.length;
        } else {
            self.isPlayPCM = NO;
            break;
        }
    }
});
```

3.2 Speex编解码使用说明

类似Opus的使用，省略