

启英物联小程序接入指南

应用代码编程说明

版本号：1.0

Chipintelli Confidential

Chipintelli Technology Co., Ltd.

CONFIDENTIAL ALL RIGHTS RESERVED. This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Chipintelli Technology Co., Ltd.

[点击前往启英泰伦语音 AI 平台](#)

PAGE 1 / 11

文件历史跟踪 DOCUMENT HISTORY PAGE			
版本号 Rev. NO.	发起者 Originator	描述 Description	日期 Date
1.0	启英泰伦	新建文档	2024/03/26

Chipintelli Confidential

目录

1 启英物联小程序与设备通信协议格式.....	4
1.1 帧头（2Byte）	4
1.2 协议版本（1Byte）	4
1.3 厂商 ID（1Byte）	4
1.4 设备类型（1Byte）	4
1.5 设备编号（1Byte）	4
1.6 功能类型（1Byte-高 4 位）.....	5
1.7 数据类型（1Byte-低 4 位）.....	5
1.8 功能 ID（1Byte）	5
1.9 数据长度（2Byte）	5
1.10 数据（nByte）	5
1.11 CRC16（2Byte）	5
2 SDK 应用层代码编写指导	5
2.1 设备类型与设备编号的宏定义.....	6
2.2 设备功能结构体说明.....	6
2.3 各功能类型的编程说明.....	7
2.3.1 属性设置编程说明	7
2.3.2 事件上报编程说明	8
2.3.3 设备状态查询以及应答消息编程说明.....	9
2.4 结语	10

1 启英物联小程序与设备通信协议格式

启英物联小程序与搭载了我司 CI2311/CI2312 智能语音芯片的设备进行通讯时，需要有固定的格式规范，以下对启英物联小程序和设备通信协议格式进行简要介绍。

1.1 帧头（2Byte）

帧头固定为 0xAA5A

1.2 协议版本（1Byte）

0x01 代表 V1.0，目前固定为 0x01

1.3 厂商 ID（1Byte）

目前固定为 0x01--启英泰伦

1.4 设备类型（1Byte）

0x01: 红外遥控器
0x02: 空调
0x03: 灯控
0x04: 音响设备
0x05: 茶吧机
0x06: 风扇
0x07: 取暖桌
0x08: 取暖器
0x09: 水暖毯
0x0A: 微波炉
0x0B: 窗帘
0x0C: 加湿器

1.5 设备编号（1Byte）

由具体产品项目指定

1.6 功能类型 (1Byte-高 4 位)

- 0x1: 属性设置
- 0x2: 事件上报
- 0x3: 设备状态查询
- 0x4: 应答消息

1.7 数据类型 (1Byte-低 4 位)

- 0x1: 16 进制编码数据
- 0x2: JSON 数据

1.8 功能 ID (1Byte)

根据不同产品定义功能

1.9 数据长度 (2Byte)

每包数据长度

1.10 数据 (nByte)

用户数据

1.11 CRC16 (2Byte)

整包校验

2 SDK 应用层代码编写指导

现在以 CI231X_SDK_ASR_Offline_BLE_V1.10.4 的为例，介绍如何利用此 SDK 在我司 CI2311/CI2312 智能语音芯片上实现接入启英物联小程序。

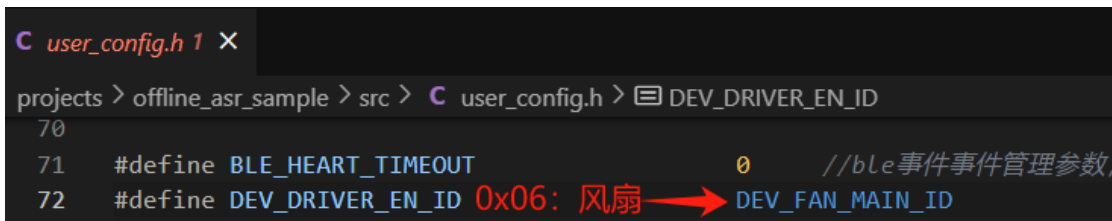
CI231X_SDK_ASR_Offline_BLE_V1.10.4 已经在底层实现了蓝牙协议栈功能，用户只需专注于应用层的代码编写，与启英物联小程序通讯的应用层代码集中在 CI231X_SDK_ASR_Offline_BLE_V1.10.4\projects\offline_asr_sample\src\demo 路径下。

Chipintelli Technology Co., Ltd.

CONFIDENTIAL ALL RIGHTS RESERVED. This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Chipintelli Technology Co., Ltd.

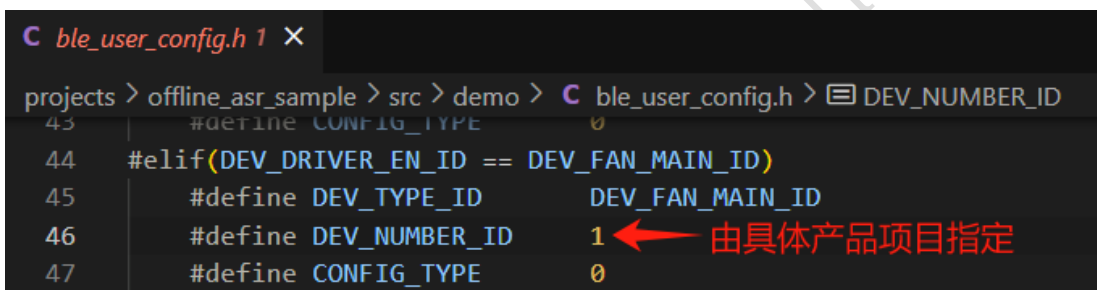
2.1 设备类型与设备编号的宏定义

协议中设备类型与设备编号需要用宏定义设置，其中设备类型在 `user_config.h` 中的 `DEV_DRIVER_EN_ID` 宏的值来决定。设备编号在 `ble_user_config.h` 中的 `DEV_NUMBER_ID` 宏的值来决定。以下以风扇为例说明：



```
C user_config.h 1 X
projects > offline_asr_sample > src > C user_config.h > DEV_DRIVER_EN_ID
70
71 #define BLE_HEART_TIMEOUT 0 //ble事件管理参数
72 #define DEV_DRIVER_EN_ID 0x06: 风扇 → DEV_FAN_MAIN_ID
```

图一

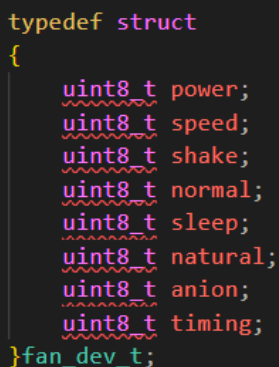


```
C ble_user_config.h 1 X
projects > offline_asr_sample > src > demo > C ble_user_config.h > DEV_NUMBER_ID
43 #define CONFIG_TYPE 0
44 #elif(DEV_DRIVER_EN_ID == DEV_FAN_MAIN_ID)
45 #define DEV_TYPE_ID DEV_FAN_MAIN_ID
46 #define DEV_NUMBER_ID 1 ← 由具体产品项目指定
47 #define CONFIG_TYPE 0
```

图二

2.2 设备功能结构体说明

每一种产品都会有其对应的功能，每次进行语音控制或者小程序控制都会改变对应功能的数值，在 SDK 中根据不同产品功能定义了对应的设备功能结构体，以风扇为例该结构体定义在 `cias_fan_msg_deal.h` 中，每次控制设备后都要将对应数值写回到该结构体中，用于确保设备与小程序同步时数值的一致性。



```
typedef struct
{
    uint8_t power;
    uint8_t speed;
    uint8_t shake;
    uint8_t normal;
    uint8_t sleep;
    uint8_t natural;
    uint8_t anion;
    uint8_t timing;
}fan_dev_t;
```

图三

2.3 各功能类型的编程说明

与小程序通讯协议有 4 种功能类型，每一种功能类型代表不同含义的数据，下面以风扇为例，对与小程序进行通讯的代码编写进行说明。

2.3.1 属性设置编程说明

小程序在设备控制界面进行按按键控制的时候，会下发对应的属性设置（功能类型为 0x01）数据到 2312 硬件，对应代码在 `cias_fan_msg_deal.c` 中的 `void fan_callback(ble_msg_V1_t msg)` 中，这时候程序要将接收到的协议进行解析，判断小程序是对哪一个功能进行设置，然后将对应的数据发送给主控。

```
void fan_callback(ble_msg_V1_t msg)
{
    int8_t ret = -1;
    uint8_t *p;
    uint8_t i = 0;
    p = (uint8_t *)&msg;
    mprintf("\nCallback: ");
    for (i = 0; i < sizeof(ble_msg_V1_t); i++)
    {
        mprintf("%02x", p[i]);
    }
    mprintf("\n");
    switch (msg.function_id)
    {
        case FAN_POWER:
        {
            if (msg.data[0] == FUN_TURN_OFF)
            {
                mprintf("关风扇\r\n");
                #if PLAY_OTHER_CMD_EN
                prompt_play_by_cmd_id(TURN_OFF, -1, default_play_done_callback, true);
                #endif
                uart_send_asr(TURN_OFF);
            }
            else if (msg.data[0] == FUN_TURN_ON)
            {
                mprintf("开风扇\r\n");
                #if PLAY_OTHER_CMD_EN
                prompt_play_by_cmd_id(TURN_ON, -1, default_play_done_callback, true);
                #endif
                uart_send_asr(TURN_ON);
            }
            fan_dev.power = msg.data[0];
            break;
        }
    }
}
```

图三

Chipintelli Technology Co., Ltd.

CONFIDENTIAL ALL RIGHTS RESERVED. This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Chipintelli Technology Co., Ltd.

2.3.2 事件上报编程说明

每当语音控制或者其他控制对设备进行控制后，需要将控制信息也上报给小程序，这时候需要用到事件上报（功能类型为 0x02），对应代码在 `cias_fan_msg_deal.c` 中的 `uint8_t fan_report(uint16_t cmd_id)` 中，这时候程序要组装好事件上报的协议，将数据填充到对应字段中，并且更新设备功能结构体 `fan_dev_t`，最后将协议上报给小程序。

```
uint8_t fan_report(uint16_t cmd_id)
{
    bool send_flag = true;
    uint8_t send_data[RF_LEN_MAX] = {0};
    uint8_t len;

    send_data[0] = 0xA5;
    send_data[1] = 0x5A;
    send_data[2] = 0x01;
    send_data[3] = 0x01;
    send_data[4] = FAN_DEV;
    send_data[5] = 0x01;
    send_data[6] = 0x21;
    send_data[8] = 0x00;
    send_data[9] = 0x01;
    switch (cmd_id)
    {
        case TURN_OFF:
        {
            send_data[7] = FAN_POWER;
            send_data[10] = FUN_TURN_OFF; //数据
            fan_dev.power = FUN_TURN_OFF;
            break;
        }
        case SPEED_LOW: //一档风
        {
            send_data[7] = FAN_SPEED;
            send_data[10] = 0x01; //数据
            fan_dev.speed = 0x01;
            break;
        }
        default:
        {
            send_flag = false;
            break;
        }
    }
}
```

图四

```
if (send_flag)
{
    uint16_t crc_cal = crc16_ccitt(0, send_data, 11);
    mprintf("crc = 0x%x\r\n", crc_cal);
    send_data[11] = crc_cal >> 8;
    send_data[12] = crc_cal & 0xFF;
    cias_crypto_data(send_data, 13);
    rf_cb_funcs.rf_send(send_data, 13);
}
return 0;
```

图五

2.3.3 设备状态查询以及应答消息编程说明

小程序进入设备控制界面的时候会显示转圈并下发状态查询（功能类型为 0x03）进行同步，这时程序需要对查询的状态进行应答，即发送应答消息（功能类型为 0x04），对应代码在 cias_fan_msg_deal.c 中的 void fan_query(ble_msg_V1_t msg)，其中可以看到，同步给小程序的信息就是由对应的设备功能结构体（风扇为：fan_dev_t）维护的，

```
void fan_query(ble_msg_V1_t msg)
{
    if ((msg.function_id < 0x01) || (msg.function_id > 0x0E))
    {
        mprintf("无效查询指令");
        return;
    }
    uint8_t send_data[RF_LEN_MAX] = {0};
    uint8_t len;

    send_data[0] = 0xA5;
    send_data[1] = 0x5A;
    send_data[2] = 0x01;
    send_data[3] = 0x01;
    send_data[4] = FAN_DEV;
    send_data[5] = 0x01;
    send_data[6] = 0x41;
    send_data[7] = msg.function_id;
    send_data[8] = 0x00;
    send_data[9] = 0x01;

    uint8_t *data = (uint8_t *)&fan_dev;
    send_data[10] = data[msg.function_id-1];
    uint16_t crc_cal = crc16_ccitt(0, send_data, 11);
    send_data[11] = crc_cal >> 8;
    send_data[12] = crc_cal & 0xFF;

    cias_crypto_data(send_data, 13);
    rf_cb_funcs.rf_send(send_data, 13);
}
```

图六

2.4 结语

应用层代码在完善上述函数功能后，将编译合成的固件烧录进 CI2311/CI2312 智能语音芯片中，可实现通过启英物联小程序对搭载该芯片的设备进行通讯。

Chipintelli Confidential

Chipintelli Technology Co., Ltd.

CONFIDENTIAL ALL RIGHTS RESERVED. This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Chipintelli Technology Co., Ltd.

[点击前往启英泰伦语音 AI 平台](#)

PAGE 10 / 11

注：本文档中使用到的第三方软件其著作权归软件作者所有，用户在使用该软件时应自行联系软件作者获取许可，启英泰伦对用户非法使用该软件及在使用该软件中产生的任何形式的损失不承担责任。

Chipintelli Confidential

Chipintelli Technology Co., Ltd.

CONFIDENTIAL ALL RIGHTS RESERVED. This document is not to be reproduced, modified, adapted, published, translated in any material form in whole or in part nor disclosed to any third party without the prior written permission of Chipintelli Technology Co., Ltd.