音效模块使用文档

V1.5

版本记录：

版本号	日期	作者	备注
V1.0	2021-9-3	Sam	初稿
V1.1	2021-12-22	Sam	1、 音效文件增加，添加相关描述
V1.2	2022-3-7	Sam	1、增加通过调音工具在线保存调音参数的功能
V1.3	2022-3-23	Sam	1、 在线保存调音参数到 flash 流程更新 2、 添加在线调音框图
V1.4	2022-5-16	Sam	1、脚本名称更新
V1.5	2022-8-10	Sam	1、添加用户 Key 管理的音效处理代码

目录

1 芯片资源介绍

山景 BP10 系列芯片拥有丰富的音频外设资源和强大的运算处理能力。

- 高性能 32 位处理器，支持 DSP 指令和 FPU 浮点运行指令。
- 芯片主频 288MHz
- 内置 320KB RAM
- 内置 32KB I-Cache 和 32KB D-Cache
- 音频资源（最多支持）：
  - 2 路立体声 ADC
  - 2 路全双工 I2S
  - 1 路立体声 DAC 和一路单声道 DAC
  - spdif 光纤同轴输入输出
  - BT 无线输入
  - USB 声卡输入输出
  - USB U 盘输入/TF 卡输入

强大的芯片硬件资源，配合灵活软件架构可以实现多通路混音和多种音效处理。BP10 芯片支持的音效资源可参见 SDK 代码包 audio_effect_api.h 中的宏说明。BP10 芯片音效选择和参数调整可以通过 USB Device 接口连接 PC，通过调音工具- ACPWorkbench 实现实时的在线调音功能，帮助用户快速落地各种特色的音频产品。

2 SDK 音效设计架构

BP10 8Mb SDK 音效和调音的软件设计架构如下图所示。

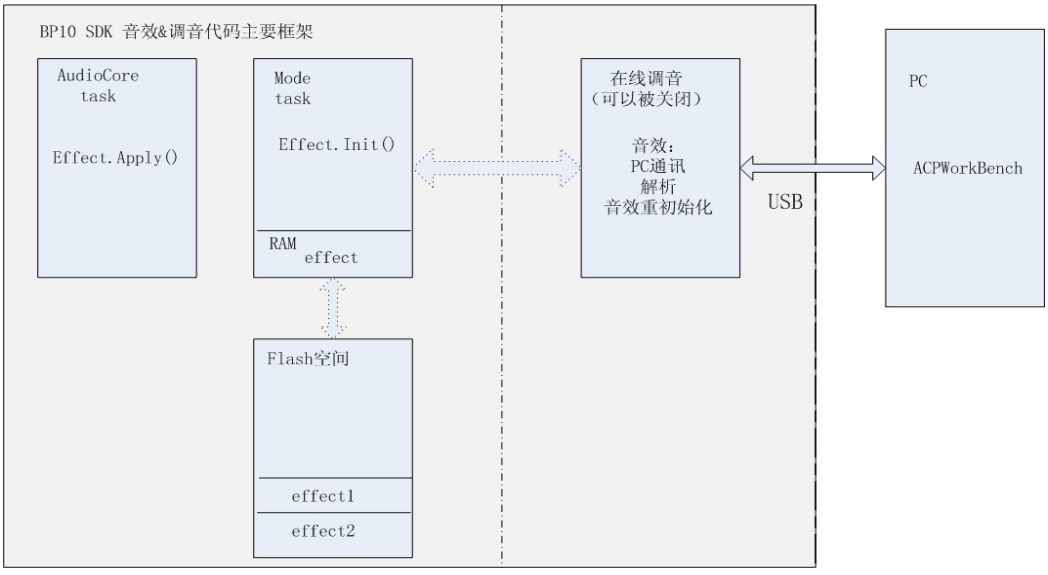

图 1 音效&音架构框图

BP10 8Mb SDK 以 AudioCore（音频流处理）为核心，实现灵活多变的音频音效处理。音效和调音的软件代码层次清晰，高内聚低耦合；将用户十分关注，需要经常修改的部分独立出来，方便客户进行二次开发；提供多个脚本工具，完成音效 C 文件和音效 Bin 文件的自动转换生成工作，加速用户二次开发进度。

BP10 8Mb SDK 调音功能主要有 2 分大部分：

1、支持在线调音功能。通过宏 CFG_FUNC_AUDIO_EFFECT_ONLINE_TUNING_EN 来控制，该功能开启时可以连接 PC 调音工具--ACPWorkbench，实现实时在线调音功能。通过 ACPWorkbench 调音工具可以调整音效参数，开启音效，关闭音效。在线调音宏可以关闭，关闭之后可以节省代码和内存开销。

注意：不能通过调音工具直接新增音效，新增音效必须通过修改固件来实现。

2、支持 SDK 加载 flash 固定区域的音效参数，通过 CFG_EFFECT_PARAM_IN_FLASH_EN 功能宏来控，这样音效参数 bin 文件可以通过量产工具烧录到 flash 指定地址，并且可以只更新音效参数 bin 文件。支持调音工具 ACPWorkbench 将音效参数更新保存到 flash 音效参数区域，该功能通过宏 CFG_EFFECT_PARAM_UPDATA_BY_ACPWORKBENCH 来控制。以上功能可以方便客户的调音工程脱离软件工程师，进行快速调音，对比音效效果，快速确定产品的音效效果。

注意：CFG_EFFECT_PARAM_UPDATA_BY_ACPWORKBENCH 功能宏可以关闭，关闭之后不能通过 ACPWorkbench 调音功能写音效参数到 flash 区域，可以节省 4156 字节 RAM。

2.1 音效&调音的文件

BP10 8M SDK 的音效功能框图如图 2 所示。

SDK 支持的音效由该图决定，根据该图会产生一个音效列表 C 文件和 h 头文件，即“comm_param.c”和“comm_param.h”两个核心文件。

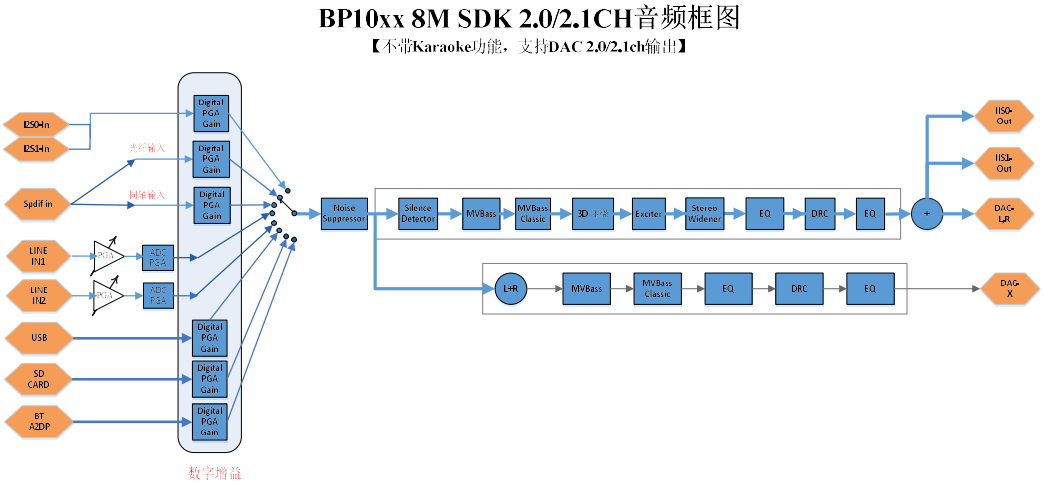

图 2 BP10 SDK 音效功能框图

BP10 8M SDK 通过音效功能宏来控制音效，便于用户开关宏来调试音效，量产时关闭部分宏来节省代码和内存的使用。具体见表 1 所描述。

表 1 SDK 音效主要宏定义和说明

宏定义	说明
CFG_FUNC_AUDIO_EFFECT_EN	音效宏总开关
CFG_FUNC_AUDIO_EFFECT_ONLINE_TUNING_EN	在线调音功能宏
CFG_EFFECT_PARAM_IN_FLASH_EN	支持 flash 音效区域加载音效参数
CFG_EFFECT_PARAM_UPDATA_BY_ACPWORKBENCH	通过在线调音工具更新 flash 音效区域的音效参数

SDK 中的音效和调音相关的文件如表 2 所示。

表 2 音效和调音 C 及说明

音效文件和目录	说明
communication.c communication.h	在线调音功能代码
audio_effect_param.c audio_effect_param.h	在线调音音效结构体，支持在线调音动态修改参数；由脚本工具 audioeffect_list_to_parameters_script 根据 comm_param.c 自动生成。
audio_effect_api.c audio_effect_api.h	音效 API，隔离音效库函数，统一应用层接口，方便用户理解和使用。
audio_effect.c audio_effect.h	音效实现应用层代码
audio_effect_process.c	音效框图具体实现 C 文件，AudioCore task 中会调用；用户需要根据实际需要进行修改
comm_param.c comm_param.h	参数列表，用户关心的核心文件
ctrlvars.c ctrlvars.h	音频硬件通路的数据结构；变量初始化
music_parameter（目录） CommParam.c AECParam_HFP.c	PC 调音工具导出的音效数组 C 文件经过脚本 audioeffect_parameters_script 工具生成。该文件必须和 comm_param.c 中的音效列表中音效顺序必须是完全一致，否则会系统异常，比如复位。
audio_effect_flash_param.c audio_effect_flash_param.h	flash 音效参数区域存储读取实现代码
audio_effect_user.c audio_effect_user.h	用户通过 Key 控制音效的 API 实现代码
eq_params.c eq_params.h	5 种用户 Key 控制音乐 EQ 音效参数文件

音效相关的文件和层次关系大致如图 3 所示。

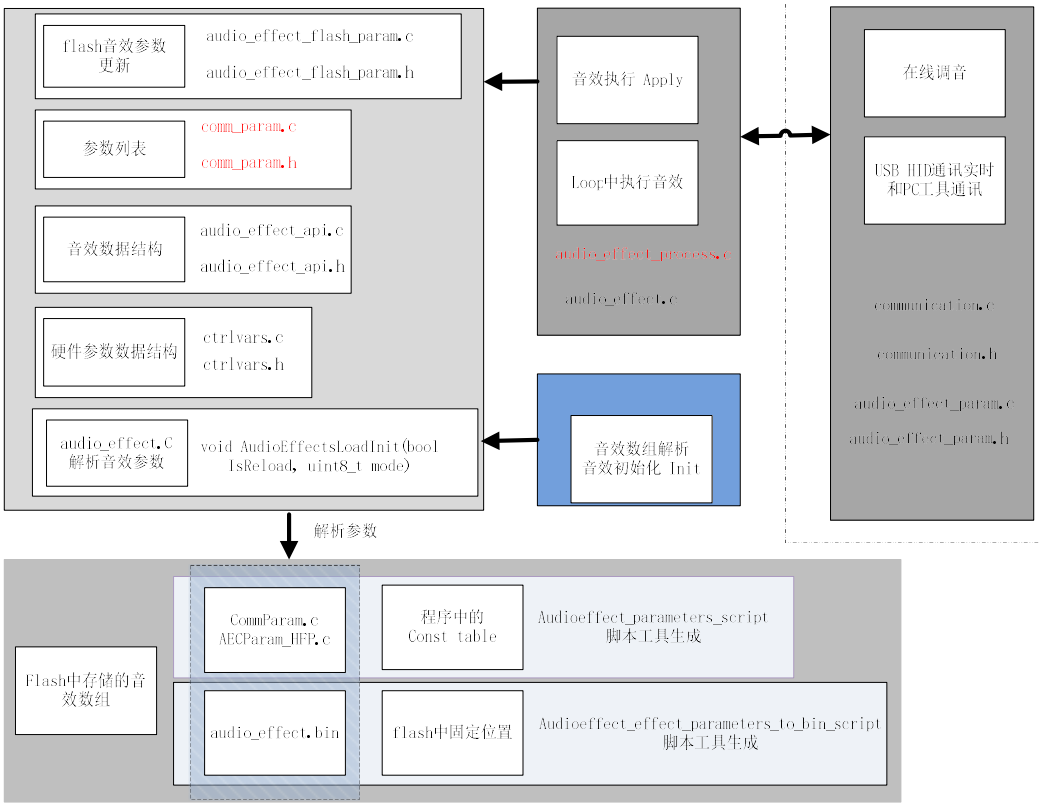

图 3 音效&调音文件结构

2.1.1 音效参数列表文件

comm_param.c 文件是音效的调音参数列表文件。文件的内容和顺序由音效功能框图（图 2）来生成。该文件用户需要关心，如果用户产品的音效列表顺序和个数发生了变化，那么用户需要调整数组中的内容。

```
const EffectComCt AudioEffectCommParam[]=
{
    {2, NodeType_Normal,    GAIN_CONTROL,           "1:BT Play Gian",           0xFF},//0x81
    {2, NodeType_Normal,    GAIN_CONTROL,           "1:USB Play Gian",          0xFF},//0x82
    {2, NodeType_Normal,    GAIN_CONTROL,           "1:SD Play Gian",           0xFF},//0x83
    {2, NodeType_Normal,    GAIN_CONTROL,           "1:I2S Gian",               0xFF},//0x84
    {2, NodeType_Normal,    GAIN_CONTROL,           "1:OPTICAL Gian ",          0xFF},//0x85
    {2, NodeType_Normal,    GAIN_CONTROL,           "1:COAXIAL Gian ",          0xFF},//0x86
    {2, NodeType_Normal,    EXPANDER,               "2:Music Noise Suppressor", 0xFF},//0x87
    {2, NodeType_Bifurcate, SILENCE_DETECTOR,       "2:Music Silence Detector", 0xFF},//0x88
    {2, NodeType_Normal,    VIRTUAL_BASS,           "2:Music VB",               0xFF},//0x89
    {2, NodeType_Normal,    VIRTUAL_BASS_CLASSIC,   "2:Music VB Classic",       0xFF},//0x8A
    {2, NodeType_Normal,    THREE_D,                "2:Music 3D",               0xFF},//0x8B
    {2, NodeType_Normal,    EXCITER,                "2:Music Exciter",          0xFF},//0x8C
    {2, NodeType_Normal,    STEREO_WIDENER,         "2:Music Stereo Widener",   0xFF},//0x8D
    {2, NodeType_Normal,    EQ,                     "2:Music Pre EQ",           0x8E},//0x8E
    {2, NodeType_Normal,    DRC,                    "2:Music DRC",              0xFF},//0x8F
    {2, NodeType_Normal,    EQ,                     "2:Music EQ",               0x90},//0x90
    {1, NodeType_Normal,    VIRTUAL_BASS,           "3:Music DACX VB",          0xFF},//0x91
    {1, NodeType_Normal,    VIRTUAL_BASS_CLASSIC,   "3:Music DACX VB Classic",  0xFF},//0x92
    {1, NodeType_Normal,    EQ,                     "3:Music DACX Pre EQ",      0xFF},//0x93
    {1, NodeType_Normal,    DRC,                    "3:Music DACX DRC",         0xFF},//0x94
    {1, NodeType_Normal,    EQ,                     "3:Music DACX EQ",          0xFF},//0x95
};
```

图 4 音效参数列表内容

这张表的每一行表示一个音效节点，音效节点数据结构如图 5 所示。各个参数的含义分

别如下：

- channel 表示该音效的声道数，1 表示单声道数据，2 表示立体声数据；
- AudioNodeType 表示这个音效处于整个音效网络节点的音效类型。
- AudioEffectType 表示音效的类型。
- EffectName 为音效的名称指针字符串，该字符串以数字+冒号开始，比如“3：Music DACX VB”，这里的“3”表示这个音效为第三组音效，音效名称为“Music DACX VB”。如果整个音效网络只有一组音效，那么就以“1：”开头。
- Index 为音效的实际编号，主要用于实现用户按键控制的音效功能，比如用户需要通过按键控制 EQ 模式（Index：0x90），则将该值设置为 0x90，SDK 中音效 apply 时会根据 index 来实现具体的按键切换 EQ 模式功能。默认值配置为 0xFF，不需要做用户按键处理。

```
23
24  typedef struct _EffectComCt
25  {
26  uint8_t     channel;
27  NodeType    AudioNodeType;
28  EffectType  AudioEffectType;
29  char*       EffectName;
30  uint8_t     Index;
31  }EffectComCt;
```

图 5 音效参数节点数据结构

comm_param.c 文件为音效调音部分的核心文件，用户需要重点关注。假如用户需要修改音效，对音效内容和音效顺序做出新调整，用户首先需要重新定义和排列音效参数表，并将新的 comm_param.c 文件放到 tools\audioeffect_list_to_parameters_script\source_file 目录下,点击运行 audioeffect_list_to_parameters_script.bat，脚本会自动生成 3 个*.c 和 1 个*.h 文件，将这 4 个文件分别 copy 并替换 SDK 包中对应的文件。注意生产的调音参数文件可能和 SDK 中默认名称有所不同，如果不同则手动修改一下文件名称和文件内定义的数组名。

注意：

1、该脚本生成的音效参数文件 CommParam.c 和 AECParam_HFP.c 文件中的音效参数值均为默认值，如果之前的音效已经调试，需要用代码比较工具对比之后再将上述 2 个文件 copy 到对应的 SDK 目录下。

2、关注 audio_effect.h 中 2 个宏。

```
20  #define AUDIO_EFFECT_GROUP_NUM          3//音效列表组数
21  #define AUDIO_EFFECT_NODE_NUM           11//一组音效最大个数
```

这里 AUDIO_EFFECT_GROUP_NUM 为 3，对应于 comm_param.c 中音效一共 3 组，AUDIO_EFFECT_NODE_NUM 为 11，表示 3 组音效中最大一组音效的数量不能超过 11 个音效。如果说用户的音效参数表修改之后，需要根据情况调整这 2 个宏的值。

2.1.2 调音音效数组文件

调音音效数组 C 文件是调音工具导出之后经过 audioeffect_parameters_script 脚本转译后生成的 C 文件，SDK 中的调音文件统一放到..\audio\music_parameter 目录下。BP10 8Mb SDK 中默认的“CommParam.c”文件音效 Index 从 0x81 开始，最大值为 0x95，一

共 21 个音效参数（不同 SDK，这个最大值会不同）。这个文件的音效个数和顺序必须和音效参数列表文件中的音效内容以及顺序完全的一致（如果脚本生成肯定是一致的，如果手动修改需要注意一致性问题）。该文件包含了 2 部分内容，芯片音频硬件的参数配置数组以及音效参数数组，文件中以 0x81 开始到文件结尾处为音效的参数数组数据，0x81 之前的数据为芯片硬件参数和系统配置参数。图 6 所示为“CommParam.c”文件的部分内容，完整内容可以阅读“CommParam.c”以及“AECParam_HFP.c”文件。

0x87 表示为所有音效中的第七个音效，音效功能为 Nosie suppressor，和图 3 中的第七个音效是一致的（必须一致，否则会导致系统异常）；0x00 表示这个音效默认处于未使能状态；数据区域一共 8Byte，2Byte 组合成一个音效参数，Nosie suppressor 音效共有 4 个参数。0x88 表示为所有音效中的第八个音效，音效功能为 Silence Detector，状态使能，这个音效只有一个参数。各个音效参数的具体含义可以参考调音工具包下的《固件与用户应用程序通信协议.pdf》。

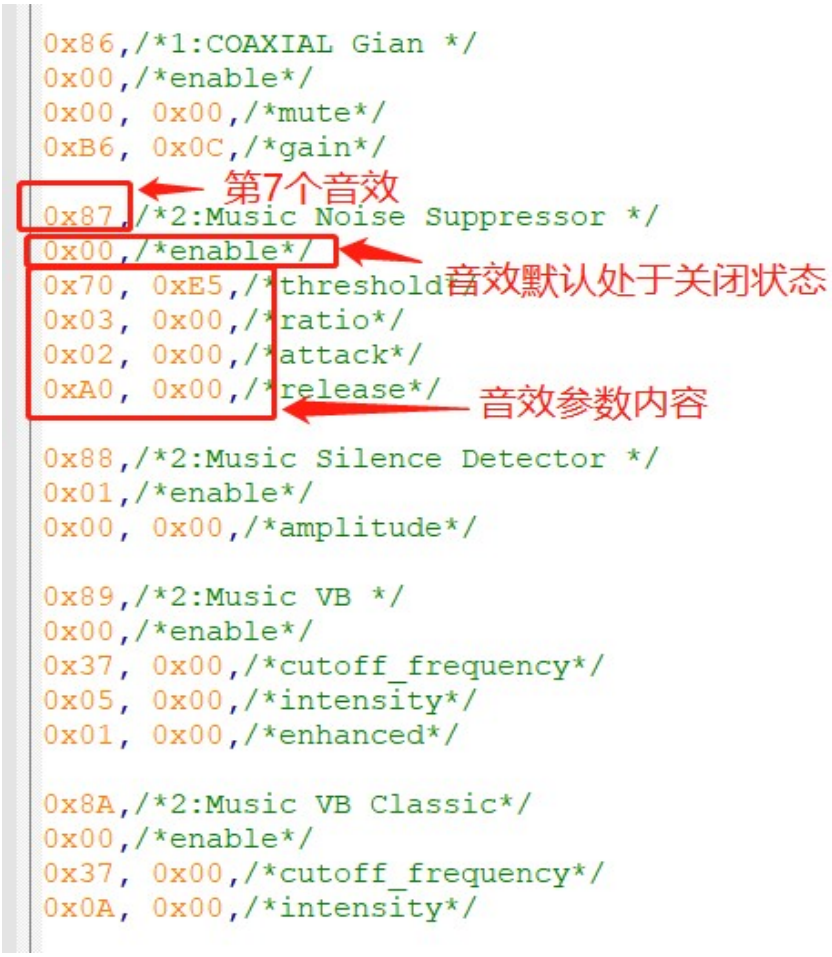

图 6 音效参数数组内容

2.2 音效部分

主要分 2 个步骤，1、音效解析并初始化，2、音效执行。

1、 音效初始化

模式切换或者按下音效模式键触发音效切换时。SDK 会导入新的音效参数数组，比如图 2 中的的 CommParam.c 中的 CommParam []，SDK 解析音效类型和音效参数，这个过

程中会根据音效类型来动态申请内存，然后初始化相关的音效。

2、 音效执行

音效执行处理，AudioCore 运转过程中会调用到对应 AudioProcess()函数，比如 8M SDK 有 2 个音效处理函数，AudioMusicProcess()和 AudioEffectProcessBTHF()，前者为音乐模式下的音效处理函数，后者为蓝牙通话下的音效处理函数。AudioProcess()函数内部会根据定义的音效已经音效使能状态分别执行各个音效的 effectApply()函数，从而实现相关的音效算法。

2.3 调音部分

BP10 8Mb SDK 实现了通过 USB HID 接口，让用户通过 PC 调音工具实时在线的方式来调整音效参数的功能。终端用户可以利用该功能快速方便的调试出适合自己产品的音效。

ACPWorkBench 调音工具版本为 v2.30.2 以及以后的版本可以在调音工具界面上显示出当前音效的名称，如果图 8 截图所示的橙色部分，当前的音效名称为“2：Music_Flash”。用户可以通过修改 SDK 代码 ctrlvars.c 文件中的代码来实现音效名称的修改，如图 7 所示。

```
29
30
31  ControlVariablesContext gCtrlVars;
32
33  #ifdef CFG_EFFECT_PARAM_IN_FLASH_EN
34  const AUDIO_EFF_PARAMAS EFFECT_TAB_FLASH[CFG_EFFECT_PARAM_IN_FLASH_ACTIVCE_NUM]={
35      {EFFECT_MODE_FLASH_HFP_AEC,          NULL,    sizeof(AECParam_HFP),    (uint8_t *)AudioEffectAECParam_HFP,  "0:HFP_AEC_FLASH"},
36      {EFFECT_MODE_FLASH_USBPHONE0_AEC,    NULL,    sizeof(AECParam_HFP),    (uint8_t *)AudioEffectAECParam_HFP,  "1:USBPHONE_AEC_FLASH"},//预留目前未使用
37      {EFFECT_MODE_FLASH_Music,            NULL,    sizeof(CommParam),       (uint8_t *)AudioEffectCommParam,     "2:Music_FLASH"},
38      {EFFECT_MODE_FLASH_Movice,           NULL,    sizeof(CommParam),       (uint8_t *)AudioEffectCommParam,     "3:Movice_FLASH"},
39      {EFFECT_MODE_FLASH_News,             NULL,    sizeof(CommParam),       (uint8_t *)AudioEffectCommParam,     "4:News_FLASH"},
40      {EFFECT_MODE_FLASH_Game,             NULL,    sizeof(CommParam),       (uint8_t *)AudioEffectCommParam,     "5:Game_FLASH"},
41  };
42  #endif
43
44  //当前SDK支持的音效参数列表
45  const AUDIO_EFF_PARAMAS EFFECT_TAB[EFFECT_MODE_NUM_ACTIVCE]={
46      {EFFECT_MODE_NORMAL,          CommParam,        sizeof(CommParam),       (uint8_t *)AudioEffectCommParam,     "Music"},
47  #if (BT_HFP_SUPPORT == ENABLE)
48      {EFFECT_MODE_HFP_AEC,         AECParam_HFP,     sizeof(AECParam_HFP),    (uint8_t *)AudioEffectAECParam_HFP,  "HFP_AEC"},
49  #endif
50  };
```

图 7

用户保存调好的音效有 2 种方式可供选择。

1、调好的音效通过 PC 工具 ACPWorkbench 导出 c 文件，放到 SDK “tools\audioeffect_parameters_script\source_file”目录下，然后调用脚本 audioeffect_parameters_script.bat，会自动将新生成的音效 C 文件 copy 到 SDK “app_src\components\audio\music_parameter”目录下。用户再次编译 SDK 并下载到 flash 中，新的音效参数就生效了。

2、调好的音效通过 PC 工具 ACPWorkbench 直接保存到 flash 指定区域。该功能需要开启固件代码中的宏 CFG_EFFECT_PARAM_IN_FLASH_EN 和宏 CFG_EFFECT_PARAM_UPDATA_BY_ACPWORKBENCH。程序正常运行之后，点击 ACPWorkbench 工具的“Download”选项，选择”Save Configurations to Flash”，如图 8 红色框 1 和红色框 2 所示。之后在弹出的对话框里选择“Use the save parameter command(0xFD)”,如图 9 所示，并点图中“Next”选项进入下一步。最后如图 10 所示，点击“Save”选项，用户的音效参数会保存到目标 flash 存储空间中。

注意：方式 2 需要使用“audioeffect_effect_c_to_bin_script”脚本工具先将目标音效参数 C 文件转换成 bin 件，或者使用“audioeffect_effect_ini_to_bin_script”脚本工具先将目标音效参数 ini 文件转换成 bin 件，然后使用“merge_script”脚本工具合并成 flash.img 文件烧录到 flash 中才可以使用。Flash 音效固定参数区域默认占用 16K 空间，可以根据需要修改。

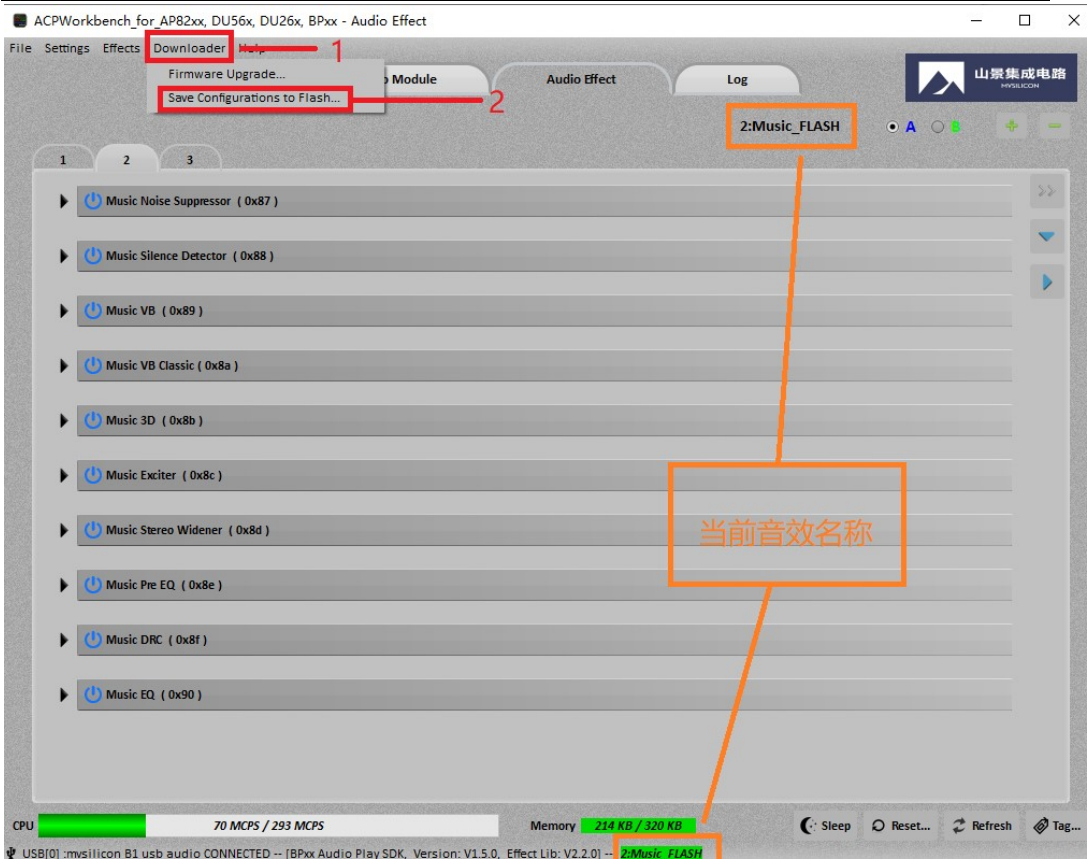
ACPWorkbench_for_AP82xx, DU56x, DU26x, BPxx - Audio Effect
File Settings Effects Downloader Help
1
Firmware Upgrade...
Save Configurations to Flash...
2
Audio Module
Audio Effect
Log
2:Music_FLASH
Music Noise Suppressor (0x87)
Music Silence Detector (0x88)
Music VB (0x89)
Music VB Classic (0x8a)
Music 3D (0x8b)
Music Exciter (0x8c)
Music Stereo Widener (0x8d)
Music Pre EQ (0x8e)
Music DRC (0x8f)
Music EQ (0x90)
当前音效名称
CPU
70 MCPS / 293 MCPS
Memory
214 KB / 320 KB
Sleep
Reset...
Refresh
Tag...
USB[0] :mvsilicon B1 usb audio CONNECTED -- [BPxx Audio Play SDK, Version: V1.5.0, Effect Lib: V2.2.0] -- 2:Music_FLASH

图 8

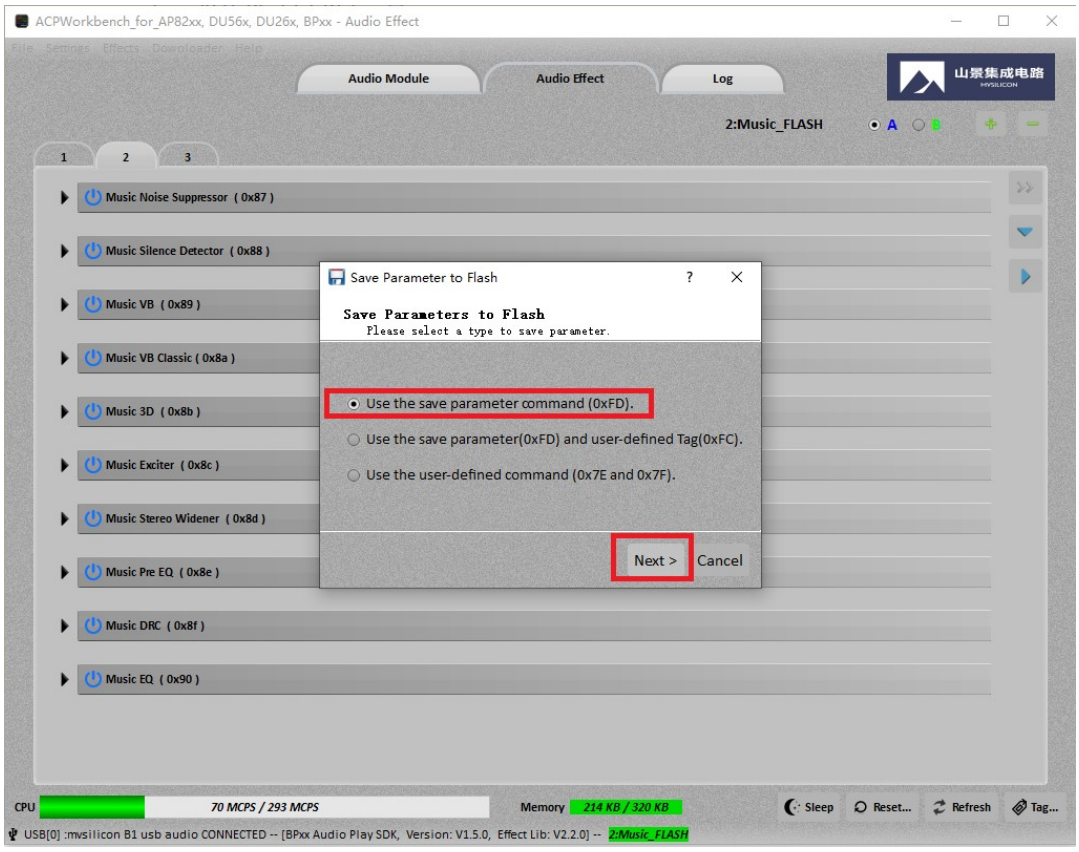
ACPWorkbench_for_AP82xx, DU56x, DU26x, BPxx - Audio Effect
Audio Module
Audio Effect
Log
2:Music_FLASH
Save Parameter to Flash
Save Parameters to Flash
Please select a type to save parameter.
Use the save parameter command (0xFD).
Use the save parameter(0xFD) and user-defined Tag(0xFC).
Use the user-defined command (0x7E and 0x7F).
Next >
Cancel
CPU
70 MCPS / 293 MCPS
Memory
214 KB / 320 KB
Sleep
Reset...
Refresh
Tag...
USB[0] :mvsilicon B1 usb audio CONNECTED -- [BPxx Audio Play SDK, Version: V1.5.0, Effect Lib: V2.2.0] -- 2:Music_FLASH

图 9

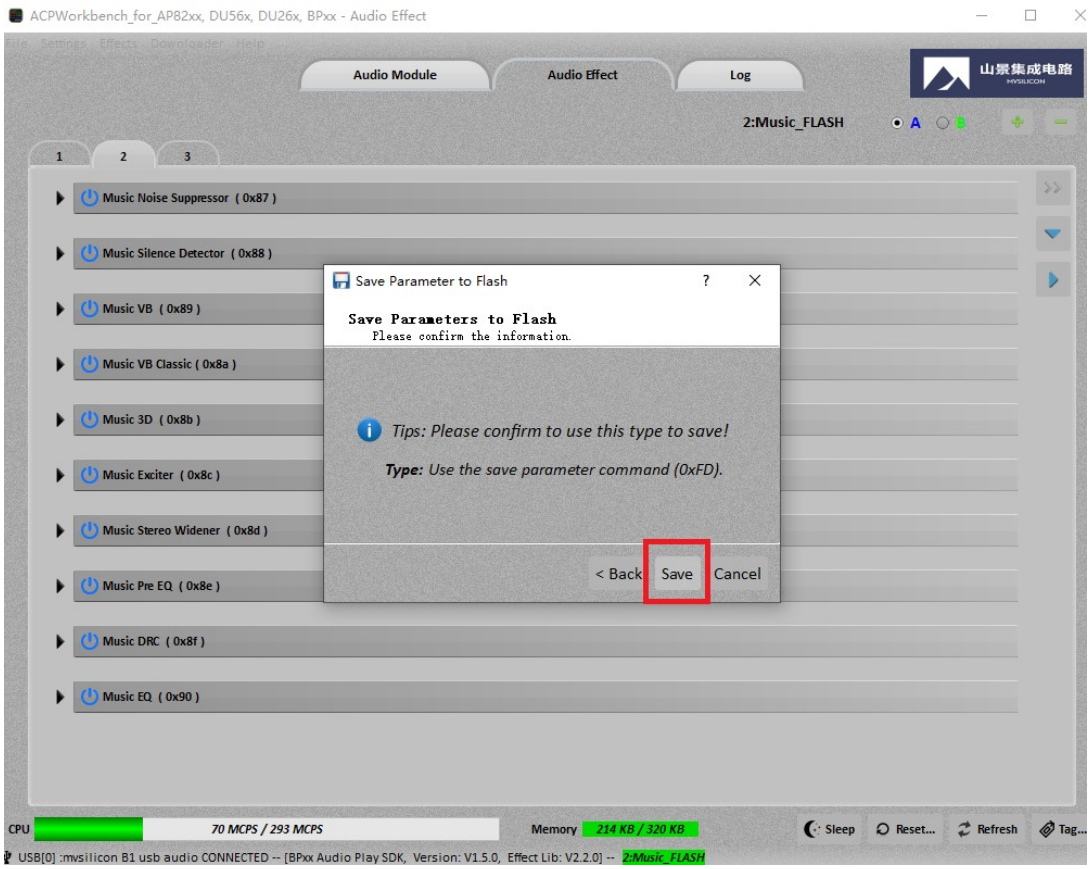

图 10

2.3.1 调音

山景提供了在实时线调音功能，关于调音工具的使用说明请仔细阅读《ACPWorkbench-CHS.pdf》，也提供了英文版本手册。另外工具包内提供了《固件与用户应用程序通信协议.pdf》，用户有兴趣可以进一步详细阅读。

3 用户快速定制

3.1 宏的选择

BP10 8Mb SDK 中对于各种音效用宏进行了管理，当某些音效确定不会使用时，将 audio_effect_api.h 文件中对应音效的宏配置为“0”，这样这部分代码以及相关的音效库函数均不会被包含到 SDK 代码中来，这样可以减少代码量。

注意：音效宏打开或者关闭之后，需要确认修改 comm_param.c 文件中的音效列表，以及对应的音效模式 C 文件中的音效顺序。

3.2 如何新增一个音效

3.2.1 新增音效为已有音效

这种情况下修改相对简单，修改 comm_param.c 文件中的音效列表内容，在合适的位置添加上用户想要添加的音效。比如在图 11 红框的 EQ 后面添加一个 Silence Detector。

```
const EffectComCt AudioEffectCommParam[]=
{
    {2, 0, EXPANDER,          "1:Music Noise Suppressor "},
    {2, 1, SILENCE_DETECTOR,"1:Music Silence Detector "},
    {2, 2, GAIN_CONTROL,      "1:Music Pre Gain "},
    {2, 0, EQ,                "1:Music Pre EQ "},
    {2, 0, VIRTUAL_BASS,      "1:Music VB "},
    {2, 0, THREE_D,           "1:Music 3D "},
    {2, 0, DRC,               "1:Music DRC "},
    {2, 0, EQ,                "1:Music EQ "},
    {1, 3, GAIN_CONTROL,      "2:Music DACX Pre Gain "},
    {1, 0, EQ,                "2:Music DACX Pre EQ "},
    {1, 0, VIRTUAL_BASS,      "2:Music DACX VB "},
    {1, 0, DRC,               "2:Music DACX DRC "},
    {1, 0, EQ,                "2:Music DACX EQ "},
};
```

图 11 SDK 音效列表

修改之后如图 12 所示：

```
6  const EffectComCt AudioEffectCommParam[]=
7  {
8      {2, 0, EXPANDER,          "1:Music Noise Suppressor "},
9      {2, 1, SILENCE_DETECTOR,"1:Music Silence Detector "},
10     {2, 2, GAIN_CONTROL,      "1:Music Pre Gain "},
11     {2, 0, EQ,                "1:Music Pre EQ "},
12     {2, 0, SILENCE_DETECTOR,"1:Music Silence Detector "},
13     {2, 0, VIRTUAL_BASS,      "1:Music VB "},
14     {2, 0, THREE_D,           "1:Music 3D "},
15     {2, 0, DRC,               "1:Music DRC "},
16     {2, 0, EQ,                "1:Music EQ "},
17     {1, 3, GAIN_CONTROL,      "2:Music DACX Pre Gain "},
18     {1, 0, EQ,                "2:Music DACX Pre EQ "},
19     {1, 0, VIRTUAL_BASS,      "2:Music DACX VB "},
20     {1, 0, DRC,               "2:Music DACX DRC "},
21     {1, 0, EQ,                "2:Music DACX EQ "},
22 };
```

图 12 SDK 音效列表

之后必须修改 CommParam.c 文件。可以手动修改，也可以通过脚本自动化生成。

手动修改音效参数 C 文件：

原 C 文件内容部分截图如图 13 所示：

```
158 0x00, 0x00,/*filter10_enable*/
159 0x00, 0x00,/*filter10_type*/
160 0x00, 0x00,/*filter10_f0*/
161 0x00, 0x00,/*filter10_Q*/
162 0x00, 0x00,/*filter10_gain*/
163
164 0x85,/*1:Music VB */
165 0x01,/*enable*/
166 0x64, 0x00,/*cutoff_frequency*/
167 0x23, 0x00,/*intensity*/
168 0x01, 0x00,/*enhanced*/
169
170 0x86,/*1:Music 3D */
171 0x01,/*enable*/
172 0x32, 0x00,/*intensity*/
173
174 0x87,/*1:Music DRC */
175 0x01,/*enable*/
176 0x2C, 0x01,/*crossover frequency*/
177 0x00, 0x00,/*mode*/
```

图 13 音效参数文件部分内容

修改之后文件内容如图 14 所示，如截图部分，在 0x85 音效修改为 silence detector 音效外，原 0x85 音效以之后的音效 Index 全部+1。并修改这个 buf 的总长度。

```
159 0x00, 0x00,/*filter10_type*/
160 0x00, 0x00,/*filter10_f0*/
161 0x00, 0x00,/*filter10_Q*/
162 0x00, 0x00,/*filter10_gain*/
163
164 0x85,/*1:Music Silence Detector */
165 0x01,/*enable*/
166 0xB7, 0x00,/*amplitude*/
167
168 0x86,/*1:Music VB */
169 0x01,/*enable*/
170 0x64, 0x00,/*cutoff_frequency*/
171 0x23, 0x00,/*intensity*/
172 0x01, 0x00,/*enhanced*/
173
174 0x87,/*1:Music 3D */
175 0x01,/*enable*/
176 0x32, 0x00,/*intensity*/
177
178 0x88,/*1:Music DRC */
179 0x01,/*enable*/
180 0x2C, 0x01,/*crossover frequency*/
181 0x00, 0x00,/*mode*/
182 0xD4, 0x02,/*q1*/
183 0xD4, 0x02,/*q2*/
184 0x00, 0x00, /*threshold1*/
```

图 14 音效参数文件部分内容

脚本自动修改：

将 comm_param.c 文件 copy 到“tools\audioeffect_list_to_paramters_scrip\source_flie”目录下，点击 audioeffect_list_to_parameters_script.bat 批处理，自动生成的文件会保存在“target_file”目录下，将 CommParam.c 文件 copy 到对应的 SDK 目录下即可。

注意：脚本自动生成的 CommParam.c 参数配置的都是默认值。

3.2.2 全新的音效

以添加 VBClass 音效为例（8M SDK 目前已经包含该音效）。

1、audio_effect_api.c 文件中实现 2 个 API，AudioEffectVBClassInit (VBClassUnit *unit, uint8_t channel, uint32_t sample_rate)；AudioEffectVBClassApply (VBClassUnit *unit, int16_t *pcm_in, int16_t *pcm_out, uint32_t n)；注意 apply 函数参数形参必须是 4 个，并且顺序不能调换。

2、audio_effect_api.h 添加 c 文件中的 2 个 API 函数的函数声明。实现 VBClassUnit 结构体的数据结构。enum EffectType 枚举值中添加一个字段，注意该音效的 Index 值，该值《固件与用户应用程序通信协议 Vx.x.x.pdf》中查找，如果是自定义音效，则该值需要大于 100，并且不能改变已有的音效 Index 值。

3、audio_effect.c 文件中实现具体函数调用。

AudioEffectsInit() 函数中添加 VBClass 初始函数 AudioEffectVBClassInit((VBClassUnit *)pNode->EffectUnit,ChannelNum,SampleRate); AudioEffectParsePackage()函数中申请音效结构体内存，并且为结构体成员函数执行赋值 apply 函数 pNode->FuncAudioEffect = (AudioEffectApplyFunc)AudioEffectVBClassApply;

4、communication.c 中实现调音函数。void Communication_Effect_VBClass(uint8_t Control, EffectNode* addr, uint8_t *buf, uint32_t len)

注意，

1、客户添加的自定义音效请参考 AEC 这个音效的定义，定义这个通讯数据必须遵守山景的规范，请参考《固件与用户应用程序通信协议 Vx.x.x.pdf》自定义音效部分。音效的 Index 不能重复，并且大于 100。

2、修改 comm_param.c 和音效导出 map 文件。请参考 3.2.1。

3.3 如何去掉一个音效

去掉一个音效可以参考 3.2.1，做减法操作。

3.4 用户按键控制音效操作

BP10 8M SDK 支持用户通过按键操作来更新关联的音效参数。例如通过 Key 控制音乐 EQ 模式的切换。为了方便用户移植修改这部分代码，V1.7 版本 SDK 增加了 4 个文件，将用户可能需要修改的代码集中。

表 3

audio_effect_user.c	用户通过 Key 控制音效的 API 实现代码
audio_effect_user.h	
eq_params.c	5 种用户 Key 控制音乐 EQ 音效参数文件
eq_params.h	

以 EQ 模式切换为举例。用户需要修改步骤大致如下：

1、 打开宏 CFG_FUNC_MUSIC_EQ_MODE_EN

2、 comm_param.c 音效列表文件中将需要控制的 EQ Index 参数设置为对应的值。

3、 确认 audio_effect.c 文件中 AudioEffectParsePackage() API 中关于 EQ 配置是否已经做了处理，将 EQ 对应的音效指针赋值给 music_mode_eq_unit。

SDK 中用户只要打开 CFG_FUNC_MUSIC_EQ_MODE_EN 宏，配置对应的按键消息关联即可。

3.5 调音辅助脚本工具说明

BP10 8M SDK 包 tools 目录下提供了多个脚本工具，用于辅助开发，具体功能和说明见下表。

表 4 脚本工具说明

脚本名称	功能说明
audioeffect_parameters_script	调音工具导出的音效参数 C 文件转换为 SDK 中简化格式的音效 C 文件。调用批处理会自动 copy 目标 C 文件到 SDK 包中。
audioeffect_list_to_parameters_script	根据音效参数表, 即 comm_param.c 文件生成 SDK 音效 C 文件，音效参数均为默认值。需要用户手动 copy 对应 C 文件到 SDK 中。如果需要历史音效参数也需要用户手动对比 copy。
audioeffect_effect_c_to_bin_script	调音工具导出的多个音效参数 C 文件，转化成音效 bin 文件，可以手动烧录到 flash 固定位置，也可以和 SDK bin 文件一起打包之后烧录到 flash 中。
audioeffect_effect_ini_to_bin_script	调音工具导出的多个音效参数 C 文件，转化成音效 bin 文件，可以手动烧录到 flash 固定位置，也可以和 SDK bin 文件一起打包之后烧录到 flash 中。

4 注意事项和常见问题

1、 用户添加音效之后，重新编译会出现异常，比如系统复位。可能的问题点：

a) 一组的音效长度超过了定义的最大长度，修改宏 AUDIO_EFFECT_NODE_NUM
b) 添加的音效为新的一个 groud，修改宏 AUDIO_EFFECT_GROUP_NUM
c) 音效参数列表和实际的音效参数不匹配，检查 comm_param.c 中的列表音效和 music_parameter 目录下的音效 table 是否一致，包括顺序
d) 添加的音效对应的宏是否已经正确打开，检查 audio_effect_api.h 文件音效宏是否已经使能为 1