



Ameba DuerOS user guide

The document describes how to porting DuerOS into Ameba SDK and demonstrates how to let Ameba connect to Baidu Cloud and make the voice commands get responses.

Table of Contents

1	硬件环境.....	3
2	软件环境.....	4
2.1	DSP CODEC 固件	4
2.2	Ameba1 软件包	4
3	DuerOS	5
3.1	DuerOS 开放平台	5
3.2	Duer Profile	6
3.3	DuerOS 组件	7
3.4	DuerOS 补丁	7
3.5	CODEC 适配	8
3.6	本地提示音.....	10
3.7	Duer 示例入口.....	12
4	演示.....	12
4.1	WiFi 配置	12
4.2	轻唤 “小度小度”	14
4.3	发出语音命令.....	15
4.4	停止播放.....	15
5	故障排除.....	16

1 硬件环境

为了搭建硬件环境，需要如下设备：

➤ Ameba1 开发板

请购买 Realtek RTL8195AM 或者 RTL8711AM 开发板（下文称 Ameba1）

➤ DSP CODEC 或者 CODEC

如果选择 DSP CODEC，像 Realtek ALC5680，就可以做到本地语音解析来唤醒系统。

如果选择普通 CODEC，像 Realtek ALC5616，就只能使用按键来唤醒系统。

➤ MicroUSB

MicroUSB 线一端接在 PC 上，另一端接在 Ameba1 开发板上为其供电，并可以查看 Ameba1 上的串口 LOG，串口波特率需设置为 38400

➤ 放音设备

CODEC 音频输出口接一个自带电池的音箱（低功率，手机用）或者耳机



2 软件环境

2.1 DSP CODEC 固件

DSP CODEC 主要用于唤醒词识别、音频采集和播放功能。

从我司购买到的 ALC5680 开发板，里面会预烧好唤醒词为“小度小度”的固件。如果需要更新唤醒词，需要和我司 CODEC 部门提出需求，由我们调试生成新固件，并透过 Ameba1 下载进去。

固件更新方法请参考 Dueros 补丁中的文档 UM0151。

2.2 Ameba1 软件包

Ameba1 软件包 sdk-ameba-v4.0a_without_NDA.zip 下载地址见下面的链接：

<https://www.amebaiot.com.cn/ameba-sdk-download/>

开发环境使用 IAR，可以到 IAR 官网下载，请使用 7.20 以上的版本。

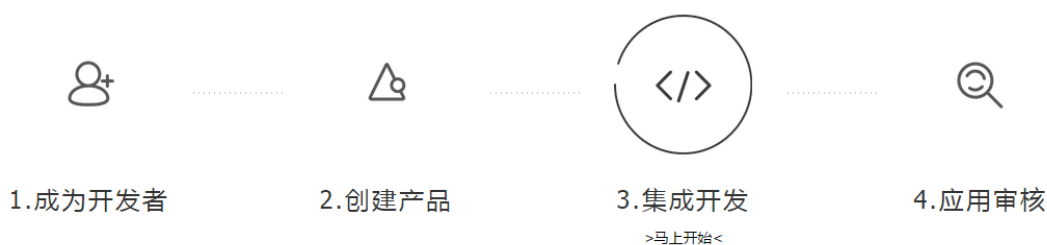
3 DuerOS

3.1 DuerOS 开放平台

DuerOS 开发平台旨在打造智能终端的人工智能（AI）时代。主页地址如下：

<http://open.duer.baidu.com/opensduer/main/index>

DuerOS 开放平台，仅需简单 4 步，即可帮助您完成产品从创建到商用，从 0 到 1 打造您的专属人工智能产品。



为方便测试，我们申请了一款名为“阿米巴”的故事机产品，详细信息如下：

产品类别： 故事机

产品名称： 阿米巴

APPID： dm7F222F9DAA7A81C7

APPKEY： 2E291126FE4EA337898B827F31C7A13D

语音AK： CpkulbUnRQIHhPjWGkDi3OHw

语音SK： b69f2212e741911ae4cb170eab454935

创建时间： 2017-06-16 18:11:27

可以在“控制台”页面左侧的“DuerOS 服务”“定制化”菜单项，看到以下服务列表，通过右侧的启用/停用可以开关某项服务。目前“儿童故事”默认使停用状态。

DuerOS服务列表

儿童故事	面向儿童的故事、儿歌等丰富资源	启用
系统画像	自定义产品形象和语音交互问答	编辑 停用
百科问答	时间、翻译、百科、问答等通用问题	停用
聊天	DuerOS智能聊天机器人	停用
天气	灵敏、智能的天气机器人	停用
娱乐	海量歌曲、音乐、有声、新闻、笑话等娱乐资源	停用
语音控制设备	使用语音指令控制设备	停用

3.2 Duer Profile

在开发平台上申请好的某款产品，通过百度审核之后，可以拿到一组 Duer Profile，以 UUID 命名，可以分发给不同设备使用。Duer Profile 中包含 UUID，Token，服务器地址

September 8, 2017

端口号和根证书等信息，用于跟云端建立 TLS 连接并向云端注册。每个 UUID 只能给一个设备使用。

为了方便大家试用，在下文将要提到的 `duerapp.c` 中，我们预留了 5 个 Duer Profile，通过设置 `DUER_PROFILE_UUID` 为 5, 6, 7, 8, 9 分别选定 5 个 Duer Profile，如下图所示：

```
#elif (DUER_PROFILE_UUID == 9)
SDRAM_DATA_SECTION static const char duer_profile[] =
{"\\configures\\": "\\}\\", "bindToken\\": "5cee4091ee86beaacf33d99b5c3667cb\\", "coapPort\\": 9001, \\
\\token\\": "H7G5hfPH\\", \\serverAddr\\": "device.iot.baidu.com\\", "lwm2mPort\\": 9001, \\
\\uuid\\": "00d500000000009\\", "rsaCaCrt\\": "-----BEGIN CERTIFICATE-----\\n\\
MIIDUDCCAjgCCQCmVPUeRMymCjANBgkqhkiG9w0BAQUFADBGMQswCQYDVQQGEwJD\\n\\
TjETMBEGA1UECAwKU29tZS1TdGF0ZTEOMAwGA1UECgwFYmFpZHUxGDAWBgNVBAMM\\n\\
DyouaW90LmJhaWR1LmNvbTEcMBoGCSqGSIb3DQEJARYNaW90QGJhaWR1LmNvbTAe\\n\\
Fw0xNjAzMTEwMzZMNDlaFw0yNjAzMDkwMzZMNDlaMGoxCzAJBgNVBAYTAkNOMRMw\\n\\
EQYDVQQIDApTb211LVN0YXRIMQ4wDAYDVQQKDAViYWlkTEYMBYGA1UEAwwPKi5p\\n\\
b3Q3YmFpZHUxY29tMRwwGgYJKoZIhvcNAQkBFg1pb3RAYmFpZHUxY29tMIIBJAN\\n\\
BgkqhkiG9w0BAQEFAAOCAQCAIAIBCCgKCAQEAatbhiIeiN7pznzuMwsLKQj2xB02+51\\n\\
OvCJ5d116ZFJecp9qtlqOfn7bm+Aja5N2aAHJtsetcTHMitY4dtGmOpw4dlGqx\\n\\
luoz50kWJWQjVR+z6DLPnGE4uELOS8vbKHUoYPPQTT80eNVnl9S9h/I7DcjEAYJC\\n\\
IYJb6+K9x+T19VRChvWcvqZQHMRym9j1g/7CKGMCiWkC+6ihkGD/XG40r7KRCyH\\n\\
bD53KnBjBO9FH4IL3rGIZWKWzWm3zC6RTS2ekfEsgAtYDvROKd4rNs+uDU9xaBLO\\n\\
dXTI0uxgudH2VnVzWtj09OUbBTxcQFD2IhmOL20BrckYul+HEIMR0eDibwIDAQAB\\n\\
MA0GCSqGSIb3DQEBBQUAA4IBAQC2TTH91jNh/uYBEFekSVNG1h1kPSujlEDDf/W\\n\\
pjQpJPqrZvW0w0cmYsYibNDy985JB87MJMfJVESG/v0Y/YbvcnRoi5gAenWXQNL4\\n\\
h2hf08A5wEQfLO/EaD1GTH3OIierKYZ6GItGrz4uFKHV5fTmifABCdu37ALGjrA\\n\\
rIjwjxQG6WwLr9468hkKrWNG3dMBHKvmqO8x42sZOFrJMKqBbKzabD1uW4xY5XwM\\n\\
S1QX56tVrgO0A3S+4dEgSuiLVN4YVP/Vqh4SMtYkL7ZZiZAxD9GtNnhRyFsWIC2r\\n\\
OVSdXs1ttZxEaEBGul7tgsBte556BIvufZX+BXGyycVJdBu3\\n\\
-----END CERTIFICATE-----\\n\\", "macId\\": "\\", "version\\": 266}";
#endif
```

3.3 DuerOS 组件

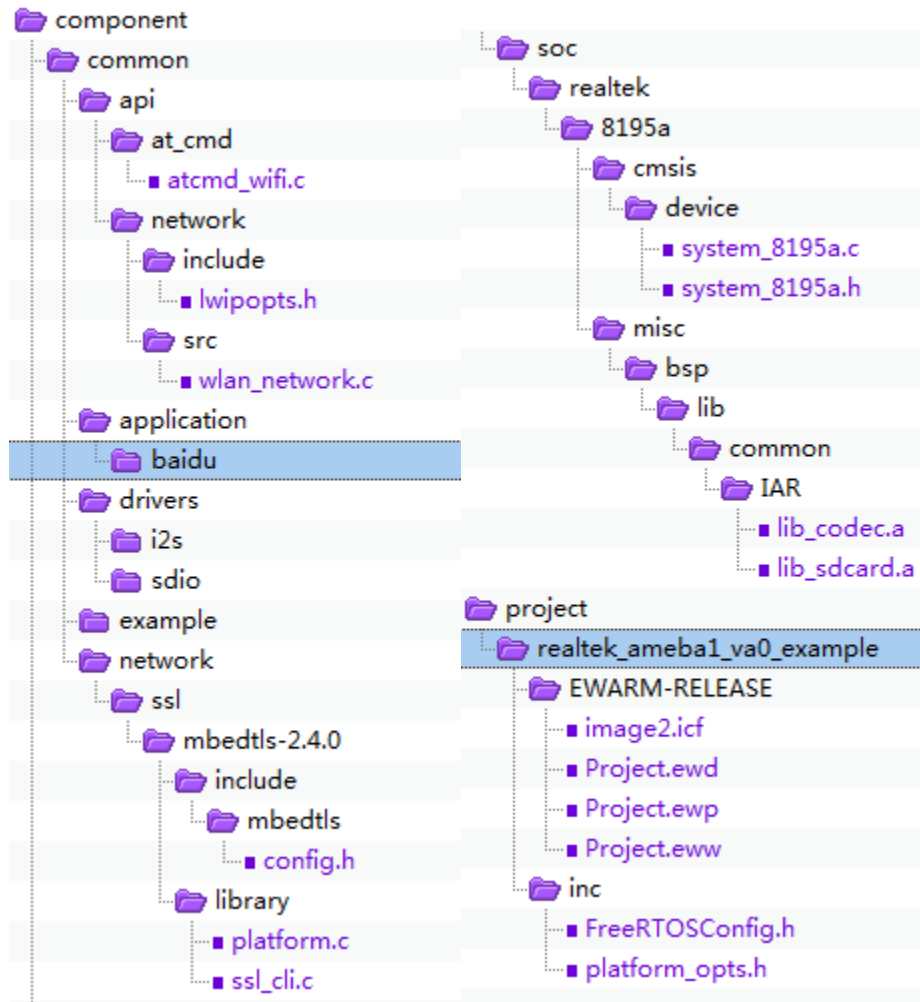
DuerOS 主要由 Lightduer, Baidu CA, Speex 三大部分组成, 它们都编译在 libduer-device.a 中。Lightduer 是 DuerOS 跟外部的接口, 包括跟 Freertos、LWIP、MbedTLS、标准库函数等对接, Baidu CA 主要实现跟百度云的连接和交流, Speex 负责语音压缩。

3.4 DuerOS 补丁

我们基于 `sdk-ameba-v4.0a_without_NDA.zip`，提供了一包适配 Realtek ALC5680 的 DuerOS 补丁，`4.0a_patch_dueros(vxx).zip`，下载地址见下面的链接：

<https://www.amebaiot.com.cn/ameba-sdk-faq/>

Patch 中主要文件及目录如下图所示:



使用方法：首先将 sdk-ameba-v4.0a_without_NDA.zip 和 4.0a_patch_dueros(vxx).zip 解压缩，并将 4.0a_patch_dueros(vxx)目录下所有文件拷贝粘贴到 sdk-ameba-v4.0a_without_NDA 目录下面，全部选择覆盖，这样就完成了。

然后请打开 sdk-ameba-v4.0a_without_NDA\project\realtek_ameba1_va0_example\EWARM-RELEASE\下面的 Project.eww，编译并透过 CMSIS-DAP 下载到 Ameba1 中，就完成固件烧录了。

3.5 CODEC 适配

CODEC 适配指的是 Ameba1 和不同型号 CODEC 之间实现通信，以完成声音采集、MP3 下载播放等功能，适配文件存放在\component\common\application\baidu\duerapp 文件夹下。

ALC5680 的适配文件如下:

➤ **duerapp_alc5680_play.c**

Ameba1 通过与 ALC5680 的 I2S 通信实现音频播放功能。

若需要适配其他型号的 CODEC, 需要按照如下说明并参考该文件以实现接口函数:

```
/**
 * @brief Execute some operations when audio starts to play
 * @param None
 * @retval None
 */
void audio_player_start();

/**
 * @brief Execute some operations when audio stop playing
 * @param None
 * @retval None
 * @Note It is necessary to initialize audio as recorder after playing music
 */
void audio_player_stop();

/**
 * @brief Tx PCM data to codec
 * @param buf: pointer to buffer which keep the pcm data
 * @param len: length of pcm data to be transmitted
 * @retval None
 */
void audio_play_pcm(char *buf, int len);

/**
 * @brief Initializes the interface with codec to be able to tx data
 * @param para: pointer to a player_t structure which stores initialization parameters
 * @retval can be 0 or -1:
 *         0: Init success
 *         -1: Init fail
 */
int initialize_audio_as_player(player_t *para);
```

➤ **duerapp_alc5680_record.c**

Ameba1 通过与 ALC5680 的 I2S 通信实现音频录制功能。

若需要适配其他型号的 CODEC, 需要按照如下说明并参考该文件以实现接口函数:

```
/**
 * @brief Execute some operations when start to record
 * @param timeout_ms: Period to be silent before stop recording
 * @retval None
 * @Note It is necessary to set audio_is_recording to 1 in this function
 */
void audio_record_start(uint32_t timeout_ms);

/**
 * @brief Execute some operations when stop recording
 * @param None
 * @retval None
 * @Note It is necessary to reset audio_is_recording to 0 in this function
 */
void audio_record_stop();

/**
 * @brief Initializes the interface with codec to be able to rx data
 * @param None
 * @retval 0: Init success
 */
int initialize_audio_as_recorder();
```

➤ **duerapp_peripheral.c**

用于调节 CODEC 音频播放音量、控制 LED 指示灯等功能。若需要适配其他型号的 CODEC，需要按照如下方式修改 duerapp_peripheral.c 中接口函数的宏定义：

```
#define CODEC_INIT
#define SET_VOLUME
#define GET_VOLUME
alc5680_i2c_init
set_alc5680_volume
get_alc5680_volume
```

3.6 本地提示音

我们加入了三种本地提示音以提升用户体验：

- “你好”
轻唤“小度小度”把 Ameba1 唤醒之后，系统会回复“你好”，然后开始录音
- “没有听清，请再说一遍”
在灯亮之后 3s 录音时间内没有采集到有效的语音，系统会做出如上回复
- “暂时无法播放，请稍后再试”
当遇到无法解析的文件格式时，系统会做出如上回复

目前本地提示音存放在 SDRAM 中，三段语音的长度分别是 1.5KB、6KB 和 5KB。若要修改本地提示音，可按如下步骤操作：

(1) 生成自定义的本地提示音，注意必须是 mp3 格式。

使用百度提供的如下语音广播开放平台可生成度秘合成音：

<http://developer.baidu.com/vcast>

- (2) 若要去除 mp3 文件的冗余部分，通过一些音频编辑器(如 Audacity 等)做剪切即可实现
- (3) 根据生成的本地提示音文件，修改 `duerapp_tone_config.c` 中相应的长度、地址和数组。

```
#define TONE_HELLO_LEN          1440
#define TONE_CANNOTPLAY_LEN    6048
#define TONE_CANNOTHEARD_LEN   4968

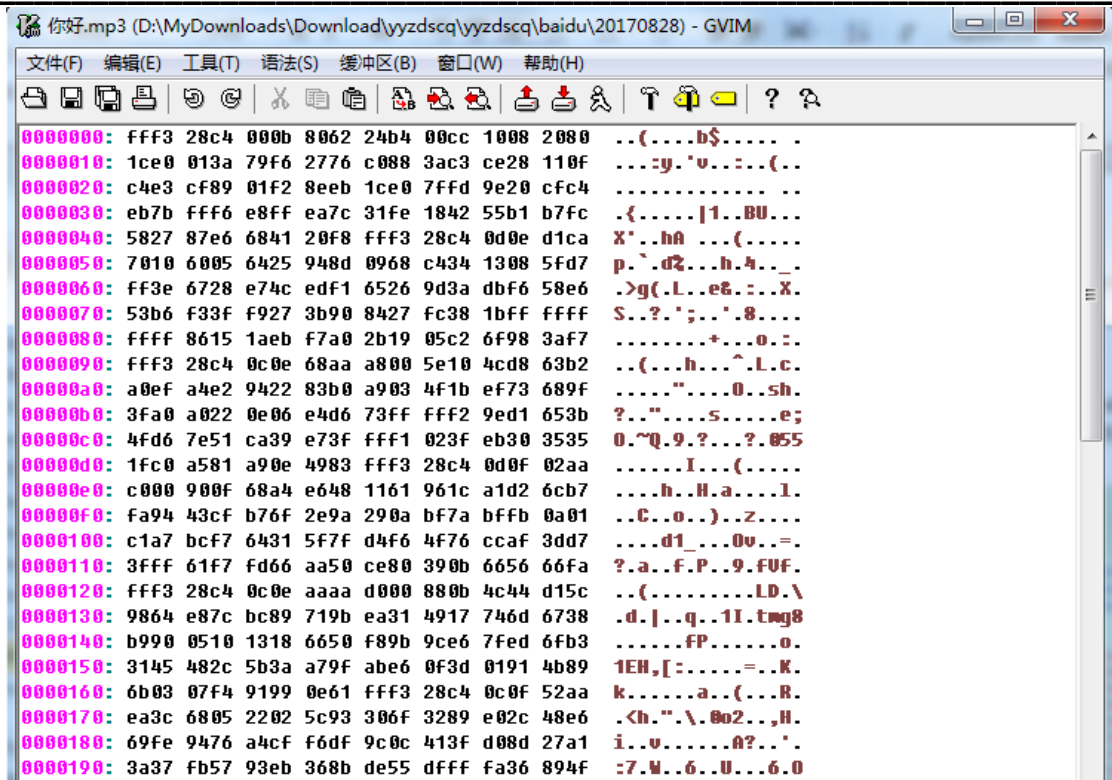
#define TONE_HELLO_ADDR         hello          //(unsigned char *)0x980C0000
#define TONE_CANNOTPLAY_ADDR    cannotPlay     //(unsigned char *)0x980C8000
#define TONE_CANNOTHEARD_ADDR   cannotHeard    //(unsigned char *)0x980D0000

SDRAM_DATA_SECTION unsigned char hello[] = {
0xFF,0xF3,0x28,0xC4,0x00,0x0B,0x80,0x62,0x24,0xB4,0x00,0xCC,0x10,0x08,0x20,0x80,
0x1C,0xE0,0x01,0x3A,0x79,0xF6,0x27,0x76,0xC0,0x88,0x3A,0xC3,0xCE,0x28,0x11,0x0F,
0xC4,0xE3,0xCF,0x89,0x01,0xF2,0x8E,0xEB,0x1C,0xE0,0x7F,0xFD,0x9E,0x20,0xCF,0xC4,
0xEB,0x7B,0xFF,0xF6,0xE8,0xFF,0xEA,0x7C,0x31,0xFE,0x18,0x42,0x55,0xB1,0xB7,0xFC,
0x58,0x27,0x87,0xE6,0x68,0x41,0x20,0xF8,0xFF,0xF3,0x28,0xC4,0x0D,0x0E,0xD1,0xCA,
0x70,0x10,0x60,0x05,0x64,0x25,0x94,0x8D,0x09,0x68,0xC4,0x34,0x13,0x08,0x5F,0xD7,
0xFF,0x3E,0x67,0x28,0xE7,0x4C,0xED,0xF1,0x65,0x26,0x9D,0x3A,0xDB,0xF6,0x58,0xE6,
0x53,0xB6,0xF3,0x3F,0xF9,0x27,0x3B,0x90,0x84,0x27,0xFC,0x38,0x1B,0xFF,0xFF,0xFF,
0xFF,0xFF,0x86,0x15,0x1A,0xEB,0xF7,0xA0,0x2B,0x19,0x05,0xC2,0x6F,0x98,0x3A,0xF7,
0xFF,0xF3,0x28,0xC4,0x0C,0x0E,0x68,0xAA,0xA8,0x00,0x5E,0x10,0x4C,0xD8,0x63,0xB2,
0xA0,0xEF,0xA4,0xE2,0x94,0x22,0x83,0xB0,0xA9,0x03,0x4F,0x1B,0xEF,0x73,0x68,0x9F,
0x3F,0xA0,0xA0,0x22,0x0E,0x06,0xE4,0xD6,0x73,0xFF,0xFF,0xF2,0x9E,0xD1,0x65,0x3B,
0x4F,0xD6,0x7E,0x51,0xCA,0x39,0xE7,0x3F,0xFF,0xF1,0x02,0x3F,0xEB,0x30,0x35,0x35,
0x1F,0xC0,0xA5,0x81,0xA9,0x0E,0x49,0x83,0xFF,0xF3,0x28,0xC4,0x0D,0x0F,0x02,0xAA,
0xC0,0x00,0x90,0x0F,0x68,0xA4,0xE6,0x48,0x11,0x61,0x96,0x1C,0xA1,0xD2,0x6C,0xB7,
0xFA,0x94,0x43,0xCF,0xB7,0x6F,0x2E,0x9A,0x29,0x0A,0xBF,0x7A,0xBF,0xFB,0x0A,0x01,
0xC1,0xA7,0xBC,0xF7,0x64,0x31,0x5F,0x7F,0xD4,0xF6,0x4F,0x76,0xCC,0xAF,0x3D,0xD7,
0x3F,0xFF,0x61,0xF7,0xFD,0x66,0xAA,0x50,0xCE,0x80,0x39,0x0B,0x66,0x56,0x66,0xFA,
0xFF,0xF3,0x28,0xC4,0x0C,0x0E,0xAA,0xAA,0xD0,0x00,0x88,0x0B,0x4C,0x44,0xD1,0x5C,
0x98,0x64,0xE8,0x7C,0xBC,0x89,0x71,0x9B,0xEA,0x31,0x49,0x17,0x74,0x6D,0x67,0x38,
0xB9,0x90,0x05,0x10,0x13,0x18,0x66,0x50,0xF8,0x9B,0x9C,0xE6,0x7F,0xED,0x6F,0xB3,
```

例如修改第一段系统提示音，则需要修改 `TONE_HELLO_LEN` 和 `hello` 数组。下图是 `hello.mp3` 以十六进制显示，需要把这个文件以十六进制的形式填入到 `hello` 数组中，如上图所示。

若要从 Flash 播放系统提示音，则需要修改 `TONE_HELLO_LEN` 和 `TONE_HELLO_ADDR`，其中 `TONE_HELLO_ADDR` 是 Flash 中存放该段提示音的起始地址。此时 `hello` 数组可以删除。

- (4) 若要从 Flash 播放本地提示音，除了上述步骤之外，还需要把生成的自定义语音下载到 Flash 指定地址，把 .mp3 后缀改成 .bin 再通过 Image Tool 下载即可实现。
- (5) 重新编译 project，下载生成的 image，reboot 之后就可以看到效果。
- (6) 这个版本设定提示音可以打断当前 HTTP MP3 播放。用户只需要调用 `tone_enqueue(TONE_CANNOTPLAY, 0)` 就可以实现。可以在 HTTP Mp3 播放时，使用串口命令 ‘ATPP’ 来测试打断效果。



```

00000000: fff3 28c4 000b 8062 24b4 00cc 1008 2080 ..(....b$. ....
00000010: 1ce0 013a 79f6 2776 c088 3ac3 ce28 110f ...:y.'v...(.
00000020: c4e3 cf89 01f2 8eeb 1ce0 7ffd 9e20 cfc4 .....
00000030: eb7b fff6 e8ff ea7c 31fe 1842 55b1 b7fc .{.....|1..BU..
00000040: 5827 87e6 6841 20f8 fff3 28c4 0d0e d1ca x'..hA...(.
00000050: 7010 6005 6425 948d 0968 c434 1308 5fd7 p.'d%....h.4...
00000060: ff3e 6728 e74c edf1 6526 9d3a dbf6 58e6 .>q(.L..e8...X.
00000070: 53b6 f33f f927 3b90 8427 fc38 1bff ffff S..?.';...'8...
00000080: ffff 8615 1aeb f7a0 2b19 05c2 6f98 3af7 .....+...o...
00000090: fff3 28c4 0c0e 68aa a800 5e10 4cd8 63b2 ..(.h...^..L.c.
000000a0: a0ef a4e2 9422 83b0 a903 4f1b ef73 689f ....."....0..sh.
000000b0: 3fa0 a022 0e06 e4d6 73ff fff2 9ed1 653b ?.."....s....e;
000000c0: 4fd6 7e51 ca39 e73f fff1 023f eb30 3535 0.^Q.9....?.055
000000d0: 1fc0 a581 a90e 4983 fff3 28c4 0d0f 02aa .....I...(.
000000e0: c000 900f 68a4 e648 1161 961c a1d2 6cb7 ....h..H.a...l.
000000f0: fa94 43cf b76f 2e9a 290a bf7a bffb 0a01 ..C..o...).z...
00001000: c1a7 bcf7 6431 5f7f d4f6 4f76 ccaf 3dd7 ....d1...0v...=
00001010: 3fff 61f7 fd66 aa50 ce80 390b 6656 66fa ?.a..f.P..9.fUf.
00001020: fff3 28c4 0c0e aaaa d000 880b 4c44 d15c ..(.....LD.\
00001030: 9864 e87c bc89 719b ea31 4917 746d 6738 .d.|..q..1l.tmg8
00001040: b990 0510 1318 6650 f89b 9ce6 7fed 6fb3 .....fP.....o.
00001050: 3145 482c 5b3a a79f abe6 0f3d 0191 4b89 1EH,[::.....=..K.
00001060: 6b03 07f4 9199 0e61 fff3 28c4 0c0f 52aa k.....a...R.
00001070: ea3c 6805 2202 5c93 306f 3289 e02c 48e6 .<h.".\.0o2...H.
00001080: 69fe 9476 a4cf f6df 9c0c 413f d08d 27a1 i..v.....0?...'.
00001090: 3a37 fb57 93eb 368b de55 dfff fa36 894f :7.W..6..U...6.0

```

3.7 Duer 示例入口

DuerOS 的入口函数为 `example_duer()`，函数定义在 `sdk-ameba-v4.0a_without_NDA\component\common\application\baidu\duerapp\duerapp.c` 中。可以通过定义 `platform_opts.h` 中的宏 `CONFIG_BAIDU_DUER` 为 1 或者为 0 来决定是否运行 DuerOS example。

4 演示

4.1 WiFi 配置

如果板子没有连接过 AP 或者之前连接的 AP 不存在了（连接 5 次都失败后），Ameba1 会自动进入 Simple Config 模式，等待用户使用手机软件 SimpleConfigWizard 进行配网。关于 SimpleConfigWizard 的使用方法可以参考文档 AN0011 第 4 章。

成功连接 AP 后，这个 profile 会被写到 Flash 偏移地址为(0x200000-0x1000)的位置，下次重启后会自动连接这个 AP。

假设 AP profile 已经存储在 Flash 中，上电后 UART LOG 中可以看到以下内容：

```
=====
ROM Version: 0.3
Build Toolchain Version: gcc version 4.8.3 (Realtek ASDK-4.8.3p1 Build 2003)
=====
Check boot type form eFuse
SPI Initial
Image1 length: 0x3a18, Image Addr: 0x10000bc8
Image1 Validate OK, Going jump to Image1
BOOT from Flash:YES
===== Enter Image 1 =====
SDR Controller Init

Load NEW fw 0
Flash Image2:Addr 0xb000, Len 204588, Load to SRAM 0x10006000
Image3 length: 0x90f70, Image3 Addr: 0x30000000
Img2 Sign: RTKwin, InfaStart @ 0x10006061
===== Enter Image 2 =====
#
duerapp: Initializing Peripheral....
Volume Register Data afaf

duerapp: volume Get, index 7, value 44975

duerapp: Peripheral initialized.
interface 0 is initialized
interface 1 is initialized

Initializing WIFI ...
Start LOG SERVICE MODE

#
duerapp: WiFi is not ready.
RTL8195A[Driver]: set ssid [FAST]

RTL8195A[Driver]: start auth to 78:eb:14:00:b9:38
RTL8195A[Driver]: auth success, start assoc
RTL8195A[Driver]: association success(res=2)

duerapp: WiFi is not ready.
Interface 0 IP address : 192.168.1.119
WIFI initialized

init_thread(55), Available heap 0x24548
```

“Interface 0 IP address : 192.168.1.119” 表示 WiFi 成功连线并拿到正确 IP。

```
duerapp: Initializing audio codec....
duerapp: Initializing audio as recorder....
duerapp: Audio has initialized as recorder.
RTL8195A[HAL]: ISR 5 had been allocated!!!
The mp3_context_t size is:- 19056
duerapp: Audio codec initialized.
duerapp: object: 1445 with 0x10057218
duerapp: DNS lookup succeeded. IP=115.239.217.99
BCAUNDEF: client state: 0
BCAUNDEF: client state: 1
BCAUNDEF: client state: 2
duerapp: <== event end = 0x30079d75, timespent = 131
BCAUNDEF: client state: 2
BCAUNDEF: client state: 3
BCAUNDEF: client state: 4
BCAUNDEF: client state: 5
BCAUNDEF: client state: 6
BCAUNDEF: client state: 7
BCAUNDEF: client state: 8
BCAUNDEF: client state: 9
BCAUNDEF: client state: 10
BCAUNDEF: client state: 11
BCAUNDEF: client state: 12
duerapp: <== event end = 0x30079e3b, timespent = 303
BCAUNDEF: client state: 12
BCAUNDEF: client state: 13
BCAUNDEF: client state: 14
BCAUNDEF: client state: 15
duerapp: event: 0
```

“BCAUNDEF: client state: 15”表示已经成功连接百度云，接下来可以进行语音控制了。

4.2 轻唤“小度小度”

“小度小度”这个唤醒词是由 ALC5680 本地识别，在识别出来后，ALC5680 会给 Ameba1 的 GPIOC_5 引脚上一个 GPIO 中断，收到中断后 Ameba1 才会启动录音功能。所

以在开始语音命令之前，需要先轻唤“小度小度”来唤醒 Ameba1，Ameba1 收到后首先会本地播放“你好”的 mp3 文件，播放结束后，再点亮 ALC5680 开发板上的指示灯。指示灯亮后，才可以开始发出语音命令。

4.3 发出语音命令

在指示灯亮之后，可以发出语音指令，例如：“天气”，“今年父亲节几号”，“唱首歌”，“刘若英 后来”，“说个笑话”，“相声”，“评书”，“你叫什么名字”，“你在哪里”等等，可以尽情地跟它聊天。

轻唤“小度小度”到播放“天气”的 LOG 如下图，详细 LOG 可以参考 Patch 目录中的 ameba1+alc5680_dueros_ok.log。

```
#
duerapp: voice irq → 收到ALC5680发过来的中断，表明唤醒词唤醒
duerapp: play tone index: 0
duerapp: Initialize audio as player. 播放提示音：你好
duerapp: audio play finished!
duerapp: Initialize audio as recorder.
duerapp: Start to record voice command. Current volume:546,average volume:56 正在录音，并发送到云端
duerapp: Record total length is 37760

duerapp: result: {"id":12560,"speech":{"content":"苏州今天多云，23℃~29℃ 空气质量31，优","url":"http://res.iot.baidu.com/api/v1/voice//939c304ac0e775f5f65fba519c55fcbc.mp3"},"logId":"1504852172230_5878","extra":{"runtime":732,"txt":"天气"},"resource":{"data":{"timeout":0},"type":"weather"},"source_type":"duer_weather","payload":{"url":"http://res.iot.baidu.com/api/v1/voice//939c304ac0e775f5f65fba519c55fcbc.mp3"},"header":{}}}
duerapp: <== event end = 0x30066e09, timespent = 132
duerapp: Downloadurl: http://res.iot.baidu.com/api/v1/voice//939c304ac0e775f5f65fba519c55fcbc.mp3 云端识别并返回结果
duerapp: Host: res.iot.baidu.com URL:/api/v1/voice//939c304ac0e775f5f65fba519c55fcbc.mp3 Dest Port: 80
duerapp: HTTP Header: HTTP/1.1 200 OK 播放文件所在URL
Cache-Control: no-cache
Connection: keep-alive
Content-Disposition: attachment;filename=939c304ac0e775f5f65fba519c55fcbc.mp3
Content-Length: 14976
Content-Type: audio/mpeg; charset=UTF-8
Date: Fri, 08 Sep 2017 06:29:34 GMT
Server: iotx/1.11.2.3
X-Bce-Request-Id: 0cecf7f5-74dd-4c31-aed4-ece30ed1cd13 播放天气预报

duerapp: Initialize audio as player.
duerapp: http content-length = 14976 bytes, download resource size = 14976 bytes, cost time 7595
duerapp: HTTP Download Thread Exit!
duerapp: audio play finished!
duerapp: Initialize audio as recorder.
```

4.4 停止播放

如果能让正在播放的音乐或者其他内容停下来，有两种方法可以做到。一种是通过呼叫“小度小度”，一种是可以给 GPIOC_5 一个上升沿。Ameba1 收到 GPIOC_5 的中断，就会停止播音并转成录音模式。

播放音乐中，利用小度小度来喊停，目前效果还没有非常理想，CODEC ALC5680 上的固件在继续调优中，后期会更新。

5 故障排除

■ 关于 ALC5680 底板上 LED 闪灯的几点说明：

- (1) 从 Wi-Fi 连线到连上百度云这段期间，间隔 300ms 闪灭一次；
- (2) 如果 5 次连接 Wi-Fi 失败，则清除 flash 进入 Simple Config，此时 150ms 闪灭一次；
- (3) 成功连上百度云之后，灯灭。

■ 关于录音的几点说明：

- (1) 在灯亮之后开始录音。灯亮前 Audio 在 play 模式，播放本地提示音“你好”，之后才会切回到 record 模式并点灯；
- (2) 灯亮之后，开始监测声音变化，如果声音高于三倍环境音，判断为录音开始；
- (3) 如果在灯亮后，声音高度始终没有满足(2)中条件，3 秒后停止录音等待重新唤醒；
- (4) 录音过程中，如果声音高度小于三倍环境音并持续 0.5 秒，则判定为断句；
- (5) 本地提示音共存有三个：“你好”，“没有听清，请再说一遍”，“暂时无法播放，请稍后再试”。播放“没有听清，请再说一遍”表示录音没录到，不会上传到百度云。播放“暂时无法播放，请稍后再试”表示百度云提供的链接文件格式暂时不支持，例如：.m4a 文件；
- (6) 目前最长录音时间设置的是 5 秒钟，RECORD_LEN 设置为 160*1024。

■ Q/A

Q1: “小度小度”无法唤醒系统

A1: 可以尝试将语速略微放慢，并吐字清晰，可以明显提高成功率。

Q2: 语音指令发出后，播出“没有听清，请再说一遍”

A2: 这个是本地提示音，发生的原因可能是在指示灯没有亮之前就发出语音命令。

Q3: 语音指令发出后，播出“暂时无法播放，请稍后再试”

A3: 这个是本地提示音，发生的原因是云端传送过来的文件格式，我们暂时无法播放，例如：m4a 格式。

Q4: 立即重启后，发现 LOG 打印“client state: 2”之后，很长时间不动，过一会才能恢复

A4: 原因是上一次的 TLS 连接被重启异常关闭，服务器在没有判定客户端离线前不让接入。如果需要重启，可以调用 ATW8，正常走完 4 次挥手，重启后就能很快连上服务器。

Q5: 有时候语音指令要下很多遍，才能听到正确的结果

A5: 请保持一定的语速，并且咬字清晰。下图中，LOG 里面可以看到云端从语音命令中识别出来的文字，可以跟发出的命令核对。请将串口工具编码方式修改为 UTF-8，否则这里是乱码。

```
duerapp: result: {"id":49458,"speech":{"content":"苏州今天多云, 27°C-33°C 南风3-4  
49a5-ae73-7a817cf29519.mp3"},"extra":{"runtime":911,"txt":"天气"},"resource":{}}
```

Q6: 灯亮后，还没有下命令，灯就灭了，语音命令没有响应

A6: 我们在灯亮后，会等待 3 秒的时间，如果 3 秒内没有语音命令，就会关闭录音。所以在灯亮前或者灯灭后开始语音指令，都是录不下来的。