WIFI故事机AI_SDK开发文档

杰理 WIFI故事机 文档

前言

本文档定义了杰理WIFI故事机的AI SDK的接入流程以及开发说明等内容。
本文档最终解释权归珠海杰理科技股份有限公司。

修订记录

修订时间	修订版本	修订人	修订描述
2018/09/14	V0.1	梁泳新	初稿

目录

1 名词解释

1.1 AI SDK

指用户自行开发或者我们已经提供的AI第三方服务平台，要求需要实现语音识别、播放资源链接返回、播放状态上报这三个基本功能，另外可实现通过语音播放控制的可选功能。

1.2 AI SERVER

通过调用server_open()打开ai_server服务后，再进行connect请求，可以实现同时打开多个已经在SDK上注册的AI第三方服务平台，所有小机和服务器之间的交互都是通过ai_server进行的。

2 AI SDK的开发流程

2.1 注册

AI SDK的注册方式如下所示：

```
REGISTER_AI_SDK(xxx_sdk_api) = {
    .name           = "xxx",                //AI第三方服务平台的名称
    .connect        = xxx_sdk_open,     //初始化和连接AI第三方服务平台
    .state_check    = xxx_sdk_check,        //检查AI第三方服务平台的连接状态
    .do_event       = xxx_sdk_do_event,     //AI第三方服务平台的对应事件处理
    .disconnect     = xxx_sdk_disconnect,   //关闭退出AI第三方服务平台
};
```

2.2 AI服务的事件处理

用户需要在xxx_sdk_do_event()将上层应用通知下来的事件通知给对应的AI第三方服务平台的服务器，若服务器暂不支持某些事件可以不做处理，具体AI服务事件列表如下：

```
enum ai_server_event {
    AI_EVENT_SPEAK_END      = 0x01,       //语音链接播放完成
    AI_EVENT_MEDIA_END      = 0x02,       //资源链接播放完成
    AI_EVENT_PLAY_PAUSE     = 0x03,       //播放暂停
    AI_EVENT_PREVIOUS_SONG  = 0x04,       //播放上一首
    AI_EVENT_NEXT_SONG      = 0x05,       //播放下一首
    AI_EVENT_VOLUME_CHANGE  = 0X06,       //设置音量具体值
    AI_EVENT_VOLUME_INCR    = 0x07,       //调大音量
    AI_EVENT_VOLUME_DECR    = 0x08,       //调小音量
    AI_EVENT_VOLUME_MUTE    = 0x09,       //设置静音
    AI_EVENT_RECORD_START   = 0x0a,       //开始录音
    AI_EVENT_RECORD_STOP    = 0x0c,       //结束录音
    AI_EVENT_VOICE_MODE     = 0x0d,       //改变语音识别模式
    AI_EVENT_PLAY_TIME      = 0x0e,       //播放时间
    AI_EVENT_MEDIA_STOP     = 0x0f,       //主动打断当前播放
    AI_EVENT_QUIT           = 0xff,      //退出AI第三方服务平台
};
```

2.2 处理AI第三方平台服务器返回事件

所有AI第三方平台服务器返回的事件都通过server_event_handler()通知上层应用处理。
播放链接：ai_server_event_url(&xxx_sdk_api, url, AI_SERVER_EVENT_URL);
播放暂停：ai_server_event_url(&xxx_sdk_api, url, AI_SERVER_EVENT_PAUSE);
播放继续：ai_server_event_url(&xxx_sdk_api, url, AI_SERVER_EVENT_CONTINUE);

3 微信语音的开发流程

微信语音封装成server的形式。

3.1 微信请求事件

类型	说明
WECHAT_STATE_OPEN	初始化wechat core 相关
WECHAT_STATE_START	设置 wechat 状态
WECHAT_STATE_ENC_AMR_START	开始编码AMR数据
WECHAT_STATE_ENC_AMR_STOP	结束编码AMR数据，和发送语音数据
WECHAT_STATE_SEND	没用
WECHAT_STATE_PAUSE	没用
WECHAT_STATE_STOP	设置 wechat 状态
WECHAT_STATE_CLOSE	关闭 wechat core

使用server开始AMR编码的例子

```
static int app_wechat_amr_enc_start()
{
    int err;
    struct wechat_req req;

    if (!__this->wechat_connected) {
        app_music_play_voice_prompt("016.mp3", __this->dec_ops->dec_breakpoint);
        return 0;
    }
    if (__this->wechat_state & 0x1) {
        return -1;
    }

    if (__this->dec_ops) {
        __this->dec_ops->dec_stop(1);
    }

    app_music_play_voice_prompt("rec.mp3", NULL);

    os_time_dly(50); /*防止按键声音*/

    req.cmd = WECHAT_STATE_ENC_AMR_START;
    err = server_request(__this->wechat_server, WECHAT_REQ, &req);
    if (err == 0) {
        __this->wechat_state |= 0x1;
    }

    return err;
}
```

3.2 微信接收语音事件

类型	说明
WECHAT_SERVER_SPEECH_URL_EVENT	接收到语音数据的URL
WECHAT_SERVER_AMR_ERR_EVENT	AMR编码异常事件返回app_core层

```
static void wechat_server_event_handler(void *priv, int argc, int *argv)
{
    char buffer[7];
    switch (argv[0]) {
    case WECHAT_SERVER_SPEECH_URL_EVENT:
        printf("WECHAT_SERVER_SPEECH_URL_EVENT\n");
        strcpy(__this->wechat_speech_url, (char *)argv[1]);
        __this->wechat_speech_read_flag = 1;
        app_music_play_voice_prompt("009.mp3", __this->dec_ops->dec_breakpoint);
        break;
    case WECHAT_SERVER_AMR_ERR_EVENT:
        printf("WECHAT_SERVER_AMR_ERR_EVENT\n");
        app_music_play_voice_prompt("019.mp3", __this->dec_ops->dec_breakpoint);
        break;

    }
}
```

3.3 微信公众号

设备触发事件，比如设备按键下一首对应事件WECHAT_NEXT_SONG。

类型	说明
WECHAT_NEXT_SONG	下一首
WECHAT_PRE_SONG	上一首
WECHAT_PAUSE_SONG	暂停
WECHAT_CONTINUE_SONG	继续
WECHAT_VOLUME_CHANGE	声音变化，没实现
WECHAT_MEDIA_END	当前播放结束

类型	说明
WECHAT_PROGRESS_INFO	进度，没实现
WECHAT_MEDIA_STOP	停止播放

公众号触发使用mqtt事件

发布者	Topic	说明	实现
system	robot/{uuid}/system/wechat/message/new	推送微信语音信息	
user	robot/{uuid}/user/music/play	播放音频文件	
user	robot/{uuid}/user/music/next	下一首	
user	robot/{uuid}/user/music/pre	上一首	
user	robot/{uuid}/user/music/pause	暂停	
user	robot/{uuid}/user/music/continue	继续播放	
user	robot/{uuid}/user/music/volume	调整音量	
user	robot/{uuid}/user/device/status	用户请求设备发送状态	
user	robot/{uuid}/user/device/update	用户请求设备进行升级	
device	robot/{uuid}/device/status/online	是否在线	
device	robot/{uuid}/device/status/quantity	当前电量及是否在充电	
device	robot/{uuid}/device/status/volume	声音音量	
device	robot/{uuid}/device/status/version	固件版本 0.0.0	
device	robot/{uuid}/device/status/play	播放内容	
device	robot/{uuid}/device/status/play_msg	播放进度信息	
device	robot/{uuid}/device/status/play_status	播放、暂停	
device	robot/{uuid}/device/status/mac	MAC地址	
device	robot/{uuid}/device/status/wifi	已连wifi名	
device	robot/{uuid}/device/status/storage	存储空间	