

# 启英泰伦离在线应用对接说明

## 1. 语音芯片和 WiFi 交互说明

### ● 语音芯片上传音频数据参数：

采样率：16K 声道：单声道 采样精度:16bit

### ● 语音芯片上传 PCM 数据压缩格式为 SPEEX/OPUS：

#### ➤ speex 格式说明：

- ✧ 音频数据为 10ms 一帧，单帧音频数据 160 个点(short 类型)；两帧压缩为 1 帧，640 字节被编码为 43 字节，14 倍压缩率(第一个字节为 2A，表示数据长度)，再加串口通信协议头 16 个字节上传(共 59 字节)；如果大模型不支持 speex 格式数据，云端收到 speex 格式编码后的音频数据需要通过 speex 解码成 pcm 格式的原始音频数据再送个大模型识别。
- ✧ speex 参数配置可参考离在线 SDK 中 `audio_speex_task_init ( )` 函数

#### ➤ opus 格式说明：

- ✧ 音频数据为 10ms 一帧，单帧音频数据 160 个点(short 类型)；4 帧压缩为 1 帧，1280 字节被编码为 80 字节，16 倍压缩率，再加串口通信协议头 16 个字节上传(共 96 字节)；如果大模型不支持 opus 格式数据，云端收到 opus 格式编码后的音频数据需要通过 opusj 解码成 pcm 格式的原始音频数据再送个大模型识别。
- ✧ opus 参数配置可参考离在线 SDK 中 `audio_opus_task_init ( )` 函数

### ● 语音芯片当前只支持 mp3 格式音频数据播放

- ✧ wifi 发开始播放音频指令，语音芯片再根据空闲内存大小向 WiFi 请求播放数据(默认单次请求 1KB)；避免语音芯片内存不够导致音频播放数据丢失，语音芯片不接收 WiFi 主动发的音频播放数据。
- ✧ wifi 端可以通过指令控制语音芯片播放开始和播放结束
- ✧ wifi 端和云端通信获取播放数据也建议用请求播放数据的方式，避免云端发送太快 WiFi 内存不够丢数据

## ● 语音芯片 VAD 说明

- 语音芯片通过 VAD START 和 VAD END 检测算法，自动检测用户一段说话音频数据开始和结束；检测到 VAD START 开始采集音频数据并压缩后流式传输给 WiFi，检测到 VAD END 后停止传输音频，并给到 WiFi 结束上传音频指令

## ● 语音芯片和 WiFi 通过串口通信

- 指令数据和音频数据串口可以复用，通信波特率建议为921600bit/s，最大支持 2Mbit/s

## 2. 语音芯片和 WiFi 交互协议-V1.0

### 2.1 串口命令格式概述

说明：该协议内非单字节数据输出过程中，采用**低字节在先，高字节在后**。

名称	长度(字节)	模型描述
head	4	帧标识头，固定为:0x5a5aa5a5； WiFi 端收到的数据为：a5 a5 5a 5a
checksum	2	校验和（只校验数据部分）
cmd type	2	指令类型
len	2	数据长度 n
version	2	版本信息
fill_data	4	填充数据：控制指令用（DEF_FILL） DEF_FILL = 0x12345678,

		SPEAK_FILL = 0x12345677,  INVAILD_SPEAK_FILL = 0x12345666,  MUSIC_START_FILL = 0x12345699,  MP3_FILL = 0x12345688,  M4A_FILL = 0x123456aa,  WAV_FILL = 0x123456bb,  REPEAT_FILL = 0x123456ab,
data	n byte	数据部分

## 2.2 指令类型说明：

### ➤ 语音数据相关

指令名称	指令类型	指令方向 (wifi-audio)	备注(预留当前版本指令未实现)
LOCAL_ASR_RESULT_NOTIFY	0x0101	audio->wifi	本地语音识别成功
WAKE_UP	0x0102	audio->wifi	唤醒成功
VAD_END	0x0103	audio->wifi	预留
SKIP_INVALID_SPEAK	0x0104	audio->wifi	上一次传输音频为无效
PCM_MIDDLE	0x0105	audio->wifi	音频数据
PCM_FINISH	0x0106	audio->wifi	PCM 数据结发送完成
PCM_IDLE	0x0107	audio->wifi	预留

VAD_START	0x0108	audio->wifi	检测到 vad start
EXIT_WAKE_UP	0x0109	audio->wifi	退出唤醒
SET_VAD_SENSITIVITY	0x010A	wifi->audio	设置 VAD 灵敏度 (45db-60db), 带参数 1 字节, 默认值 53
VAD_START_BY_KEY	0x010B	wifi->audio	预留(手动触发 vad start)
VAD_END_BY_KEY	0x010C	wifi->audio	预留(手动触发 vad end)
SET_AUDIO_EXIT_WAKE_UP	0x010D	wifi->audio	设置语音退出唤醒, 带参数 1 字节 1-带退出提示音(默认) 0-不带退出唤醒提示音
PCM_DENOISE_ENABLE	0x010E	wifi->audio	设置上传音频是否带功能降噪, 带参数 1 字节 1-带降噪(默认) 0-不带降噪
SET_VAD_FILTER_FRAME	0x010F	wifi->audio	设置过滤 vad start 和 end 之间帧数(10ms 一帧), 小于该设定值被判断为无效语音, 语音芯片不上传, 带参数 1 字节 (范围(15-40), 默认 25)
SET_VAD_SENSITIVITY_ACTIVATE_LENGTH	0x0110	wifi->audio	设置多长时间的静音产生 vad end, 带参数 1 字节(默认 20), 参考值设置: //允许语音停顿的时间; 6:约 300ms; 15:约 400ms; 20:约 500ms; 50:约 1S 80:约 1.5S 120:约 2S (如果有背景噪声没有消除干净, 延时时间会变长, 不支持设置其他值)
SET_VAD_START_MAX_TIMEOUT	0x0111	wifi->audio	设置 vad start 最大时间, 超过设置时间则强制结束上传语音, 单位秒, 默认 5S, 带参

			数 2 字节
SET_PLAY_VOICE_ID	0x0112	wifi->audio	设置播放离线播报音，带参数 3 字节(前 2 个字节表示 id 号，第三个字节表示要不要打断当前播报 1-打断 0-不中断)
SET_WAKE_UP_CONTINUE_TIME	0x0113	wifi->audio	设置持续唤醒时间，超过设置时间，语音芯片强制退出唤醒，带参数 2 字节(默认 15S)
SET_ENTER_WAKE_UP	0x0114	wifi->audio	通过指令让语音芯片进入唤醒状态，不用等待唤醒词唤醒，带参数 1 字节 1-带进入唤醒提示音(默认) 0-不带进入唤醒提示音
SET_INTERACTION_MULTI_ROUND_ENABLE	0x0115	wifi->audio	设置单轮还是多轮，带参数 1 字节，0-单轮 1-多轮(默认)
UPLOAD_PLAY_FULL_DUPLEX_ENABLE	0x0116	wifi->audio	设置播放音频时是否上传录音，带参数 1 字节 0-不上传(默认) 1-上传
SET_AUDIO_VOLUME	0x0117	wifi->audio	设置语音芯片音量，带参数 1 字节，范围(1-7)，默认 7
SET_AUDIO_COMPRESS_TYPE	0x0118	wifi->audio	预留
SET_VOLUME_MUTE_STATE	0x0119	wifi->audio	设置语音 mute 声音参数 1 字节，状态 1-设置为静音 0-设置为非静音
SET_AUDIO_START_RECORD	0x011A	wifi->audio	通过指令开始录音
SET_AUDIO_STOP_RECORD	0x011B	wifi->audio	通过指令结束录音(和 SET_AUDIO_START_RECORD 必须配对使用)
SET_CLOUD_ANS	0x011C	wifi->audio	设置退出云端响应超时

_TIMEOUT_EXIT			
---------------	--	--	--

## ➤ 网络播放指令

指令名称	指令类型	指令方向 (wifi-audio)	备注(预留指令未实现)
NET_PLAY_START	0x0201	wifi->audio	开始播放网络音频指令
NET_PLAY_PAUSE	0x0202	wifi->audio	预留(播放暂停)
NET_PLAY_RESUME	0x0203	wifi->audio	预留(恢复播放)
NET_PLAY_STOP	0x0204	wifi->audio	停止播放网络音频
NET_PLAY_RESTART	0x0205	wifi->audio	预留(重新播放)
NET_PLAY_NEXT	0x0206	wifi->audio	预留(播放下一首)
NET_PLAY_LOCAL_TTS	0x0207	wifi->audio	预留(播放本地 TTS)
NET_PLAY_END	0x0208	wifi->audio	播放数据发送完成
NET_PLAY_RECONNECT_URL	0x0209	wifi->audio	预留(重新获取连接)
PLAY_DATA_GET	0x020a	audio->wifi	语音芯片获取后续播放数据指令
PLAY_DATA_RECV	0x020b	wifi->audio	WIFI 发送播放音频数据到语音芯片
PLAY_DATA_END	0x020c	wifi->audio	wifi 播放数据发送完成
PLAY_TTS_END	0x020d	audio->wifi	音频播放结束
PLAY_EMPTY	0x020e	wifi->audio	预留(播放空指令)

PLAY_NEXT	0x020f	wifi->audio	预留(播放完上一首,主动播放下一首)
PLAYING_TTS	0x0210	wifi->audio	预留(当前正在播放 TTS 音频)
PLAY_RESUME_ERROR	0x0211	wifi->audio	预留(重播失败)
PLAY_LAST	0x0212	wifi->audio	预留(播放上一首)
PLAY_AUDIO_SIZE	0x0213	wifi->audio	预留(播放音频数据总长度)
PLAY_AUDIO_TYPE	0x0214	wifi->audio	预留(播放数据类型)
SET_AUDIO_PLAY_MODE	0x0215	wifi->audio	预留(设置播放模式,带参数 1 字节,1-打断当前播报 0-不 打断当前播报,顺序播放默认))
VAD_START_STOP_PLAY	0x0216	wifi->audio	设置全双工模式下, vad start 是否停止当前播放,带参数 1 字节,1-停止播放 0-不停止播放)
LOCAL_AUDIO_PLAY_START	0x0217	wifi->audio	预留(本地播放音频开始)
LOCAL_AUDIO_PLAY_STOP	0x0218	wifi->audio	预留(本地播放音频开始)

### ➤ 配网网络/网络状态指令

指令名称	指令类型	指令方向 (wifi-audio)	备注(预留指令未实现)
ENTER_NET_CONFIG	0x0401	wifi->audio	预留(进入配网模式)
NET_CONFIGING	0x0402	wifi->audio	预留(配网中)

EXIT_NET_CONFIG	0x0403	wifi->audio	预留(退出配网模式)
INIT_SMARTCONFIG	0x0404	wifi->audio	预留(初始密码状态 出厂配置状态)
WIFI_DISCONNECTED	0x0405	wifi->audio	预留(网络断开,带参数 3 字节(前 2 个字节表示 id 号, 第三个字节表示要不要打断当前播报 1-打断 0-不中断))
WIFI_CONNECTED	0x0406	wifi->audio	网络连接成功,带参数 3 字节(前 2 个字节表示 id 号, 第三个字节表示要不要打断当前播报 1-打断 0-不中断)
GET_PROFILE	0x0407	wifi->audio	预留(已获取鉴权文件)
NEED_PROFILE	0x0408	wifi->audio	预留(需要鉴权文件)
CLOUD_CONNECTED	0x0409	wifi->audio	云端已连接,带参数 3 字节(前 2 个字节表示 id 号, 第三个字节表示要不要打断当前播报 1-打断 0-不中断)
CLOUD_DISCONNECTED	0x040a	wifi->audio	预留(云端已断开,带参数 3 字节(前 2 个字节表示 id 号, 第三个字节表示要不要打断当前播报 1-打断 0-不中断))
NET_CONFIG_SUCCESS	0x040b	wifi->audio	预留(配网成功,带参数 3 字节(前 2 个字节表示 id 号, 第三个字节表示要不要打断当前播报 1-打断 0-不中断))
NET_CONFIG_FAIL	0x040c	wifi->audio	预留(配网失败,带参数 3 字节(前 2 个字节表示 id 号, 第三个字节表示要不要打断当前播报 1-打断 0-不中断))
NET_CONFIG_CLEAR	0x040d	wifi->audio	预留(配网信息清除)

AN			
----	--	--	--

### ➤ OTA/生产测试/红外指令

指令名称	指令类型	指令方向 (wifi-audio)	备注(预留指令未实现)
CIAS_OTA_START	0x0501	wifi->audio	开始 OTA
CIAS_OTA_DATA	0x0502	wifi->audio	OTA 数据
CIAS_OTA_SUCES S	0x0503	audio->wifif	OTA 升级成功
CIAS_FACTORY_S TART	0x0504	wifi->audio	开始生产测试,带参数 1 字节表示是否需要语音芯片实时上传检测到的音频能量值(0x0-不上传(默认) 0x01-上传)
CIAS_FACTORY_O K	0x0505	audio->wifi	预留(生产测试成功)
CIAS_FACTORY_F AIL	0x0506	audio->wifi	预留(生产测试失败)
CIAS_FACTORY_S ELF_TEST_START	0x0507	wifi->audio	预留(开始生产自测试)
CIAS_IR_DATA	0x0508	wifi->audio	预留(红外数据发送)
CIAS_IR_LOADING _DATA	0x0509	wifi->audio	预留(红外码库下载中)
CIAS_IR_LOAD_DA TA_OVER	0x050a	wifi->audio	预留(红外码库下载完成)
CIAS_IR_LOAD_DA TA_START	0x050b	wifi->audio	预留(红外下载码库开始)
CIAS_FACTORY_T EST_ENG_THR_S ET	0x050c	wifi->audio	音频通路设置检测播报音能量阈值设置,范围(0-100db),sdk 默认设置的 50db,带参数

			1 字节表示要设置的能量阈值
CIAS_FACTORY_T EST_ENG_GET	0x050d	audio->wifi	生产测试结果通过该指令返回，0x01-测试通过 0x02-测试失败
CIAS_FACTORY_T EST_REAL_VAL_G ET	0x050e	audio->wifi	生产测试过程中实时值上传； 生产测试开始以后，如果 CIAS_FACTORY_START 参 数设置为 1，则语音芯片主动 上传检测到的能量值

### ➤ 语音芯片系统状态相关

指令名称	指令类型	指令方向 (wifi-audio)	备注(预留指令未实现)
CIAS_AUDIO_SYS _READY	0x0601	audio->wifi	语音芯片系统 ready
CIAS_AUDIO_SYS _ERR	0x0602	audio->wifi	语音芯片系统异常
CIAS_AUDIO_RST	0x0603	wifi->audio	复位语音芯片

### ➤ 自学习指令

指令名称	指令类型	指令方向 (wifi-audio)	备注(预留指令未实现)
CWSL_UART_REG ISTRATION_WAKE	0x0701	wifi->audio	串口进入学习唤醒词
CWSL_UART_REG ISTRATION_WAKE _ING	0x0702	audio->wifi	语音芯片返回正在学习唤醒 词状态中
CWSL_UART_REG ISTRATION_WAKE _END_SUCCESSF UL	0x0703	audio->wifi	语音芯片返回学习唤醒词成 功

CWSL_UART_REGISTRATION_WAKE_END_FAILED	0x0704	audio->wifi	语音芯片返回学习唤醒词失败
CWSL_UART_EXIT_REGISTRATION	0x0705	wifi->audio	串口指令退出学习
CWSL_UART_DELETE_WAKE	0x0706	wifi->audio	串口指令删除唤醒词

### ➤ 指令执行状态相关

指令名称	指令类型	指令方向 (wifi-audio)	备注(预留指令未实现)
CIAS_CMD_EXECUTE_STATE	0x0801	wifi->audio	指令执行状态, 返回数据部分3字节 前两字节表示语音芯片接收到指令的类型, 第3字节表示指令执行结果(0x01-执行成功 0x02-执行失败)

## 2.3 串口数据解析参考代码：

```
#define CIAS_STANDARD_MAGIC 0x5a5aa5a5
#define CIAS_SEND_MSG_BUF_LEN 1024 + 16
```

- 帧头定义：

```
typedef struct
{
    unsigned int magic; /* Start data for frame. Let's define as 0x5a5aa5a5; */
    unsigned short checksum; /* checksum */
    unsigned short type; /* Define command type. */
    unsigned short len; /* Define data stream len. */
    unsigned short version; /*版本信息*/
    unsigned int fill_data; /*填充数据*/
}cias_standard_head_t;
```

- **数据部分定义：**

```
#pragma pack(1)
typedef struct
{
    unsigned char data[CIAS_SEND_MSG_BUF_LEN];
    //unsigned char *data;
    unsigned short length;
    unsigned short type;
    unsigned int ack_flag; /*需要 ACK*/
}cias_send_msg_t;
#pragma pack()
```

- **组包函数:**

```
static int cias_init_send_msg_header(unsigned char *input, unsigned char size,
unsigned short len, unsigned short type, unsigned int fill_data, unsigned short version)
{
    memset(input, 0x00, size);
    if (input == NULL)
    {
        return -1;
    }
    cias_standard_head_t *head = (cias_standard_head_t *)input;
    head->magic = CIAS_STANDARD_MAGIC;
    head->type = type;
    head->len = len;
    head->version = version;
    head->fill_data = fill_data;
    return 0;
}
/**
```

\* 将 wifi 数据打包发送给 audio 进行处理，打包数据后发送到

```

cias_send_slave_msg_queue 消息队列
*/

void cias_message_send_interface(unsigned short cmd, cias_fill_type_t type, int len,
unsigned char *buf)
{
    cias_send_msg_t send_msg;
    int32_t ret = 0;
    unsigned int fill_data = type;
    send_msg.type = cmd;
    cias_init_send_msg_header(send_msg.data, 16, len, send_msg.type, fill_data,
0x00);
    if (len > 0 && (len <= CIAS_SEND_MSG_BUF_LEN))
    {
        memcpy(send_msg.data + 16, buf, len);
        // CIAS_LOG_ERR("memcpy send_msg.data buf(%d)\r\n", *buf);
    }
    send_msg.length = 16 + len;
    // 发送 to audio 侧数据
    if ( cias_queue_send( &cias_send_slave_msg_queue,    &send_msg,
pdMS_TO_TICKS(100)) != CIAS_OK)
    {
        CIAS_LOG_ERR("sizeof(send_msg) send fail...\r\n");
    }
}

```

## 2.4 API 使用实例：

- WiFi 联网成功发送指令到语音端：

```

cias_message_send_interface(CLOUD_CONNECTED, DEF_FILL, 0, NULL);

```

- WiFi 收到语音端 PLAY\_DATA\_GET 指令后返回音频数据：

```

cias_message_send_interface(PLAY_DATA_RECV, DEF_FILL, 4096, data)

```

### 3.在线方案生产测试

#### 3.1 测试播放及录音硬件回路，使用如下指令：

- 设置音频能量阈值，通路检测超过该值则表示测试成功：

A5A55A5A00000c0501000001000000003C(0x3C-60db, sdk 默认设置的 50db, 用户可以根据需求设置)

- 设置开始测试且不带中间能量数据上传(开始测试以后 3S 内播放测试音频，否则会退出测试流程，返回测试失败)：

A5A55A5A00000405000000010000000000

- 播放测试音频编号为：[56789]产测专用 1khz 音频，禁止删除.mp3

A5A55A5A000012010300000100000000D5DD00

#### 4.详细指令说明：

注意：需要重点关注的部分：

黄色部分-指令类型 蓝色部分-数据部分 校验和-语音芯片端暂未开启校验

测试指令说明(括号中为当前指令设置的状态)	指令 ID	指令
设置音量指令(设置为 7)	0x0117	A5A55A5A00001701010000010000000007
强制开始上传音频	0x011A	A5A55A5A00001A010000000100000000
强制结束上传音频	0x011B	A5A55A5A00001B010000000100000000
VAD 灵敏度设置(设置为 45db)	0x010A	A5A55A5A00000A0101000001320000002D
设置语音退出唤醒(带播报音)	0x010D	A5A55A5A00000D01010000011E00000001
设置上传音频是否带降噪(设置为带降噪)	0x010E	A5A55A5A00000E01010000011E00000001
播放指定播报音音频(播放 id 为 2060 的音频，带打断功能)	0x0112	A5A55A5A0000120103000001000000000C0801

设置唤醒时间设置 ( 设置退出唤醒时间为 5S )	0x0113	A5A55A5A000013010200000100000000 0500
设置单轮/多轮(设置为多轮)	0x0115	A5A55A5A000015010100000100000000 01
设置播放时是否允许上传录音(设置为允许)	0x0116	A5A55A5A000016010100000100000000 01
设置 vad start 最长时间 (设置为 5S)	0x0111	A5A55A5A000011010200000100000000 0500
设置强制进入唤醒状态 (带唤醒播报)	0x0114	A5A55A5A000014010100000100000000 01
设置 vad 多长时间静音产生 vad end ( 设置为 0x14 , 对应 500ms )	0x0110	A5A55A5A000010010100000100000000 14
设置过滤 vad start 和 end 之间帧数(设置为过滤 25 帧)	0x010F	A5A55A5A00000F01010000011E000000 19
设置语音 mute 声音(设置为静音)	0x0119	A5A55A5A00001901010000011E000000 01
开始播放	0x0201	A5A55A5A000001020100000100785634 12
播放结束	0x020C	A5A55A5A00000C020100000100785634 12
立即播放停止	0x0204	A5A55A5A000004020100000100785634 12
生产测试开始(设置语音芯片上传检测过程中的能量值)	0x0504	A5A55A5A000004050100000100000000 01
设置生产通路测试阈值 , 能量超过 45db 才表示硬件播放录音通路正常	0x050c	A5A55A5A00000c050100000100000000 2d

设置退出云端响应超时	0x011C	A5A55A5A00001C010000000100000000
开始学习唤醒词	0x0701	A5A55A5A000001070000000100000000
退出学习	0x0705	A5A55A5A000005070000000100000000
删除学习的唤醒词	0x0706	A5A55A5A000006070000000100000000