# 驱动软件对外接口

1. DIO卡
   1. 参数类型枚举

typedef enum **\_EnableStatus**

{

UNABLE\_STATUS = 0, 不使能

ENABLE\_STATUS = 1 使能

} **EnableStatus**;

说明：使能枚举

typedef enum **\_Outp**utMode{

NULL\_OUTPUT\_MODE = 0x00, 无模式

LEVEL\_OUTPUT\_MODE = 0x01, 电平模式

PWM\_OUTPUT\_MODE = 0x02, PWM模式

BIT\_OUTPUT\_MODE = 0x03 BIT数据流模式

} OutputMode;

说明：DIO输出模式枚举

typedef enum **\_OutputImpedanceMode**

{

HIGH\_IMPEDANCE\_MODE = 0, 高阻模式

PULL\_MODE = 0X01, 拉模式

PUSH\_MODE = 0X02, 推模式

PUSH\_AND\_PULL\_MODE = 0X03 推拉模式

} **OutputImpedanceMode**;

说明：输出阻抗模式

typedef enum **\_VoltageLevel**

{

LOW\_LEVEL = 0x00, 低电平

HIGH\_LEVEL = 0x01, 高电平

INVALID\_LEVEL

} **VoltageLevel**;

说明：电平枚举

typedef enum **\_OutputReferenceVoltage**

{

OUTPUT\_5V\_REF = 0X01, 输出5v

OUTPUT\_12V\_REF = 0X02, 输出12v

OUTPUT\_OUTSIDE\_REF = 0X03 输出外部参考

} **OutputReferenceVoltage**;

说明：输出参考电压

typedef enum **\_ReferenceClock**

{

REFCLK\_100\_MHZ = 0, 100M参考时钟

REFCLK\_20\_MHZ = 1 20M参考时钟

} **ReferenceClock**;

说明：参考时钟

* 1. 参数数据结构体

typedef struct **\_DOLevelConfigure**

{

OutputImpedanceMode eMode; 输出阻抗

OutputReferenceVoltage eRef; 输出参考电压

VoltageLevel eLevel; 输出电平

} **DOLevelConfigure**;

说明：输出电平配置

typedef struct **\_DOPWMConfigure**

{

double dFreq; 频率Hz

double dDuty; 占空比

OutputImpedanceMode eMode; 输出阻抗

OutputReferenceVoltage eRef; 输出参考电压

} **DOPWMConfigure**;

说明：输出PWM配置

typedef struct **\_DIPWMConfigure**

{

double dDurationTime; 采样时间

ReferenceClock eRefClk; 参考时钟（保留，目前为自适应频率）

} **DIPWMConfigure**;

说明：输入PWM配置

typedef struct **\_PWMProperty**

{

double dFreq; 频率Hz

double dDuty; 占空比

} **PWMProperty**;

说明：PWM属性

typedef struct **\_DOBitConfigure**

{

OutputImpedanceMode eMode; 输出阻抗

long lRate; 速率

OutputReferenceVoltage eRef; 输出参考电压

} **DOBitConfigure**;

说明：输出BIT流配置

* 1. 函数接口

int **OpenDio** (const char\* strDev, void \*\*handle);

说明： 打开DIO对应设备文件，获取对应设备文件指针

参数： strDev DIO对应设备文件，如“/dev/dio\_3\_in”，表示打卡3槽DIO输入部分

handle对应设备文件指针

返回值：正常为0

备注：

int **GetDioVersion** (void \*handle,

char\* pStr,

int iLength,

int\* pActLength);

说明： 获取当前DIO卡逻辑固件版本信息

参数： handle 资源设备对应指针

pStr日期版本字符串首地址，用户申请字符数组之数组地址

iLength用户申请的字节数组长度

pActLength获取的版本信息实际有效字节数组长度

返回值：正常为0

备注：

int **CloseDio** (void\* handle);

说明： 关闭DIO对应设备文件

参数： handle资源设备对应指针

返回值：正常为0

备注：

int **SetDoChannelEnable** (void \*handle,

int iChannel,

EnableStatus eStatus);

说明： 设置DIO卡指定通道输出使能

参数： handle资源设备对应指针

iChannel 输出通道号,范围0~23

eStatus使能状态，ENABLE\_STATUS使能输出

返回值：正常为0

备注：

int **SetDoAllChannelEnable** (void \*handle, unsigned int iStatus);

说明： 设置DIO卡全通道输出使能

参数： handle资源设备对应指针

iStatus全通道使能状态，通道按位表示，0bit位为0通道，1为使能，0为未使能，如参数0x03表示输出通道0与1使能输出，其余30个输出通道均未使能输出

返回值：正常为0

备注：

int **GetDoAllChannelEnable** (void \*handle, unsigned int\* pStatus);

说明： 获取全通道使能状态

参数： handle资源设备对应指针

pStatus全通道使能状态，通道按位表示，0bit位为0通道，1为使能，0为未使能，如获取0x03表示输出通道0与1为使能输出状态，其余30个输出通道均未使能输出

返回值：正常为0

备注：

int **SetDoMode** (void \*handle, int iChannel, OutputMode eMode);

说明： 设置DIO卡指定输出通道模式

参数： handle资源设备对应指针

iChannel输出通道号，范围0~23

eMode输出模式，主要有电平模式，PWM模式以及BIT数据流模式

返回值：正常为0

备注：

int **GetDoMode** (void \*handle,

unsigned int\* pHiMode,

unsigned int\* pLoMode);

说明： 获取所有输出通道模式

参数： handle资源设备对应指针

pHiMode 通道16~23输出模式指针，如获取0x01，表示输出通道16模式为电平模式

pLoMode 通道0~15输出模式指针 如获取0x02,表示输出通道0模式为PWM模式

返回值：正常为0

备注：

int **SetDiReferenceVoltage** (void \*handle,

int iChannel,

int iVoltage);;

说明： 设置输入通道比较参考电压

参数： handle资源设备对应指针

iChannel通道号，范围0~31，0~7为一组比较电压参考值，8~15为一组比较电压参考值，16~23为一组比较电压参考值，24~31为一组比较电压参考值，如输入通道参数1与7，其实下发的是同一组比较电压值

iVoltage参考电压，单位毫伏，数值在0~25V之间

返回值：正常为0

备注：

int **GetDiLevel** (void \*handle, int iChannel, int\* iVoltage);

说明： 获取指定输入通道电平值

参数： handle资源设备对应指针

iChannel通道号

iVoltage输入电平，0为低，1为高

返回值：正常为0

备注：

int **GetDiAllLevel** (void \*handle, int\* iLevel);

说明： 获取所有输入通道输入电平

参数： handle资源设备对应指针

iLevel所有通道的输入电平，通道按位表示，0bit位为0通道，1为高电平，0为低电平，如获取0x01表示0通道为高电平输入，其余31个通道为低电平输入

返回值：正常为0

备注：

int **SetDoLevelConfigure** (void \*handle,

int iChannel,

DOLevelConfigure stDOLevelCfg);

说明： 设置指定输出通道电平配置

参数： handle资源设备对应指针

iChannel输出通道号，0~23

stDOLevelCfg输出电平配置结构

返回值：正常为0

备注：

int **SetDoLevel** (void \*handle, int iChannel, VoltageLevel eLevel);

说明： 设置指定输出通道电平值

参数： handle资源设备对应指针

iChannel通道号，范围0~23

eLevel 输出电平

返回值：正常为0

备注：

int **SetDiPWMCaptureEnable** (void \*handle,

int iChannel,

EnableStatus eStatus);

说明： 设置指定输入通道PWM捕获使能

参数： handle资源设备对应指针

iChannel通道号，范围0~31

eStatus 使能标志

返回值：正常为0

备注：

int **SetDiPWMAllCaptureEnable** (void \*handle, unsigned int iStatus);

说明： 设置所有输入通道PWM捕获使能

参数： handle资源设备对应指针

iStatus所有输入通道PWM采集使能，0bit位为0通道，1为使能，0为不使能，如0x03表示输入通道0、1PWM捕获使能，其余30个通道捕获未使能

返回值：正常为0

备注：

int **GetDiPWMAllCaptureEnable** (void \*handle, unsigned int\* pStatus);

说明： 获取所有输入通道PWM捕获使能状态

参数： handle资源设备对应指针

pStatus所有输入通道PWM采集使能状态指针，0Bit位0通道，1位使能，0为不使能，如0x01表示输入通道0为捕获使能状态，其余输入通道未捕获使能

返回值：正常为0

备注：

int **SetDoPWMConfigure** (void \*handle,

int iChannel,

DOPWMConfigure stDOPWMCfg);

说明： 设置指定输出通道PWM波形配置

参数： handle资源设备对应指针

iChannel输出通道号，范围0~23

stDOPWMCfg输出pwm配置数据结构

返回值：正常为0

备注：

int **SetDiPWMConfigure** (void \*handle,

int iChannel,

DIPWMConfigure stDIPWMCfg);

说明： 设置指定输入通道PWM波形采集设置

参数： handle资源设备对应指针

iChannel通道号，范围0~31

stDIPWMCfg输入pwm配置数据结构

返回值：正常为0

备注：

int **GetDiPWMCapture** (void \*handle, PWMProperty arrPWMProper[32]);

说明： 获取所有输入通道PWM波形采集数据

参数： handle资源设备对应指针

arrPWMProper波形数据结构体

返回值：正常为0

备注：

int **ClearDoOverProtectionStatus** (void \*handle);

说明： 清除板卡过流保护状态标志，过流指示灯恢复为绿色正常

参数： handle资源设备对应指针

返回值：正常为0

备注：

int **SetDoBitConfigure** (void \*handle,

int iChannel,

DOBitConfigure stDOBitCfg);

说明： 设置指定输出通道BIT数据流配置

参数： handle资源设备对应指针

iChannel通道号，范围0~23

stDOBitCfg数据流配置结构

返回值：正常为0

备注：

int **WriteDoBITData** (void \*handle, uint64\_t\* pArr, int iLength);

说明： 指定DIO卡发送BIT数据流

参数： handle资源设备对应指针

pArr数据流缓冲区首地址

iLength数据流缓冲区长度

返回值：正常为0

备注：数据流为uint64\_t类型数组，每uint64\_t数据，表示在指定通道配置为bit流模式下，24路输出通道bit数据同时输出。 如已配置好0、1通道输出模式为bit流模式，下发数据0x03，0x01，0x02，表示通道0输出BIT流信息110，通道1输出BIT流信息101

int **SetDoPWMDutyCalibration** (void \*handle,

int iChannel,

OutputReferenceVoltage eRef,

int8\_t iValue);

说明： 设置指定输出通道指定参考电压PWM占空比校准数据

参数： handle资源设备对应指针

iChannel通道号，范围0~23

eRef输出参考电压

iValue调整时间计数，单位纳秒

返回值：正常为0

备注：指定输出通道外接示波器，根据示波器上PWM占空比与指定输入的比较，调整时间计数，以满足输出要求

int **SetDioPpsUpdateCount**(void\* handle, uint32\_t sec);

说明： 设置PPS秒记数配置

参数： handle资源设备对应指针

sec pps秒计数值

返回值：正常为0

备注：

int **GetDioBitStreamSendTime** (void\* handle, uint64\_t \*mic\_sec);

说明： 获取BIT流发送时间戳

参数： handle资源设备对应指针

mic\_sec 时间戳微秒计数值

返回值：正常为0

备注：

int **GetDioPpsSecCount** (void\* handle, uint32\_t \*sec);

说明： 获取当前pps秒计数值

参数： handle资源设备对应指针

sec pps秒计数值

返回值：正常为0

备注：

* 1. 工作流程操作
     1. 设备初始化与反初始化



1

图 1设备初始化流程图

·

图 2设备反初始化流程图

* + 1. DIO电平输出

设置输出电平模式

SetDoMode

设置输出电平配置参数

，

包括阻抗和电压参考

SetDoLevelConfigure

设置输出电平高低数值

SetDoLevel

设置使能输出

SetDoChannelEnable

开始

结束

图 3电平输出流程图

* + 1. DIO电平输入

设置输入比较参考电压

SetDiReferenceVoltage

开始

结束

获取通道输入电平信号数值

getDiLevel

或

getDiAllLevel

成功

？

Y

N

图 4电平输入流程图

* + 1. DIO PWM波形输入

设置输入比较参考电压

SetDiReferenceVoltage

设置输入

PWM

配置参数

SetDiPWMConfigure

开始

结束

设置输入

PWM

捕获使能

SetDiPWMAllCaptureEnable

获取输入

PWM

波形信息

GetDiPWMCapture

成功

？

Y

N

图 5PWM输入流程图

* + 1. DIO PWM波形输出

设置输出

PWM

模式

setDoMode

设置输出

PWM

配置参

数

，

包括阻抗和电压参

考

，

占空比

，

频率等

setDoPWMConfigure

设置使能输出

setDoChannelEnable

开始

结束

图 6PWM输出流程图

* + 1. DIO Bit流信息输出

设置输出

BIT

流模式

setDoMode

设置输出

BIT

流配置参

数

，

包括阻抗等

setDoBitConfigure

设置输出

BIT

流数据

writeDoBITData

开始

结束

图 7BIT流输出流程图

* 1. PWM输出占空比校准

1. 配置DIO PWM波形输出
2. 通过配置校准通道数据调整占空比
3. 测量输出波形是否与下发占空比一致，不然，重复1、2步骤

设置输出

PWM

模式

setDoMode

设置输出

PWM

配置参

数

，

包括阻抗和电压参

考

，

占空比

，

频率等

setDoPWMConfigure

设置使能输出

setDoChannelEnable

开始

结束

设置指定通道目标电压校准

数据

setDoPWMDutyCalibration

观察占空比是否匹配

指定参数

？

Y

N

图 8PWM占空比校准流程图

1. AIO卡
   1. 参数类型枚举

typedef enum \_PGIACONFIG

{

PGIA\_24\_48V, 24.48v

PGIA\_0\_64V, 0.64v

PGIA\_1\_28V, 1.28v

PGIA\_2\_56V, 2.56v

PGIA\_5\_12V, 5.12v

PGIA\_10\_24V 10.24v

} PGIACONFIG;

说明：增益配置枚举，电压输入档位

typedef enum \_WAVEFILE{

WAVE1, 查询波形一文件状态

WAVE2, 查询波形二文件状态

WAVE3, 查询波形三文件状态

WAVE4 查询波形四文件状态

} WAVEFILE;

说明：波形文件选择枚举

typedef struct **\_INPUTINFO**{

float pgia;

} INPUTINFO;

说明：输入采集信息

* 1. 函数接口

int **OpenAIO**(const char\* strDev, void\*\* handle);

说明： 打开AIO对应设备文件，获取对应设备文件指针

参数： strDev AIO对应设备文件，如“/dev/aio\_3\_in”，表示打卡3槽AIO输入部分

handle对应设备文件指针

返回值：正常为0，失败返回-1

备注：

int **GetAIOVersion**(void\* handle, char\* pStr, int iLength,

int\* pActLength);

说明： 获取当前AIO卡逻辑固件版本信息

参数： handle资源设备对应指针

pStr日期版本字符串首地址

iLength用户申请的字节数组长度

pActLength实际有效字节数组长度

返回值：正常为0，失败返回-1

备注：

int **CloseAIO**(void\* handle);;

说明： 关闭AIO对应设备文件

参数： handle资源设备对应指针

返回值：正常为0，失败返回-1

备注：

int **SetAIOInputLevelConfig**(void\* handle, PGIACONFIG pgia,

int levelDelay,int levelAvg);

说明： 设置AIO卡电平输入电压档位配置

参数： handle资源设备对应指针

pgia 增益挡位，选用枚举值

返回值：正常为0

备注： 使用时选择适配输入电压的增益挡位，以提高采集精度

int **GetAIOInputLevel**(void\* handle, int \* arr);

说明： 读取AIO卡所有输入通道电压值 (单位mA)

参数： handle资源设备对应指针

arr 存放32路采集电压值数组

返回值: 正常返回0

int **SetAIOInputWaveFormConfig**(void\* handle, PGIACONFIG pgia, uint32\_t channel);;

说明： 设置指定输入通道波形采集增益配置

参数： handle资源设备对应指针

pgia增益挡位，选用枚举值，

channel 要配置的通道号

返回值:正常返回0

备注：

int **SetAIOInputWaveFormFileName**(void\* handle, char\* FilePath, uint32\_t length);

说明： 根据输入增益配置参数，采集输入波形至指定文件保存

参数： handle资源设备对应指针

FilePath 预存储的全路径波形文件名，如/home/wave.bin

length 预存储的波形文件长度

返回值：返回-1打开文件操作失败

返回-2写文件操作失败

返回0正常

备注：根据**SetAIOInputWaveFormConfig**函数下发的通道和增益档位，采集对应的输入通道数据，并保存文件。

int **GetAIOInputWaveFormStatus**(void\* handle, WAVEFILE buf\_fd, uint32\_t\* buf\_ddr\_len, uint32\_t\* buf\_total\_read);

说明： 查询波形文件和缓冲区当前的数据大小状态

参数： handle资源设备对应指针

buf\_fd 对应通道文件标识，选用枚举值，参数保留，下发WAVE1即可

buf\_ddr\_len DDR缓冲区长度

buf\_total\_read 当前文件已写入大小

返回值;正常返回0

备注：

int **SetAIOOutputLevel**(void\* handle, uint32\_t channel, uint32\_t vol);

说明：设置指定输出通道电压值

参数：handle资源设备对应指针

channel 输出通道号（0-31）

vol 输出电压（单位mV）

返回值：正常返回0

备注：

int **SetAIOOutputEnable**(void\* handle, int is\_en, int dac\_num);

说明：设置指定输出通道组使能

参数：handle资源设备对应指针

dac\_num输出通道组 0表示0~7通道，1表示8~15通道，2表示16~23通道，3表示24~31通道

is\_en使能1，不使能0

返回值：正常返回0

备注：该接口用于输出过流后，设备处于保护状态时，即设备指示灯显示为过流保护状态，若须重新使能输出接口，则须先清除过流保护状态**ClearOverCurrentProtectStatus**，然后使用此接口使能输出通道组，最后配置下发输出的具体电压数值

int **ClearOverCurrentProtectStatus**(void\* handle);

说明： 清除AIO卡过流状态，并恢复过流指示灯正常状态显示

参数： handle资源设备对应指针

返回值：正常返回0

备注：

int **SetAioPpsUpdateCount**(void\* handle, uint32\_t sec);

说明： 设置PPS秒记数配置

参数： handle资源设备对应指针

sec pps秒计数值

返回值：正常为0

备注：

int **GetAioWaveCaptureTime**(void\* handle, uint64\_t \*mic\_sec);

说明： 获取波形采集时间戳

参数： handle资源设备对应指针

mic\_sec 时间戳微秒计数值

返回值：正常为0

备注：

int **GetAioPpsSecCount**(void\* handle, uint32\_t \*sec);

说明： 获取当前pps秒计数值

参数： handle资源设备对应指针

sec pps秒计数值

返回值：正常为0

备注：

* 1. 函数接口操作流程
     1. 电平输入



图 9 电平输入

* + 1. 波形输入



图 10 波形输入

* + 1. 电平输出



图 11 电平输出

1. Audio卡
   1. 参数类型枚举

typedef enum \_SAMPLING\_RATE

{

F48kHz, 采样频率48000Hz

F96kHz, 采样频率96000Hz

F192kHz 采样频率192000Hz

} SAMPLING\_RATE;

说明：音频输入采样率

typedef enum **\_A2BRATE**{

R48k\_400KHZ, 音频帧速48k IIC数据速率400k

R48k\_100KHZ, 音频帧速48k IIC数据速率100k

R44\_1k\_400KHZ, 音频帧速44.1k IIC数据速率400k

R44\_1k\_100KHZ 音频帧速44.1k IIC数据速率100k

}A2BRATE;

说明：A2B速率相关配置

typedef enum **\_A2BTXMODE**{

SINGLE,

MASTERT,

SLAVE0,

SLAVE1

}A2BTXMODE;

说明：A2B传输模式

typedef enum **\_ERRORTYPE**{

OPEN\_ERR = -1,

WRITE\_ERR = -2,

READ\_ERR = -3,

DDR\_EMPTY\_ERR = -4,

INPUT\_MODE\_ERR = -5,

OUTPUT\_MODE\_ERR = -6,

AWAKEN\_ERR = -7,

A2B\_CONFIG\_ERR = -8,

} ERRORTYPE;

说明：错误码标识

* 1. 参数类型结构体

typedef struct **\_**channelInfo{

int mode = 0; 模式单声道0立体声1

int atten = 0; 固定衰减1/5 0 无1

int sampling; 采样率

} channelInfo;  
说明：获取通道信息

typedef struct **\_InputArg**{

char\* fileName; 文件名

int length; 采集数据大小

float in\_coef; 增益 默认1.0

int IS\_IEPE; 是否使用麦克风1使用0不使用

} InputArg;

说明：音频输入参数配置

typedef struct **\_OutputArg**{

char \* fileName; 文件名

uint32\_t regData; 固定数值，下发0X200F

float out\_coef; 输出增益 1.0

SAMPLING\_RATE rate; 输出采样率

} OutputArg;

说明：音频输出参数配置

* 1. 函数接口

int **OpenAudio**(const char\* strDev, void\*\* handle);;

说明：打开Audio对应设备文件，获取对应设备文件指针

参数：strDev Audio对应设备文件，如“/dev/audio\_3\_in1”，表示打卡3槽Audio输入1通道部分

handle对应设备文件指针

返回值：正常为0

备注：

int **GetAudioVersion**(void\* handle, char\* pStr, int iLength, int\* pActLength);;

说明：获取当前Audio卡逻辑固件版本信息

参数：handle对应设备文件指针

pStr日期版本字符串首地址

iLength用户申请的字节数组长度

pActLength实际有效字节数组长度

返回值：正常为0

备注：

int **CloseAudio**(void\* handle);

说明：关闭Audio对应设备文件

参数：handle对应设备文件指针

返回值：正常为0

备注：

int **SetAudioInputAtten**(void\* handle, int atten);

说明：设置指定输入通道音频衰减

参数: handle对应设备文件指针

atten固定衰减 0 1/5衰减；1 无衰减

返回值：正常为0

备注：

int **SetAudioInputRate**(void\* handle, SAMPLING\_RATE samp);

说明：设置音频卡音频输入的采样率

参数：handle对应设备文件指针

samp 采样率（选用枚举值）

返回值：正常返回0

备注：所有输入通道共用同一采样率

int **SetAudioInputMode**(void \*handle, int mode);

说明：设置音频卡指定输入通道音频采集模式

参数：handle对应设备文件指针

mode 模式 0单声道 1 立体声

返回值：正常返回0

备注

int **SetAudioInputEnable**(void \*handle, int en);

说明：设置音频卡指定输入通道音频采集使能

参数：handle对应设备文件指针

en 使能 0 禁止输入 1 使能输入

返回值：正常返回0

备注：接口保留，目前未使用，采集文件配置**SetAudioInputFile**下发时，已内部设置采集使能

int **SetAudioInputFile**(void \*handle, InputArg \*iArg);

说明：设置音频卡音频采集的文件名和采集的数据大小

参数：handle对应设备文件指针

iArg输入采集配置指针

返回值：-1 打开/创建filename失败

-2 数据写入文件失败

-3 生成wave文件失败

0 操作成功

备注：

int **GetAudioInputStatus**(void \*handle, int\*fileSize, int\* total\_len);

说明：查询音频采集文件读写数据状态和缓冲区数据大小状态

参数：handle对应设备文件指针

filesize 当前采集文件大小

total\_len缓冲区的数据长度

返回值：正常返回0

备注：

int **GetAudioInputChannelInfo**(void \*handle, channelInfo \*info);

说明：查询音频卡指定输入通道配置信息

参数：handle对应设备文件指针

info 通道信息，包括模式，衰减，采样率

返回值：正常返回0

备注：

int **SetAudioOutputEnable**(void \*handle, int en);

说明：设置音频卡指定输出通道使能

参数：handle对应设备文件指针

en 使能 0 禁止输出 1 使能输出

返回值：正常返回0

备注：接口保留，目前未使用，播放文件配置**setAudioOutputFile**下发时，已内部设置播放使能

int **SetAudioOutputMode**(void \*handle, int mode);

说明：设置音频卡指定输出通道模式

参数：handle对应设备文件指针

mode 模式 0单声道 1 立体声

返回值：正常返回0

备注

int **SetAudioOutputAMP**(void \*handle, int amp);

说明：设置音频卡指定输出通道固定增益

参数: handle对应设备文件指针

amp衰减 0 无增益；1 5倍增益

返回值：正常为0

备注：

int **SetAudioOutputFile**(void \*handle, OutputArg \*iArg);

说明：设置音频卡指定输出通道播放文件和采样率配置

参数：handle对应设备文件指针

filenNme 要输出的文件和路径（如：/home/data.wav）

iArg输出播放配置指针

返回值：-1 打开fileName失败

-2 文件读取数据失败

0 操作成功

备注：

int **SetLoopPlay**(void \*handle, int isLoop);

说明：设置音频卡指定输出通道是否循环播放

参数：handle对应设备文件指针

isLoop循环 0不循环 1 循环

返回值：正常返回0

备注：当循环时，下发参数0不循环，则播放完当前音频后禁止播放

int **A2BBoardMode**(void \*handle, int mode);

说明：设置A2B工作模式

参数：handle对应设备文件指针

mode 模式 0从模式 1 主模式

返回值：正常返回0

备注

int **A2BTXMode**(void \*handle, A2BTXMODE mode, A2BRATE mRate);

说明：设置A2B输出模式

参数：handle对应设备文件指针

Mode发送模式，保留，默认填SINGLE

mRate速率

返回值：正常返回0

备注

int **setA2BRecv**(void \*handle, char \* fileName, int length, float coef);

说明：A2B根据配置接收数据并保存为指定文件

参数：handle对应设备文件指针

fileName接收数据保存文件名

length 保存文件大小

coef接收数据幅度增益

返回值：正常返回0

备注

int **setA2BSend**(void \*handle, char \* fileName, SAMPLING\_RATE rate,

float coef);

说明：A2B根据配置打开指定文件，并发送数据

参数：handle对应设备文件指针

fileName 发送数据文件名称

rate输出采样率，当此A2B为主设备时，方生效

coef 发送数据幅度增益

返回值：正常返回0

备注

* 1. 函数接口操作流程
     1. 音频采集



图 12 音频采集

* + 1. 音频输出



图 13 音频输出

* + 1. A2B配置及数据传输



图 14A2B模式配置以及数据传输