

BP10 8Mb SDK 快速上手指南

V1.0

版本记录

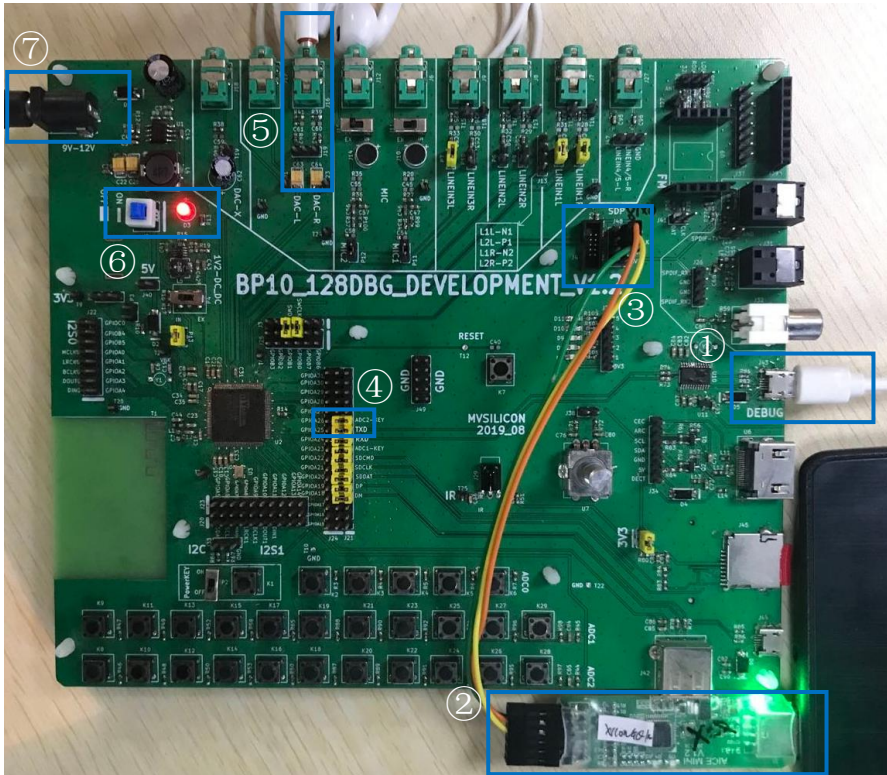
| 版本 | 修改日志 | 日期 | 修订 | 备注 |
|------|---------|------------|-----|----|
| V1.0 | 第一版正式发布 | 2021.09.03 | Ken | |
| | | | | |

目录

| | |
|--------------------------------------|----|
| BP10 8Mb 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 固件烧录与升级..... | 5 |
| 2. BP10 8Mb SDK 介绍..... | 7 |
| 2.1 MVsB1_BT_Audio_SDK_8Mb 系统框架..... | 7 |
| 2.2 MVsB1_BT_Audio_SDK_8Mb 目录介绍..... | 8 |
| 2.2.1 SDK 根目录..... | 8 |
| 2.2.2 BT_Audio_APP 目录..... | 8 |
| 2.2.2 App_framework 目录..... | 8 |
| 2.2.3 App_src 目录..... | 9 |
| 2.2.4 Driver 目录..... | 9 |
| 2.2.5 Middleware 目录..... | 10 |
| 2.2.6 Release 目录..... | 10 |
| 2.2.7 remind_res 目录..... | 10 |
| 2.2.8 Tools 目录..... | 11 |
| 2.3 运行程序..... | 11 |
| 2.4 软件运行流程..... | 12 |
| 2.4.1 概述..... | 12 |
| 2.4.2 main()说明..... | 12 |
| 2.5 主要宏所在文件介绍..... | 13 |
| 2.6 UART 打印..... | 13 |
| 2.6.1 软件模拟串口..... | 13 |
| 2.6.2 硬件串口..... | 14 |
| 2.7 按键..... | 14 |
| 2.8 显示功能..... | 14 |
| 2.9 调音功能..... | 14 |
| 2.10 蓝牙名修改..... | 14 |
| 2.11 Remind 提示音文件修改..... | 14 |
| 2.12 增加模式..... | 15 |
| 3. SDK 配置/修改简要指南..... | 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-BP10 开发板-串口打印-耳机-电源连接示意图

说明：①USB 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：

| | | | | |
|-----------|----------|----------|----------|-----|
| ② AICE | PIN4 TMS | PIN6 TCK | PIN3 GND | 不连 |
| ③ 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和连接电脑，MVSB1_BT_Audio_SDK_8Mb 默认打印 IO 为 TX:GPIOA6，波特率 512000。注意 A6 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 固件烧录与升级

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

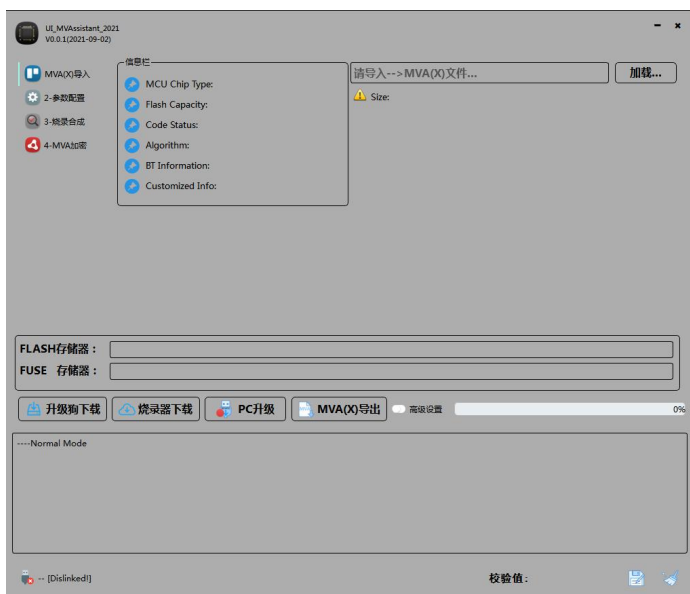
把编译生成或者上位机 MVAssistant 合成的.mva 文件拷贝到 U 盘/TF 卡根目录下；在开发板开机情况下插入即可升级；升级成功后会自动重启；

3) 升级狗和烧录器下载

上位机：MV_Assisant_2021_Vx.x.x

升级狗：新版本带屏升级狗

烧录器：新版本烧录器



（相关操作使用说明参见 [BP10 系列芯片 FLASH 升级和 KEY 加密及烧录_新工具链的操作说明 Vx.x.x.pdf](#)）

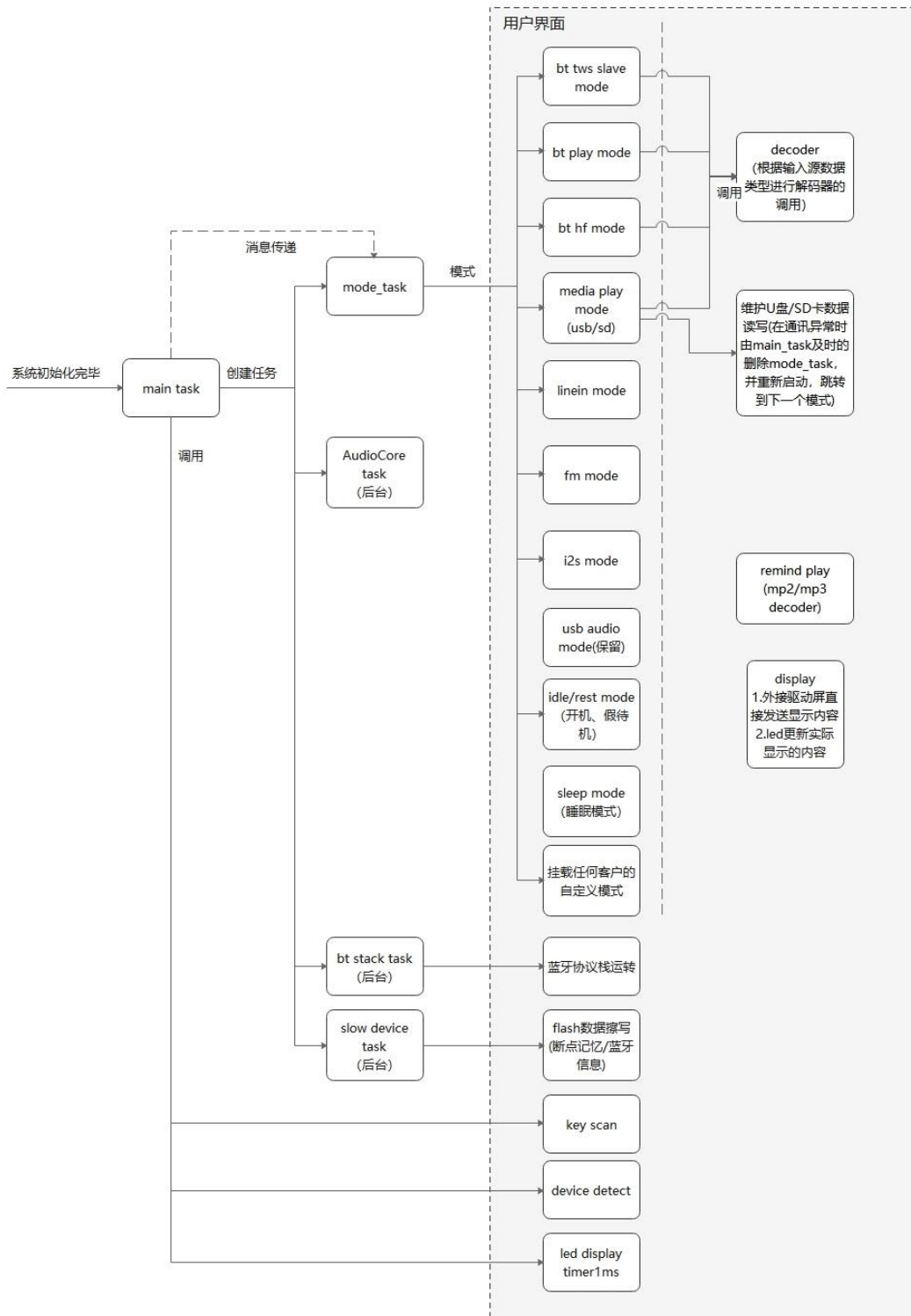
4) PC TOOLS 下载

在开发板开机状态下插入 USB 线；然后导入 SDK 生成的.mva 文件；点击下载即可



2. BP10 8Mb SDK 介绍

2.1 MVsB1_BT_Audio_SDK_8Mb 系统框架



2.2 MVsB1_BT_Audio_SDK_8Mb 目录介绍

2.2.1 SDK 根目录

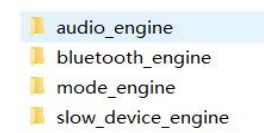


2.2.2 BT_Audio_APP 目录

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；



2.2.2 App_framework 目录



此部分主要为系统常驻的 4 个 task;

| | |
|------------------|---------------------|
| Audio_core task | 音频的输入输出控制和调节 |
| Bt_stack task | 蓝牙协议栈任务 |
| Mode task | 模式管理 |
| slow_device task | 低速设备任务-主要是 flash 擦写 |

2.2.3 App_src 目录

- app_mode_common
- app_mode_bt
- app_mode_i2s
- app_mode_idle
- app_mode_linein
- app_mode_media
- app_mode_radio
- app_mode_usb_audio
- components
- hmi
- power
- startup
- system_config
- main.c
- main_task.c
- main_task.h

| | |
|--------------------|--|
| app_mode_common | 为应用层提供系统接口；用于资源的申请释放 |
| app_mode_bt | 提供蓝牙免提、蓝牙 TWS，蓝牙播歌的用户交互应用 |
| app_mode_i2s | 提供 IIS 数字输入应用 |
| 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 |
| 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 |
| app_mode_media | TF 卡/U 盘的本地磁盘歌曲播放； |
| app_mode_radio | FM 收音机的应用与交互； |
| app_mode_usb_audio | USB 声卡的应用与交互 |
| components | 系统插件集合，audio 部分的交互处理；蓝牙的应用层接口 |
| hmi | 系统外设插拔检测部分、LED 屏显示、按键、FM 驱动部分 |
| power | 用于系统的休眠与唤醒及电池电量管理 |
| startup | 系统启动部分； |
| system_config | 系统主要配置头文件； |
| Main_task | 系统常驻 task，用于处理公共部分消息 |

2.2.4 Driver 目录

- driver
- driver_api
- readme.txt

底层驱动部分，用户不用太关注



2.2.5 Middleware 目录

- audio
- bluetooth
- fatfs
- flashboot
- flashfs
- Irc
- mv_utils
- rtc
- rtos

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

2.2.6 Release 目录

- app_framework
- app_src
- driver
- middleware
- output
- BT_Audio_APP.adx
- makefile
- objects.mk
- sources.mk

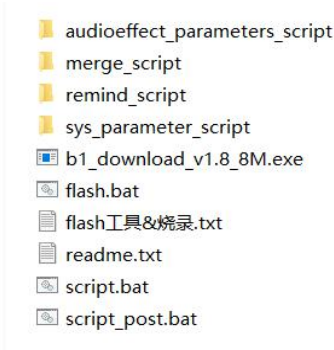
Release 是代码编译生成目标文件的目录；用于生产的、 文件都在此目录的 output 下；其他文件无需关注

2.2.7 remind_res 目录

- 16K
- BTmode.mp3
- cardmode.mp3
- connect.mp3
- disconnect.mp3
- DLguodi.mp3
- FMmode.mp3
- guanji.mp3
- i2smode.mp3
- kaiji.mp3
- LeftChannel.mp3
- RightChannel.mp3
- shangyishou.mp3
- shengkamode.mp3
- Upanmode.mp3
- Volmax.mp3
- xianlumode.mp3
- xiayisou.mp3

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

2.2.8 Tools 目录



此目录主要为一些批处理；方便更简洁的开发或转译

| | |
|-------------------------------|---|
| Audioeffect_parameters_script | 用于上位机调音文件导出后转译生成 SDK 可识别的数据文件 |
| Merge_script | 用于分配，提示音.bin; 蓝牙名.bin; bt.bin 等大小；用户也可自定义添加 |
| Remind_script | 用于打包提示音 |
| Sys_parameter_script | 用于打包蓝牙名 |
| Flash.bat | 用于打包生成可量产的.img/.mva 文件 |

2.3 运行程序

BT_Audio_APP 默认是一个蓝牙音箱的配置，并带有 TWS 功能。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 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：

```
[09:55:28.101]收←◆
/*****FlashBoot start!*****/

boot_code running.

mem size:0x2a5e0
flash space size:00040000 00040000
| flashboot Version: 3.0.3 |
Driver Version: 0.4.7 build @ Jul 28 2021 14:27:22 fe
check_user_code:00040000
ok
No Upgrade Request, jump_to_code_addr!

jump to sdk = 00040000

*****
|                               |
|           MvSilicon B1 SDK    |
| Mountain View Silicon Technology Co., Ltd. |
| SDK Version: 1.1.0           |
|                               |
*****
sys_clk =146765000
first use Powerkey!!
powerkey level high active
read flash Capacity = 0x200000 0x100000
remind_addr = 0xd3000
bp_data_addr = 0xf8000
bt_data_addr = 0xfa000
user_config_addr = 0xfe000
bt_config_addr = 0xff000
bt em size:16KB
正常启动
RstFlag = 3
power on reset
pin reset

Audio Decoder Version: 8.1.0 build @ Sep 1 2021 11:46:50
Driver Version: 0.4.7 build @ Jul 28 2021 14:27:22 fe
BtLib Version: 4.0.3 build @ Aug 30 2021 15:52:39
Fatfs presearch acc Lib Version: 1.4.0.STORY build @ May 25 2020 16:53:51

Load BP INFO START
MusicVolume:10,10
EffectMode:10,10
MicVolume:16,16
HfVolume:16,16
EqMode:0,0
ReverbStep:32,32
MusicIrebStep:7,7
MusicBassStep:7,7
power on mode BtPlay
[SYS]: Loading control vars as default
twss_mem_size:25724
AudioCore init
```

2.4 软件运行流程

2.4.1 概述

山景 BT_Audio_APP 程序嵌入了 RTOS，所以山景 BT_Audio_APP 软件系统是多任务系统。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，可以参照山景 SDK RTOS 相关用法先用，不必过于关注 RTOS 的底层和原理。

山景 BTAudioApp 程序运行流程：

RomBoot → Flashboot → Main() → Main Task → Mode/Audio core/Slow device/Bt stack Task

下面讲解进入 main()函数后程序是怎么依次运行的，请打开程序。

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 主要宏所在文件介绍

山景 SDK 很多功能都是通过宏开关控制。下面简单介绍 App_config.h 和 Bt_config.h。

App_config.h 里面主要是系统和音频相关的宏

如：

```

CFG_CHIP_BP10128 //定义芯片类型
CFG_APP_USB_PLAY_MODE_EN //USB 模式开关宏
USE_MP3_DECODER //支持 MP3 解码宏
CFG_FUNC_AUDIO_EFFECT_EN //音效开关宏

```

Bt_config.h 里是与蓝牙设置相关的宏

如：

```

#define BLE_SUPPORT DISABLE //是否支持 BLE
#define BT_NAME "BP10_BT" //蓝牙设备名字

```

2.6 UART 打印

山景串口打印支持软件模拟串口和硬件串口。打印函数是 DBG(format, ...)。

2.6.1 软件模拟串口

模拟串口可以定义任意 IO 口打印，用起来比较方便，但是比较耗 CPU 资源，建议在没有硬件串口用的情况下，用软模拟串口打印调试信息。

App_config.h 里相关宏设置：

```

#define CFG_FUNC_DEBUG_EN //定义调试串口宏，开启串口打印
#define CFG_USE_SW_UART //开启软件串口打印，开启后硬件串口打印关闭
#define SW_UART_IO_PORT SWUART_GPIO_PORT_A //设置为 A 口打印
#define SW_UART_IO_PORT_PIN_INDEX 1 //设置 A1 为软件打印 PIN
#define CFG_SW_UART_BANDRATE 460800 //波特率设置

```

2.6.2 硬件串口

硬件串口固定为几个 IO 输出，但是耗 CPU 时间较少，建议一般用硬件串口打印。山景 BT_Audio_APP 8Mb 默认硬件串口 A6 输出打印信息。

App_config.h 里相关宏设置：

```
#define CFG_FUNC_DEBUG_EN           //定义调试串口宏，开启串口打印
#define CFG_UART_TX_PORT (0)       //0: A6 输出打印, 1: A10 输出打印, 2: A25 输出打印
#define CFG_UART_BANDRATE 512000 //波特率设置
```

2.7 按键

山景 SDK 支持五种类别的按键，分别是普通 ADC 按键，电位器，编码开关，IO 按键，遥控按键。

所有按键程序流程都一样，如下：

KeyInit() → KeyScan() → MessageSend(mainAppCt.msgHandle...) → MessageSend(SysModeCt.msgHandle...)

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(SysModeCt.msgHandle...); //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；

2.8 显示功能

BT_Audio_APP，App_config.h 里面打开宏 CFG_FUNC_DISPLAY_EN，可开启 LED 显示功能。注意山景 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，并放入 TCM。

2.9 调音功能

BT_Audio_APP，App_config.h 里面打开宏 CFG_FUNC_AUDIO_EFFECT_EN，使能 SDK 音效功能。

山景 BP10 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 BP10 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《BP10_8Mb_SDK 音效模块使用文档》。

2.10 蓝牙名修改

在目录..BT_Audio_APP\tools\sys_parameter_script 的 parameter.ini 修改

```
; 这里是注释
; 第一个参数是长度，第二个以后是值

bt_LocalDeviceName = 40, "BP10 BT 8Mb"
ble_LocalDeviceName = 40, "BP10_BT_8Mb_Ble"

; 下面是测试数据
test_addr = 6, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66
bt_gain   = 1, 15
bt_vol    = 2, 0x20
```

2.11 Remind 提示音文件修改

1) 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：提示音文件转换成 MP3/44.1K 单声道音频并替换原来的音频即可；

新增：如果要新增提示音只要把转换好的提示音放在此目录下即可；

删除：直接删除不需要的提示文件即可

注：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；修改后需关注 bin 文件大小；并 4K 对齐填入大小即可；

如 all.bin 为 141K；则 $(141+3) * 1024 = 0x24000$ ；多+3 是为了 4K 对齐；

\\> Data (D:) > MV_test_SDK > MVsB1_BT_Audio_SDK_8Mb > BT_Audio_APP > tools > merge_script

| 名称 | 修改日期 | 类型 | 大小 |
|-----------------------|-----------------|--------|--------|
| bp_data.bin | 2021/8/23 10:57 | BIN 文件 | 8 KB |
| bt_data.bin | 2021/8/23 10:57 | BIN 文件 | 8 KB |
| merge.ini | 2021/8/23 10:57 | 配置设置 | 1 KB |
| merge_bin.exe | 2021/8/23 10:57 | 应用程序 | 62 KB |
| MiddleConfigTools.exe | 2021/8/23 10:57 | 应用程序 | 944 KB |

merge.ini - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
;table info start
[table]
maxlen = 0x1000
;table info stop

; package info start

[remind]
virtual = 0
fullpath=..\tools\remind_script\all.bin
maxlen = 0x25000
enable = 1

[bp_data]
virtual = 0
fullpath=..\tools\merge_script\bp_data.bin
maxlen = 0x2000
enable = 1

[bt_data]
virtual = 0
fullpath=..\tools\merge_script\bt_data.bin
maxlen = 0x2000
''
```

2.12 增加模式

1) 枚举类型 SysModeNumber 里增加一个值

```
typedef enum
{
    ModeIdle= 0,
    ModeBtAudioPlay,
    ModeUDiskAudioPlay,
    ModeCardAudioPlay,
    ModeLineAudioPlay,
    ModeUsbDevicePlay, //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
    {ModeUsbDevicePlay, 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
    #if (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_SOUND_INTERRUPT_PLAY|REMIND_SOUND_NEED_MUTE_APP_SOURCE);
```

SOUND_REMIND_MODE_TEST 宏由提示音工具自动生成,在 remind_sound_item.h 里面可以找到。提示音制作请查阅相关文档。

6) 编译, 下载, 按转模式键就能转到 ModeTest mode 了, 转到后一直打印"modetest"

3. SDK 配置/修改简要指南

3.1 山景 SDK 默认配置

BP10 8Mb SDK 默认是蓝牙音箱配置, 并支持 TWS 功能

3.2. 新增功能范例

3.2.1 TimeOut 用法

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

1. TIMER Timer_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《8Mb SDK 系统应用开发 FAQ 说明文档_V x.x》

(---结束---)