

# HDMI 补丁升级和扩充说明

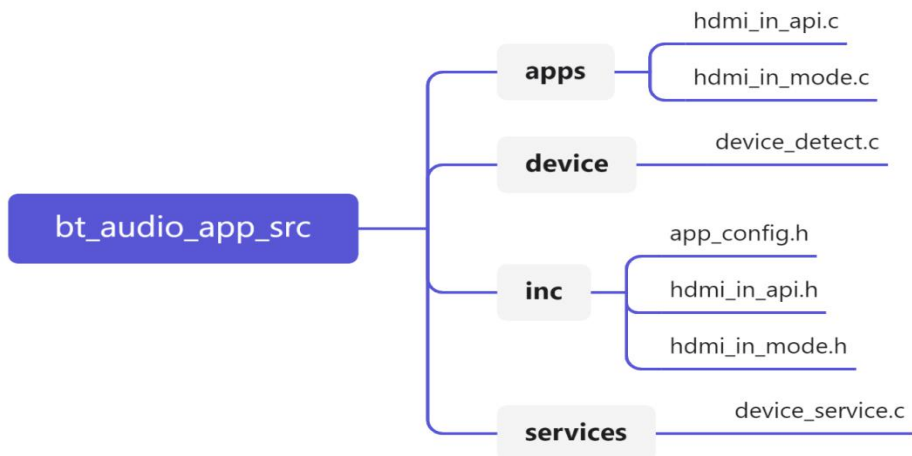
## 1. 简介

为了更好的兼容不同的 HDMI ARC 产品，HDMI 模式相关软件做了部分调整。该补丁说明书主要分两部分来说明：一部分是罗列出所涉及到的文件，一部分是遇到新的 HDMI ARC 产品如何补充进来。

## 2. 补丁文件

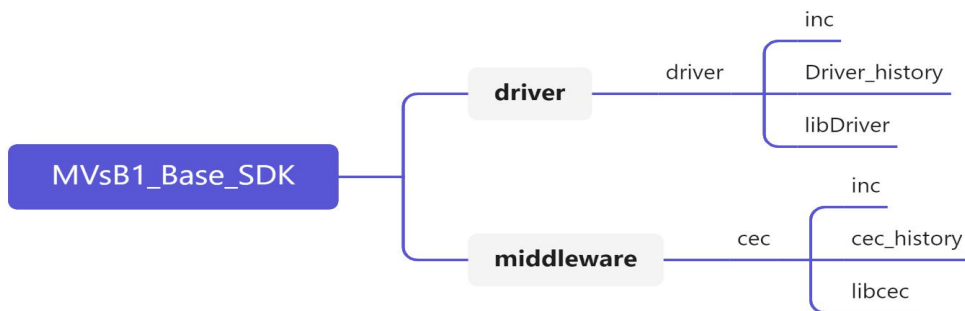
### 2.1 BT\_Audio\_APP 文件夹

该补丁涉及到文件如下图所示，软件的主要修改点集中在 hdmi\_in\_api 和 hdmi\_in\_mode 相关的源文件和头文件，其他文件只是略作修改。



### 2.2 MVsB1\_Base\_SDK 文件夹

此次更改涉及到了 Driver 里的 spdif 驱动和 CEC 库的 cec 驱动，具体文件目录如下：



### 3. HDMI 软件说明

#### 3.1 信息结构体

本次软件调整是为了更好地兼容不同类型 HDMI ARC 产品，同时方便用户在遇到新的不兼容的 ARC 产品时，能够去快速更改软件达到兼容新的 ARC 产品的目的。

```
const HDMI_TV_Info HDMIInfo[HDMI_COMPATIBLE_TV_NUM] =
{
    //TV Type      TV Product      thinking time      scan function      arbitration time
    {TV_SAMSUNG_1670, {'s', 'a', 'm', 0x16, 0x70}, 1, HDMI_CEC_Default_Scan, 20, 16 },
    {TV_SONY_03EF, {'s', 'n', 'y', 0x03, 0xef}, 1, HDMI_CEC_Default_Scan, 20, 16 },
    {TV_SONY_0537, {'s', 'n', 'y', 0x05, 0x37}, 1, HDMI_CEC_Default_Scan, 20, 16 },
    {TV_POLARIOD_010B, {'m', 't', 'c', 0x01, 0x0b}, 1, HDMI_CEC_Default_Scan, 20, 16 },
    {TV_LETV_0300, {'l', 't', 'v', 0x30, 0x00}, 2000, HDMI_CEC_UserDefined_Scan, 20, 16 },
    {TV_SONY_047C, {'s', 'n', 'y', 0x04, 0x7c}, 1, HDMI_CEC_Default_Scan, 20, 16 },
    {TV_HISENSE_3000, {'h', 'e', 'c', 0x30, 0x00}, 1, HDMI_CEC_Default_Scan, 20, 16 }
};
```

该信息结构体表存放在 hdmi\_in\_api.c 文件里，该表包含了当前时刻所兼容的 ARC 产品的相关信息，每款类型里包含 tv\_type, tv\_vendor, tv\_reaction\_time, HDMI\_CEC\_SCAN\_Func, SendArbitrTime, GetArbitrTime 信息。软件启动后会自动去识别是否为信息结构体里已兼容的产品，如果是，则自动获取信息结构体里的参数和函数并去相应配置和调用；如果没有存在在结构体中，则按照默认配置去执行。

- ◆ tv\_type: 产品类型。例如 TV\_SAMSUNG\_1670 为三星电视其中的一款产品；
- ◆ tv\_vendor: 供应商 ID，通过 HDMI\_DDC\_TVManufacturerGet 函数获得；
- ◆ tv\_reaction\_time: HDMI ARC 产品反应时间；
- ◆ HDMI\_CEC\_SCAN\_Func: CEC 通信函数；如果当前已存在的通信函数无法满足新兼容的产品，可自行定义通信函数，只要将函数信息存放到信息结构体里对应的产品一栏即可，软件会自动调用。
- ◆ SendArbitrTime 和 GetArbitrTime: 竞争机制时间；

#### 3.2 增加新的 ARC 产品

当用户遇到新的 ARC 产品，但同时当前发布的软件并没有很好的兼容时，用户可以扩展信息结构体来快速实现新产品的兼容。在扩展信息结构体时可参考已有类型的格式继续追加，软件如果匹配到对应的类型，会自动按照该类型里所包含的参数和函数去解析调用。

```
#define HDMI_COMPATIBLE_TV_NUM 8

typedef enum __TV_TYPE
{
    TV_DEFAULT = 0, //default
    TV_SAMSUNG_1670 = 1, //Samsung
    TV_HISENSE_3000 = 2, //Hisense
    TV_SONY_03EF = 3, //Sony
    TV_SONY_0537 = 4, //Sony-kd-55x8588g
    TV_POLARIOD_010B = 5, //Polariod
    TV_LETV_0300 = 6, //Letv
    TV_SONY_047C = 7, //Sony
} TV_TYPE;
```

首先在 TV\_TYPE 里定义新兼容的产品类型，注意对应的上面的宏的值也要增加。然后将新

类型的信息参考已有类型的信息罗列到信息结构体里。这样软件匹配到该类型的产品后，会找到信息结构体里的参数和函数去执行。

第二要获取新兼容产品的 tv\_vendor。 可通过 HDMI\_DDC\_TVManufacturerGet 函数获取当前产品的厂商信息，并将该参数写入信息结构体对应的 tv\_vendor 参数里。

```
uint8_t HDMI_DDC_GetInfo(void)
{
    const uint8_t IEEERegist[3] = {0x03, 0x0c, 0x00};
    const uint8_t MVDDCHeader[8] = {0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00};
    char Tv_Manufacture[5];
    uint8_t i;
    uint8_t flag;
    uint16_t check_sum;
    #ifdef DDC_USE_SW_I2C
    #else
    I2C_ErrorState ret;
    #endif

    #ifdef DDC_USE_SW_I2C
    uint8_t MVDDCReadAddr_sf = 0xA0;
    #else
    uint8_t MVDDCReadAddr = 0x50;
    #endif

    WDG_Feed();

    gHdmiCt->hdmi_tv_inf.tv_type = 0;
    gHdmiCt->hdmi_tv_inf.HDMI_CEC_SCAN_Func = HDMI_CEC_Default_Scan;
    gHdmiCt->hdmi_tv_inf.tv_reaction_time = 0;
    gHdmiCt->hdmi_tv_inf.GetArbitrTime = 16;
    gHdmiCt->hdmi_tv_inf.SendArbitrTime = 20;

    memset(&gHdmiCt->edid_buf[0], 0, 256);
    #ifdef DDC_USE_SW_I2C
    I2cReadNByte(DDCI2cHandler, MVDDCReadAddr_sf, 0, &gHdmiCt->edid_buf[0], 128);
    HDMI_DDC_TVManufacturerGet(gHdmiCt->edid_buf, (uint8_t*)Tv_Manufacture);
    //读取电视的厂商名字
    DBG("TV Manufacturer: %c%c%c\n", Tv_Manufacture[0], Tv_Manufacture[1], Tv_Manufacture[2]);
    //读取电视的产品ID
```

第三确定当前产品的 think time, arbitration time 是否与默认值不同;

第四如果当前产品的 cec 交互与默认差距比较大，可自行定义 scan 函数，并把函数地址写入信息结构体对应的位置即可。

如上操作完成后，软件会自动调用对应参数。