

# MVsB5\_BT\_Audio\_SDK 快速上手指南

V0.1

## 版本记录

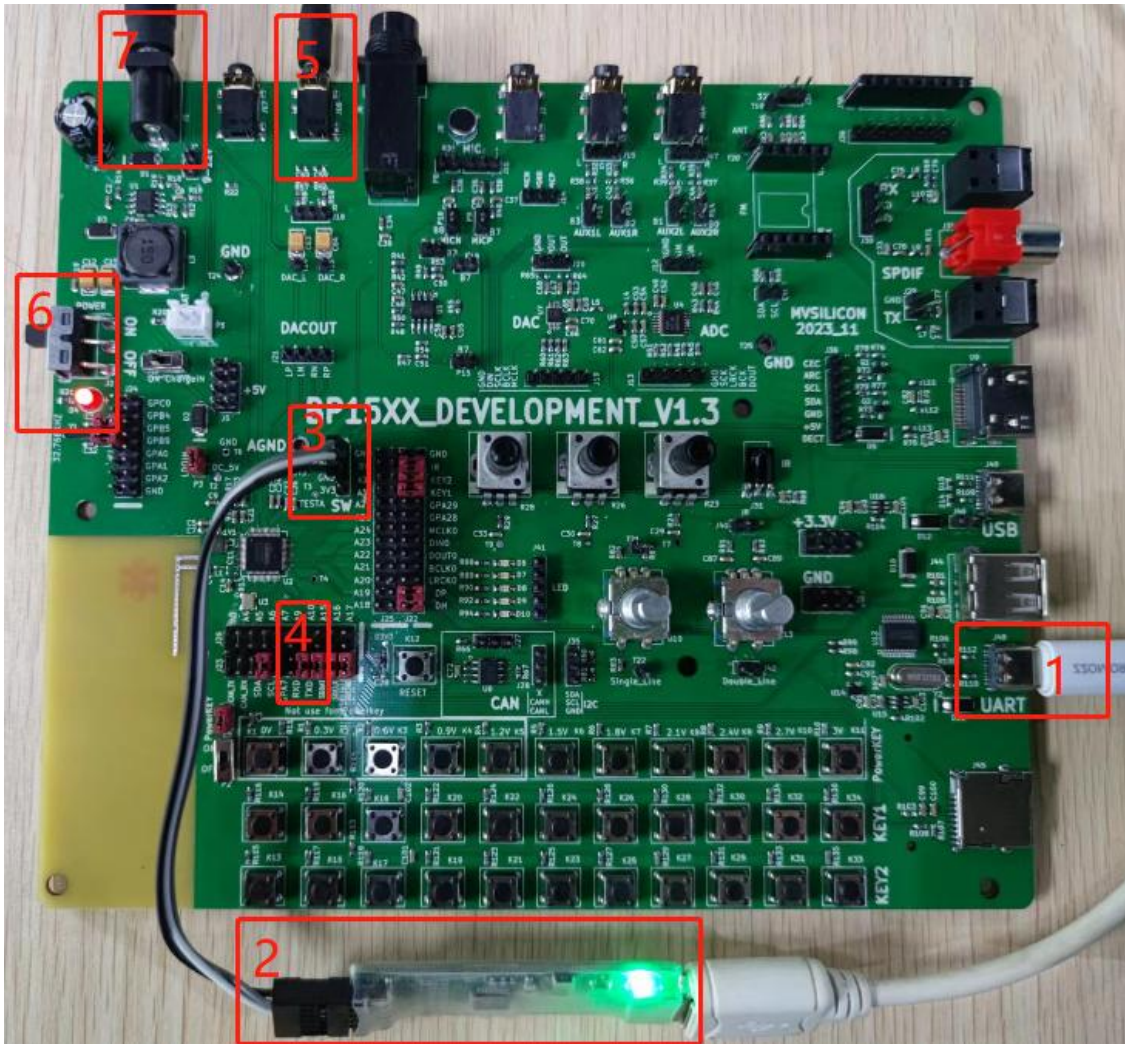
版本	作者	日期	修改日志
V0.1	Liangzx	2024-1-23	初版发布

# 目录

MVSB5_BT_Audio_SDK 快速上手指南 .....	1
版本记录 .....	2
1. 开发环境搭建 .....	4
1.1 开发板介绍 .....	4
1.2 Andes 开发环境 .....	5
1.2.1 Andes 软件 .....	5
1.2.2 固件下载 .....	5
2. MVSB5_BT_Audio_SDK 介绍 .....	6
2.1 系统框架 .....	6
2.2 目录介绍 .....	7
2.2.1 SDK 根目录 .....	7
2.2.2 BT_Audio_APP 目录 .....	7
2.2.2 app_framework 目录 .....	7
2.2.3 app_src 目录 .....	8
2.2.4 driver 目录 .....	8
2.2.5 middleware 目录 .....	9
2.2.6 Release 目录 .....	9
2.2.7 remind_res 目录 .....	10
2.2.8 tools 目录 .....	10
2.3 SDK 下载运行 .....	11
2.4 软件运行流程 .....	11
2.4.1 概述 .....	11
2.4.2 main()说明 .....	12
2.5 系统功能配置 .....	12
2.6 按键 .....	14
2.7 显示功能 .....	14
2.8 调音功能 .....	14
2.9 系统参量化数据修改 .....	14
2.10 Remind 提示音文件修改 .....	15
2.11 增加模式 .....	16
3. SDK 软件简要指南 .....	17
3.1 山景 SDK 默认配置 .....	17
3.2. 新增功能范例 .....	17
3.2.1 TimeOut 用法 .....	17
3.2.2 OS 互斥锁用法 .....	18
3.2.3 消息接收发送 .....	18
4. 常见问题 .....	18

## 1. 开发环境搭建

### 1.1 开发板介绍



电脑-AICE-BP15 开发板-串口打印-耳机-电源连接示意图

说明：1、USB 串口打印端口：通过 USB 线连接电脑。电脑上可以下载软件 SSCOM 看打印消息。

2、山景 AICE：AICE USB 端口连接电脑，另外一端连开发板上 3，接线如下。

AICE	PIN4 TMS	PIN6 TCK	PIN3 GND	不连
下载仿真调试口	SWD	SWCLK	GND	3V3

3、下载仿真调试口：通过山景 AICE 和电脑连接。

4、打印 UART：通过 1 和电脑 USB 连接。MVsB5\_BT\_Audio\_SDK 默认打印 IO 为 GPIOA10，波特率 2000000。注意 A10 要跳线才能连到 PCB 自带串口。

5、耳机接线孔：连接耳机，可以听开发板播放的声音。开发板耳机接口只有左右声道没有 MIC，所以有 MIC 的耳机连接开发板时要注意连接方式。

6、开发板电源开关和电源指示灯：接通电源后按下开关打开电源，红灯亮。

7、9V 或 12V 直流电源插孔：此接口接 9V 或 12V 直流电源。

## 1.2 Andes 开发环境

### 1.2.1 Andes 软件

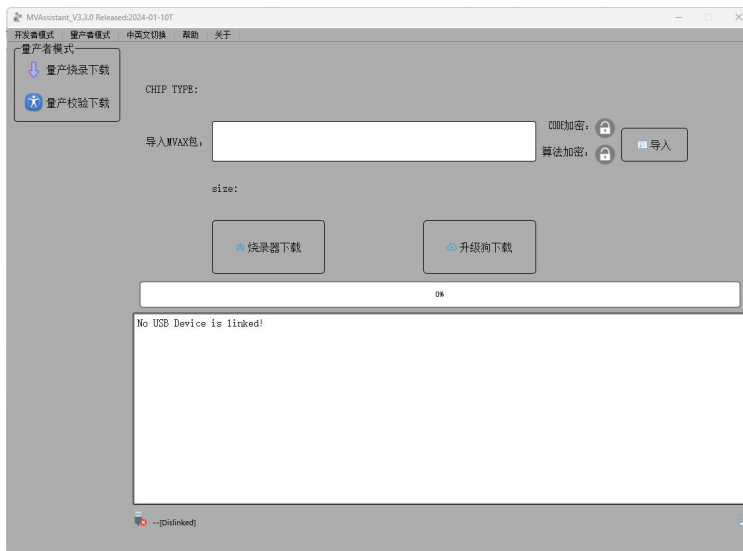
软件安装、工程导入、编译下载、仿真、断点调试，请参考文档 《AndeSight V.x.x.x 使用指南》。

### 1.2.2 固件下载

- 1) Andes AICE Mini 仿真器下载：详见《AndeSight V.x.x.x 使用指南》
- 2) U 盘/TF 升级：把编译生成或者上位机 MVAssistant 合成的.mva 文件拷贝到 U 盘/TF 卡根目录下；在开发板开机情况下插入即可升级；升级成功后会自动重启；
- 3) 升级狗升级

PC 软件：MVAssistant\_Vx.x.x（相关操作使用说明参见 MVAssistant\_Vx.x.x 工具帮助说明）

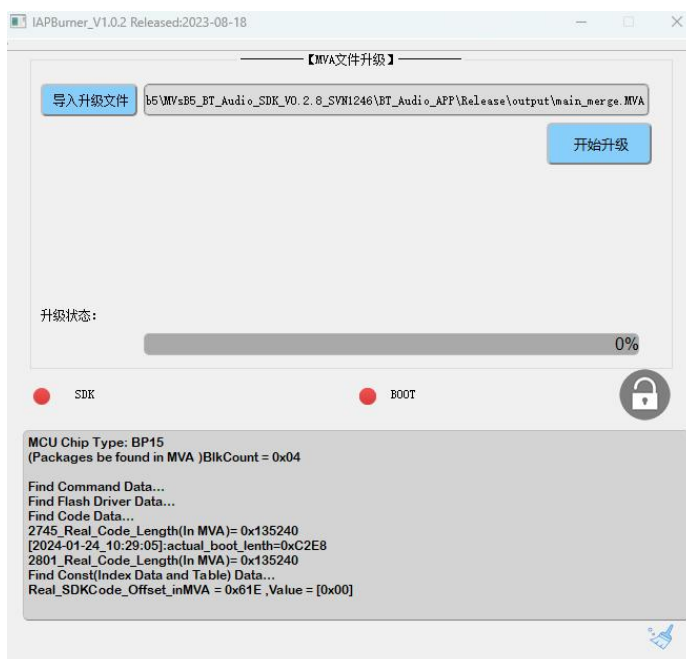
升级狗：新版本升级狗



- 4) PC 在线下载

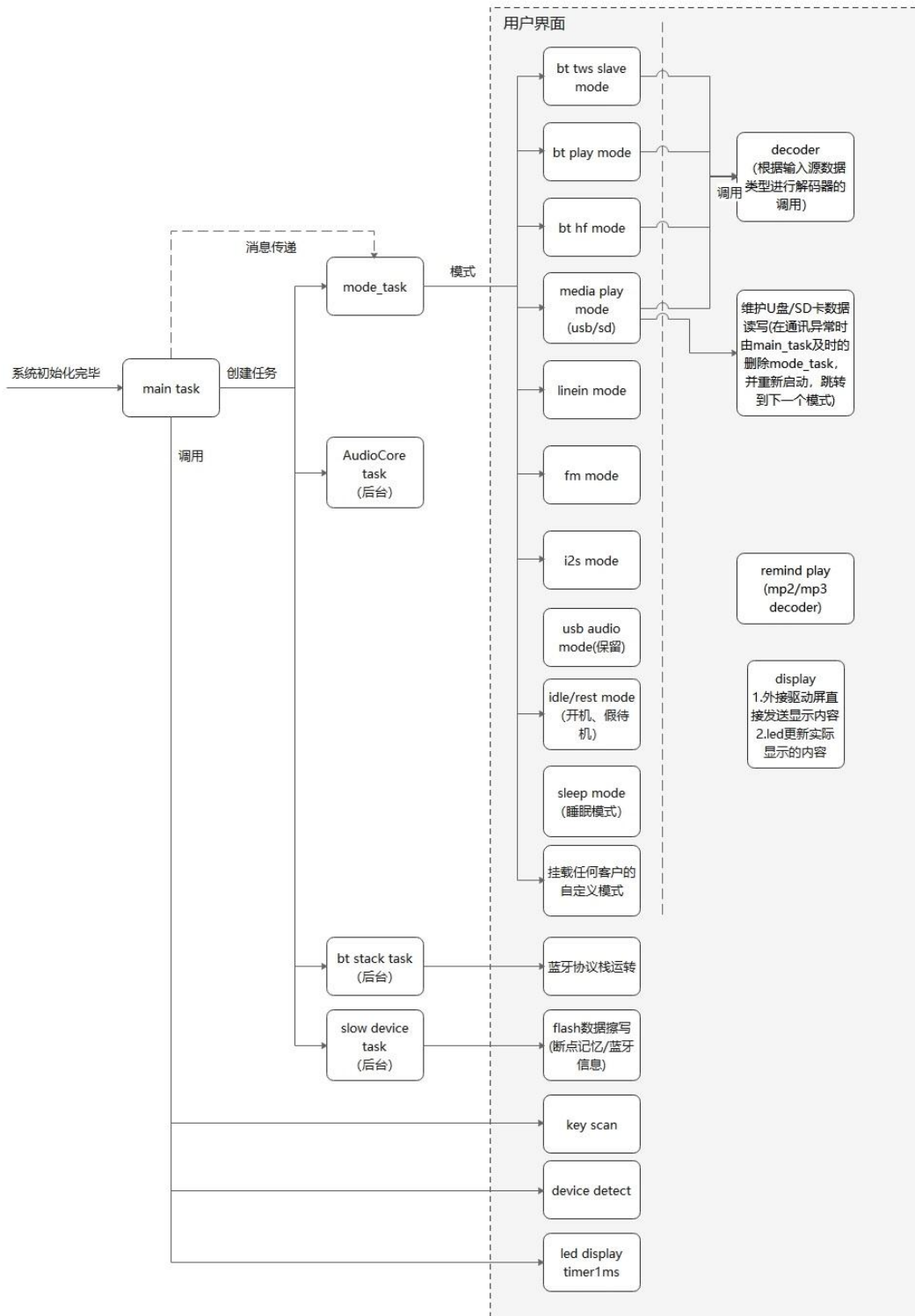
PC 软件：IAPBurner\_Vx.x.x

在开发板开机状态下插入 USB 线；然后导入 SDK 生成的.mva 文件；点击“开始升级”即可。



## 2. MVsB5\_BT\_Audio\_SDK 介绍

### 2.1 系统框架



2.2 目录介绍

2.2.1 SDK 根目录





 BT_Audio_APP	1、BT_Audio_APP: SDK 程序开发包
 documents	2、documents: 培训说明问题
 Examples	3、Examples: 参考例程

2.2.2 BT\_Audio\_APP 目录

此目录是所有的源代码、配置文件、库文件、目标文件、数据、工具：

 .settings	1、.settings: Andes 配置
 app_framework	2、app_framework: 系统框架和引擎
 app_src	3、app_src: 应用代码
 driver	4、driver: 底层驱动代码
 middleware	5、middleware: 中间件代码
 Release	6、Release: 编译后输出文件夹
 remind_res	7、remind_res: 资源文件夹（提示音文件）
 test	8、test: 测试相关文件
 tools	9、tools: SDK 软件工具集合
 .cproject	10、.cproject/.project: 工程配置文件
 .project	11、BP15x.sag: Andes SAG 文件
 BP15x.ld	12、BP15x.ld: 编译器链接文件
 BP15x.sag	13、BT_Audio_APP_history.txt: 版本历史记录
 BT_Audio_APP_history.txt	

2.2.2 app\_framework 目录

 audio_engine	此部分主要为系统常驻的 4 个 task:
 bluetooth_engine	1、audio_engine: Audio_core task, 音频的输入输出控制和调节。
 mode_engine	2、bluetooth_engine: Bt_stack task, 蓝牙协议栈任务。
 slow_device_engine	3、mode_engine: Mode task, 模式管理
	4、slow_device_engine: slow_device task, 低速设备任务-主要是 flash 擦写



2.2.3 app\_src 目录

<div><div>app_mode_common</div><div>app_mode_bt</div><div>app_mode_hdmi</div><div>app_mode_i2s</div><div>app_mode_idle</div><div>app_mode_linein</div><div>app_mode_media</div><div>app_mode_radio</div><div>app_mode_spdif</div><div>app_mode_usb_audio</div><div>components</div><div>hmi</div><div>power</div><div>startup</div><div>system_config</div><div>main.c</div><div>main_task.c</div><div>main_task.h</div></div>	<div>1、app_mode_common：为应用层提供系统接口；mode 资源申请释放/spdif_out/i2s MIX 等。</div> <div>2、app_mode_bt：提供蓝牙免提/蓝牙播歌的用户交互应用</div> <div>3、app_mode_hdmi：HDMI 应用</div> <div>4、app_mode_i2s：提供 IIS 数字输入应用</div> <div>5、app_mode_idle：开关机待机模式；包括 deep_sleep/power_down/开关机提示音及逻辑处理。</div> <div>6、app_mode_linein：音频输入功能应用，实现音频输入播放功能的用户交互</div> <div>7、app_mode_media：TF 卡/U 盘的本地磁盘歌曲播放</div> <div>8、app_mode_radio：FM 收音机的应用与交互</div> <div>9、app_mode_spdif：光纤/同轴的应用与交互</div> <div>10、app_mode_usb_audio：USB 声卡的应用与交互</div> <div>11、components：系统插件集合。audio 部分的交互处理，蓝牙的应用层接口，OTG 应用层接口，audio 音效处理，音效框图，升级接口等。</div> <div>12、hmi：系统外设插拔检测部分、LED 屏显示、按键、FM 驱动部分等。</div> <div>13、power：用于系统的休眠与唤醒及电池电量管理、</div> <div>14、startup：系统启动部分</div> <div>15、system_config：系统主要配置头文件</div> <div>16、main.c</div> <div>17、main_task.c/main_task.h：系统常驻 task，用于处理公共部分消息</div>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.2.4 driver 目录

<div><div>driver</div><div>driver_api</div><div>readme.txt</div></div>	<div>底层驱动部分，用户不用太关注。</div> <div>1、driver：底层单独模块驱动接口</div> <div>2、driver_api：对 Driver 做了集成，提供了常用的一些接口</div>
------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------



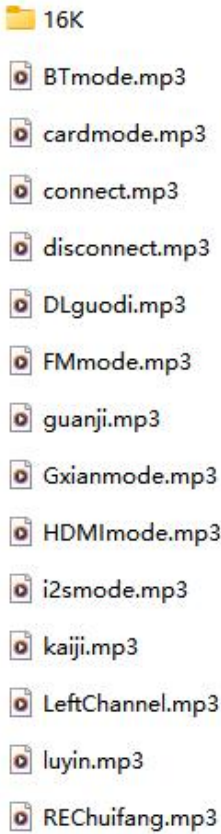
2.2.5 middleware 目录

<div><div>audio</div><div>bluetooth</div><div>cec</div><div>character_encoding</div><div>fatfs</div><div>flashboot</div><div>lrc</div><div>mv_utils</div><div>roboeffect</div><div>rtc</div><div>rtos</div><div>user</div></div>	<p>系统中间件部分，其中涉及大量与应用细节无关的库文件包含音效库、解码库、蓝牙库、flashboot、文件系统等。</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------

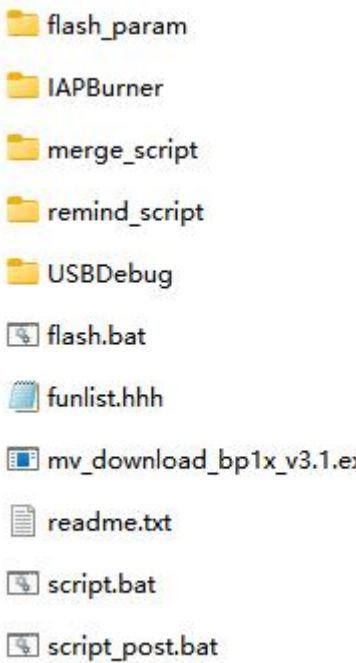
2.2.6 Release 目录

<div><div>.metadata</div><div>app_framework</div><div>app_src</div><div>driver</div><div>middleware</div><div>output</div><div>test</div><div>BT_Audio_APP.adx</div><div>makefile</div><div>objects.mk</div><div>sources.mk</div></div>	<p>Release 是代码编译生成目标文件的目录。</p> <p>用于生产的.img; .mva 文件都在此目录的 output 下。</p> <p>其他文件无需关注。</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------

### 2.2.7 remind\_res 目录

 <p>16K</p> <ul style="list-style-type: none"> <li>BTmode.mp3</li> <li>cardmode.mp3</li> <li>connect.mp3</li> <li>disconnect.mp3</li> <li>DLguodi.mp3</li> <li>FMmode.mp3</li> <li>guanji.mp3</li> <li>Gxianmode.mp3</li> <li>HDMImode.mp3</li> <li>i2smode.mp3</li> <li>kaiji.mp3</li> <li>LeftChannel.mp3</li> <li>luyin.mp3</li> <li>REChuifang.mp3</li> </ul>	<p>1、此目录为提示音资源文件目录，客户修改提示音就替换或删除此目录下的提示音文件。</p> <p>2、16K：这个目录下面一般提供 HFP 模式下 16K 采样率的提示音文件。</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------

### 2.2.8 tools 目录

 <p>flash_param</p> <p>IAPBurner</p> <p>merge_script</p> <p>remind_script</p> <p>USBDebug</p> <p>flash.bat</p> <p>funlist.hhh</p> <p>mv_download_bp1x_v3.1.exe</p> <p>readme.txt</p> <p>script.bat</p> <p>script_post.bat</p>	<p>此目录主要为一些批处理，方便更简洁的开发或转译。</p> <p>1、flash_param：参量化参数工具，在线编辑参数/离线编辑 MVA 文件中的参数。</p> <p>2、IAPBurner：上位机升级工具。</p> <p>3、merge_script：管理索引表中的内容，直接影响提示音/参量化数据/蓝牙配对等数据在 flash 中的位置。用户也可自定义添加，定义和管理自己的数据。</p> <p>4、remind_script：提示音打包脚本</p> <p>5、USBDebug：USB 打印接收工具。SDK 需要对应开启该功能，CFG_FUNC_USBDEBUG_EN</p> <p>6、flash.bat：手动打包工具，用于打包生成可量产的.img/.mva 文件</p> <p>7、funlist.hhh：ROM 中提供的函数 API</p> <p>8、mv_download_bp1x_v3.1.exe：download 下载工具。</p> <p>9、script.bat：Andes 自动调用提示音生成/参量化数据生成脚本，并更新对应头文件到工程中。</p> <p>10、script_post.bat：Andes 自动调用 merge 打包脚本，生成.img/.mva 并更新文件到 output</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 2.3 SDK 下载运行

MVsB5\_BT\_Audio\_SDK 默认是一个蓝牙音箱的配置。开发者只需按文档《AndeSight V.x.x.x 使用指南》导入工程，编译 BT\_Audio\_APP 工程并下载就可以正常运行程序。下载完成后按电源开关或按 Reset 键，程序就可以运行。运行打印信息如下图：

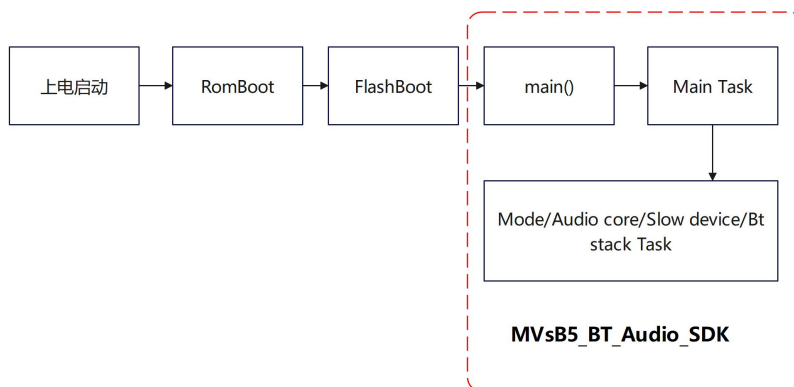
```
[19:08:15.456]收←◆
*****
                MVsB5_BT_Audio_SDK
                Mountain View Silicon Technology Co.,Ltd.
                SDK Version: 0.2.12
*****
sys clk =120000000
read flash Capacity = 0x200000 0x200000,FlashTable: 1
sys_parameter = 0x1fc000
bp_data = 0x1ce000
bt_data = 0x1d0000
remind = 0x1d2000
user_config_addr = 0x1fe000
bt_config_addr = 0x1ff000
flash_parameter_ver: Ver 1.0.0
BB_EM_SIZE=20480,EM_BT_END=17988
bt em size:20KB
RstFlag = c0
Audio Decoder Version: 8.10.4 build @ Jan 23 2024 18:59:14
Audio Effect Lib Version: 2.36.0
Roboeffect Lib Version: 2.14.0
Driver Version: 0.1.17 build @ Jan 17 2024 09:33:54 0
BtLib Version: 15_0.2.6 build @ Dec 21 2023 18:46:11
Fatfs presearch acc Lib Version: 1.6.1 build @ Dec 12 2023 11:01:31

Load BP INFO START
EffectMode:1,1
MusicVolume:24,24
MicVolume:32,32
HfVolume:32,32
MicEffectDelayStep:32,32
power on mode LinePlay
[SYS]: Loading control vars as default
AudioCore init
bluetooth stack service init.
MainApp:run
MainApp:AudioCore service created
---MSG_TASK_START---
header error
used default bt trim value
```

## 2.4 软件运行流程

### 2.4.1 概述

MVsB5\_BT\_Audio\_SDK 嵌入了 FreeRTOS，按实际需求做了相应的裁减。FreeRTOS 是一个小型嵌入式系统，功能包括：任务管理、信号量、消息队列、内存管理等。多任务系统可以简单理解为多个 while(1){...} 同时运行。对于初次接触 RTOS 的开发者，不必过于关注 RTOS 的底层和原理，可以参照 SDK RTOS 相关用法快速上手。程序启动流程如下：



2.4.2 main() 说明

```
Main()//main.c
{
    系统初始化;//晶振, 时钟, 看门狗, 打印口,系统堆建立
    ...
    osSemaphoreMutexCreate(); //互斥锁定义
    MainAppTaskStart(); //启动 main APP 任务
    vTaskStartScheduler(); //启动任务调度器, 各任务开始分时执行
}
```

2.4.3 MainAppTaskEntrance()说明

MainAppTaskStart()创建 MainApp Task。MainApp 在整个程序运行期间一直存在，主要负责系统变量初始化，创建其他任务，公共消息管理，设备检测，按键检测，模式管理。

```
MainAppTaskEntrance() //Main_task.c
{
    SystemInit(); //系统变量初始化, 创建 audio core task/bt stack task
    PublicDetect(); //公共事件检测
    PublicMsgPross(); //公共消息的处理, 注意里面有创建 mode task 和 slow device task
    WDG_Feed(); //喂狗
    DeviceServicePocess();//USB/SD/LINE IN 检测, 按键扫描, 断点消息更新
    SysModeGenerate(); //系统转模式管理
    SysModeChangeTimeoutProcess ( ) ; //模式超时管理
}
```

2.5 系统功能配置

MVsB5\_BT\_Audio\_SDK 很多功能都是通过宏开关控制。基本都在 system\_config 文件夹下面的 h 文件里面进行配置。

app\_config.h：主要是系统和音频相关的宏

<pre><i>// *****</i> <i>// 系统App功能模式选择</i> <i>// *****</i> #define CFG_APP_IDLE_MODE_EN #define CFG_APP_BT_MODE_EN #define CFG_APP_USB_PLAY_MODE_EN #define CFG_APP_CARD_PLAY_MODE_EN #define CFG_APP_LINEIN_MODE_EN // #define CFG_APP_RADIOIN_MODE_EN #define CFG_APP_USB_AUDIO_MODE_EN // #define CFG_APP_I2SIN_MODE_EN #define CFG_APP_OPTICAL_MODE_EN <i>// SPDIF 光纤模式</i> #define CFG_APP_COAXIAL_MODE_EN <i>// SPDIF 同轴模式</i> // #define CFG_APP_HDMIIN_MODE_EN</pre>	<p>系统模式配置。</p> <p>可以根据实际需求，删减不需要的模式。</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------

<pre>/**DAC通道配置选择**/ #define CFG_RES_AUDIO_DAC0_EN  /**I2S音频输出通道配置选择**/ //#define CFG_RES_AUDIO_I2SOUT_EN  /**光纤同轴音频输出通道配置选择**/ //#define CFG_RES_AUDIO_SPDIFOUT_EN</pre>	<p>音频输出配置。</p> <p>可以根据需求选择 DAC/I2S/SPDIF 输出。</p>
<pre>#define USE_MP3_DECODER #define USE_WMA_DECODER #define USE_SBC_DECODER #define USE_WAV_DECODER //#define USE_DTS_DECODER #define USE_FLAC_DECODER / //#define USE_AAC_DECODER //#define USE_AIF_DECODER //#define USE_AMR_DECODER #define USE_APE_DECODER</pre>	<p>解码器配置。</p> <p>不需要的解码格式可以在这里自行配置。</p>
<pre>#if CFG_RES_MIC_SELECT // #define CFG_FUNC_MIC_KARAOKE_EN //MIC karaoke功能选择 #endif  #ifdef CFG_FUNC_MIC_KARAOKE_EN //Line in mix for Karaoke #ifndef CFG_APP_LINEIN_MODE_EN // #define CFG_FUNC_LINEIN_MIX_MODE #endif // I2S mix mode for Karaoke //#define CFG_RES_AUDIO_I2S_MIX_OUT_EN //#define CFG_RES_AUDIO_I2S_MIX_IN_EN //USB Audio mix for Karaoke #ifdef CFG_APP_USB_AUDIO_MODE_EN // #define CFG_FUNC_USB_AUDIO_MIX_MODE //USB Audio mix需要开启USB_AUDIO_MODE #endif</pre>	<p>KARAOKE 配置。</p> <p>可以进行 LINEIN MIX/I2SMIX/USB AUDIO MIX 配置。</p>
<pre>#include "debug.h" #define CFG_FUNC_DEBUG_EN //#define CFG_FUNC_USBDEBUG_EN #ifdef CFG_FUNC_DEBUG_EN #define CFG_UART_TX_PORT          DEBUG_TX_A10 #define CFG_UART_BANDRATE        DEBUG_BAUDRATE_2000000/ #define CFG_FLASHBOOT_DEBUG_EN   ENABLE//ENABLE #endif</pre>	<p>日志打印配置。</p> <p>可以配置 UART 方式/USB 方式。</p> <p>可以配置 FLASHBOOT 打印（只支持 UART 方式）。</p>

bt\_config.h：蓝牙设置相关的宏

<pre>//蓝牙双手机连接开关 //#define BT_MULTI_LINK_SUPPORT  #define BT_NAME          "BP15_BT" #define BLE_NAME          "BP15_BLE"</pre>	<p>BT/BLE 默认名称配置。</p> <p>可以通过 OCPWorkBench 工具在线修改，或者导入 MVA 文件离线修改。</p>
---------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------

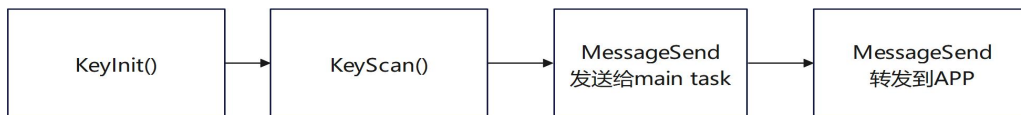


<pre>#define BT_A2DP_SUPPORT          ENABLE //A2DP和AVRCP关联 #if CFG_RES_MIC_SELECT #define BT_HFP_SUPPORT          ENABLE #else #define BT_HFP_SUPPORT          DISABLE #endif #define BT_SPP_SUPPORT          DISABLE</pre>	BT 协议配置。 A2DP/HFP/SPP 可以按需求进行配置。
<pre>#define BLE_SUPPORT          DISABLE</pre>	BLE 功能开启/关闭。

**chip\_config.h** : 芯片选型配置。  
**power\_config.h** : 系统频率配置。  
**sys\_gpio.h** : SDK 硬件 gpio 配置。

## 2.6 按键

MVsB5\_BT\_Audio\_SDK 支持多种类型的按键，包括普通 ADC\电位器\编码开关\IO 按键\IR 遥控等多种按键。并且对按键事件进行了统一处理，所有按键程序流程都一样，如下：



说明：

KeyInit(); //函数里面完成各种按键初始化

KeyScan(); //函数里面包括所有按键的处理过程：扫描，去抖，获取按键值，按键值转换成按键消息

MessageSend(mainAppCt.msgHandle...); //把按键消息发给 mainApp

MessageSend(APP...); //mainApp 处理系统级按键消息，属于模式 Task 的消息，分发给各模式

用户如果要改一个按键功能，只需在 key.c 文件 KEY 按键属性对应消息列表中替换对应的消息即可。

## 2.7 显示功能

app\_config.h 里面打开宏 CFG\_FUNC\_DISPLAY\_EN，可开启 LED 显示功能。MVsB5\_BT\_Audio\_SDK 目前刷屏是在 Timer6Interrupt（）中断里面做的，中断里面所加的程序执行时间要短。

## 2.8 调音功能

app\_config.h 里面打开宏 CFG\_FUNC\_AUDIO\_EFFECT\_EN，使能 SDK 音效功能。

山景 BP15 芯片支持在线调音功能。通过音频在线调音工具 ACPWorkbench.exe，开发者可以很方便的设置 BP15 部分音频参数和多达 30 种音效参数，如 MV3D，MVBass，Vocal Cut，Reverb，Pro，PCM Delay，ReverbPro，VoiceChangerPro，EQ 等。

详细的介绍请参考文档《V3 架构应用指导说明》。

## 2.9 系统参量化数据修改

举例修改蓝牙名称：

1、SDK 默认修改，bt\_config.h 修改 BT\_NAME。

2、在线修改，PC 用 USB 线连接目标板，打开 OCPWorkBench 工具。点击 Receive\_data，修改蓝牙名称，然后点击 Send\_data



3、MVA 文件直接修改。打开 MVA 文件，修改蓝牙名称，然后点击保存。



## 2.10 Remind 提示音文件修改

目前我们采用 MP3 格式单声道，资源文件在..\BT\_Audio\_APP\remind\_res 目录下：

替换：MP3 提示音文件替换原来的音频文件即可。

新增：如果要新增提示音只需要把 MP3 提示音放在此目录下即可。

删除：直接删除不需要的 MP3 提示音文件即可。

注：MP3 提示音文件不能含有 ID3 等标签信息。提示音删除可能会造成编译报错，只需要把代码中调用该提示音的部分屏蔽即可。修改完成后直接编译代码就可在目录..\BT\_Audio\_APP\tools\remind\_scrip 下生成提示音 all.bin。



## 2.11 增加模式

- 1) 枚举类型 SysModeNumber 里增加一个值

```
typedef enum
{
    ModeIdle= 0,
    ModeBtAudioPlay,
    ModeUDiskAudioPlay,
    ModeCardAudioPlay,
    ModeLineAudioPlay,
    ModeUshDevicePlay, //5

    ModeI2SInAudioPlay, // 6
    ModeRadioAudioPlay, //
    ModeBtHfPlay, //
    ModeTwsSlavePlay,
    ModeTest
    SysModeMax, //
} SysModeNumber;
```

- 2) 增加 init,deinit,run 函数

```
bool ModeTestInit(void)
{
    return TRUE;
}
bool ModeTestDeinit(void)
{
    return TRUE;
}
void ModeTestRun(uint16_t msgId)
{
    printf("mode test\n");
}
```

说明：init 函数内容主要包含硬件和内存初始化；Deinit 函数主要包含硬件和内存释放；run 函数是模式的运行主题，包含模式要实现的功能，间隔约 1MS 循环调用。

- 3) 模式数组里面增加一行模式定义

```
volatile SysModeStruct SysMode[]=
{
    #ifdef CFG_APP_IDLE_MODE_EN
    {ModeIdle, IdleModeInit, IdleModeDeinit, IdleModeRun, ModeStateReady},
    #endif
    #ifdef CFG_APP_BT_MODE_EN
    {ModeBtAudioPlay, BtPlayInit, BtPlayDeinit, BtPlayRun, ModeStateInit },
    #endif
    #ifdef CFG_APP_USB_PLAY_MODE_EN
    {ModeUDiskAudioPlay, MediaPlayInit, MediaPlayDeinit, MediaPlayRun, ModeStateReady },
    #endif
    #ifdef CFG_APP_CARD_PLAY_MODE_EN
    {ModeCardAudioPlay, MediaPlayInit, MediaPlayDeinit, MediaPlayRun, ModeStateReady },
    #endif
    #ifdef CFG_APP_LINEIN_MODE_EN
    {ModeLineAudioPlay, LineInPlayInit, LineInPlayDeinit, LineInPlayRun, ModeStateReady },
    #endif
    #ifdef CFG_APP_USB_AUDIO_MODE_EN
    {ModeUshDevicePlay, UshDevicePlayInit, UshDevicePlayDeinit, UshDevicePlayRun, ModeStateSusend},
    #endif
    #ifdef CFG_APP_I2SIN_MODE_EN
    {ModeI2SInAudioPlay, I2SInPlayInit, I2SInPlayDeinit, I2SInPlayRun, ModeStateReady },
    #endif
    #ifdef CFG_APP_RADIOIN_MODE_EN
    {ModeRadioAudioPlay, RadioPlayInit, RadioPlayDeinit, RadioPlayRun, ModeStateReady },
    #endif
    #if (BT_HFP_SUPPORT && defined(CFG_APP_BT_MODE_EN))
    {ModeBtHfPlay, BtHfInit, BtHfDeinit, BtHfRun, ModeStateSusend},
    #endif
    #ifdef BT_TWS_SUPPORT
    {ModeTwsSlavePlay, TwsSlavePlayInit, TwsSlavePlayDeinit, TwsSlavePlayRun, ModeStateSusend},
    #endif
    {ModeTest, ModeTestInit, ModeTestDeinit, ModeTestRun, ModeStateReady}
}
```

注意：init,deinit,run 如果是在别的文件定义的，请声明。

- 4) ModeNameStr 数组中添加 MODE debug 打印信息。注意：模式名字不要超过 15 个字符。

```
const SysModeStrAndRemind ModeNameStr[] =
{
    {ModeIdle, "Idle"},
#ifdef CFG_APP_BT_MODE_EN
    {ModeBtAudioPlay, "BtPlay"},
#endif
#ifdef CFG_APP_USB_PLAY_MODE_EN
    {ModeUDiskAudioPlay, "UDiskPlay"},
#endif
#ifdef CFG_APP_CARD_PLAY_MODE_EN
    {ModeCardAudioPlay, "CardPlay"},
#endif
#ifdef CFG_APP_LINEIN_MODE_EN
    {ModeLineAudioPlay, "LinePlay"},
#endif
#ifdef CFG_APP_USB_AUDIO_MODE_EN
    {ModeUsbDevicePlay, "UsbDevicePlay"},
#endif
#ifdef CFG_APP_I2SIN_MODE_EN
    {ModeI2SinAudioPlay, "I2SINPlay"},
#endif
#ifdef CFG_APP_RADIOIN_MODE_EN
    {ModeRadioAudioPlay, "RadioPlay"},
#endif
#ifdef BT_HFP_SUPPORT && defined(CFG_APP_BT_MODE_EN)
    {ModeBtHfPlay, "BtHfPlay"},
#endif
#ifdef BT_TWS_SUPPORT
    {ModeTwsSlavePlay, "TwsSlavePlay"},
#endif
    {ModeTest, "ModeTest"},
};
```

#### 5) 增加模式提示音

在 ModeTestInit() 函数增加下面语句：

```
RemindSoundServiceItemRequest(SOUND_REMIND_MODE_TEST, REMIND_PRIO_NORMAL);
```

SOUND\_REMIND\_MODE\_TEST 宏由提示音工具自动生成在 remind\_sound\_item.h 里面。提示音制作请查阅相关文档。

#### 6) 编译下载，按模式按键就能转到 ModeTest mode 了，转到后一直打印 modetest。

## 3. SDK 软件简要指南

### 3.1 山景 SDK 默认配置

MVSB5\_BT\_Audio\_SDK 默认是蓝牙音箱配置，并支持 U 盘/TF 卡播放功能。可以参考 app\_config.h 中的配置。

### 3.2. 新增功能范例

#### 3.2.1 TimeOut 用法

TimeOut 定时是 SDK 里面用得最多的一种定时方法，不占用系统硬件资源。

1、TIMERTimer\_Example; *//初始化*

2、TimeOutSet(&Timer\_Example, 10); *//设置定时时间为 10MS*

3、IsTimeOut(&Timer\_Example); *//查询定时到了没有，如时间到了，执行要定时的程序*

4、TimeOutSet(&Timer\_Example, 10); *//时间到了，复位定时时间，继续定时*

### 3.2.2 OS 互斥锁用法

互斥锁主要用来保护被不同任务都调用的临界区代码。保证一个任务调用临界区代码，另外的任务不能调用临界区代码。

- 1、osMutexId Mutex\_Example//定义互斥锁变量名
- 2、Mutex\_Example = xSemaphoreCreateMutex();//初始化
- 3、osMutexLock(Mutex\_Example);//加锁
- 4、程序临界区代码
- 5、osMutexUnlock(Mutex\_Example);//解锁

### 3.2.3 消息接收发送

消息主要用于任务之间传递消息，减少全局变量的使用。

- 1、MessageHandle msg\_Example;//定义消息队列
- 2、msg\_Example = MessageRegister(10);//初始化消息队列，最多存10个消息
- 3、//发送消息

```
{
    MessageContext    msgSend;//定义发送消息变量
    msgSend.msgId      = 0x8888;//赋值消息变量ID 值
    MessageSend(msg_Example, &msgSend);//发送消息
}
```
- 4、//接收消息

```
{
    MessageContext    msgRcv;//定义接收消息的变量
    MessageRcv(msg_Example, &msgRcv, 10);//（如果步骤3有执行，程序立刻收消息到msgRcv，并执行下一行代码；
    如果没有消息发过来，则接收消息的任务等待10MS，这10MS出让CPU
    给其他任务运行程序，10MS过后程序执行下一行代码）
}
```

## 4. 常见问题

请参考 MVSB5\_BT\_Audio\_SDK 系统应用开发 FAQ 说明文档\_Vx.x.pdf