



MobileCam SDK 开发指南

文档版本 01

发布日期 2016-11-28

版权所有 © 深圳市海思半导体有限公司 2016。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址：深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址：<http://www.hisilicon.com>

客户服务电话：+86-755-28788858

客户服务传真：+86-755-28357515

客户服务邮箱：support@hisilicon.com



前言

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3518E	V200
Hi3516C	V200



说明

未有特殊说明，Hi3516CV200 和 Hi3518EV200 一致。

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

文档版本 01 (2016-11-28)

第 1 次正式版本发布。



目 录

前 言.....	i
目 录.....	ii
插图目录.....	iv
1 初始化及驱动.....	1
1.1 Sample 用例目录结构.....	1
1.2 app 初始化及用例入口	2
1.3 Sensor 驱动的定义和配置	3
2 Sample 用例.....	4
2.1 Audio 音频调试	4
2.1.1 概述	4
2.1.2 实时对讲和编码流程.....	4
2.1.3 结构体说明.....	7
2.1.4 Proc 调试信息	10
2.2 Venc 视频编码及 LCD 调试	17
2.2.1 概述	17
2.2.2 视频调试流程图.....	18
2.2.3 LCD 调试	19
2.2.4 结构体说明.....	20
2.2.5 Proc 调试信息	26
2.3 Wifi 调试.....	37
2.3.1 概述	37
2.3.2 Wifi 待机	38
2.3.3 Wifi 唤醒	41
2.3.4 AP 模式和 STA 模式待机唤醒区别.....	43
2.3.5 待机唤醒时序步骤.....	43
2.3.6 Wifi 镜像生成	45
2.3.7 Wifi 待机唤醒场景实现.....	46
2.3.8 接口说明	48
3 MCU 部分.....	55



3.1 MCU 烧录.....	55
3.1.1 MCU 文件烧录	55
3.1.2 MCU 源码仿真烧录.....	57
4 RTSP 说明	58
4.1 概述.....	58
4.2 RTSP 消息格式	58
4.3 RTSP 交互过程	58
4.4 文件结构.....	59
4.5 编译方式.....	59
4.6 主要 API 接口	60
4.7 APK.....	61
4.8 RTSP response statue-code.....	62
5 UBOOT 烧写方式	64
5.1 概述.....	64
5.2 烧写步骤.....	64
5.3 注意事项.....	66



插图目录

图 1-1 sample 目录结构	2
图 1-2 初始化完成打印信息.....	3
图 2-1 mic 和 speaker 在单板中的位置.....	5
图 2-2 mic 和 speaker 在 sensor 板的位置	6
图 2-3 音频对讲和编码流程图.....	7
图 2-4 AI 调试截图	13
图 2-5 Proc 调试截图	13
图 2-6 AO 调试截图.....	15
图 2-7 VENC 用例场景.....	17
图 2-8 视频调试流程图.....	19
图 2-9 读写寄存器命令.....	20
图 2-10 VI 调试截图	29
图 2-11 VENC 调试截图.....	32
图 2-12 sys 调试截图	34
图 2-13 sys 调试截图	36
图 2-14 镜像编译模式选择.....	38
图 2-15 Sample 电路框图.....	39
图 2-16 待机流程图	40
图 2-17 唤醒流程图	42
图 2-18 AP 模式和 STA 模式网络拓扑图.....	43
图 2-19 STA 模式唤醒时序图	44
图 2-20 生成 WiFi 镜像后的 Flash 内容分布	46
图 2-21 Wifi 待机唤醒流程	47
图 3-1 MCU 类型选择	56
图 3-2 点击烧录	56



图 3-3 验证成功	57
图 3-4 源码仿真烧录	57
图 3-5 调试模式	57



1 初始化及驱动

1.1 Sample 用例目录结构

Mapi 的调试用例目录为\mapi\sample\hi3518e，如[图 1-1](#)所示。

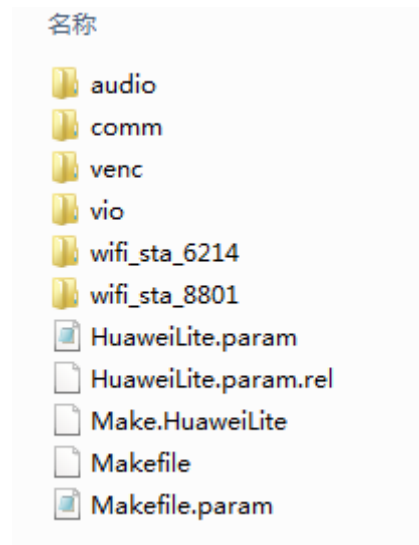
结构如下：

```
-|audio  
-|comm  
-|venc  
-|vio  
-|wifi_sta_6214  
-|wifi_sta_8801
```

- audio 目录用于音频调试有关用例，比如对讲，音频编码。
- comm 目录用于存放各公共部分文件，以及用例的部分文件。
- venc 用于视频编码调试相关用例，sensor 调用和 LCD 显示预览也在 venc 中调试。
- wifi_sta_6214 和 wifi_sta_8801 用于调试 wifi 相关业务，实现 wifi 休眠，唤醒。
- vio 调试视频输入到输出，venc 也可实现此功能，可不另调试。



图1-1 sample 目录结构



1.2 app 初始化及用例入口

app 初始化函数目录: \mapi\sample\hi3518e\comm\ app_init.c

- 板端初始化入口函数: app_init
- sample 用例入口函数: app_main

板端初始化完成打印信息如图 1-2 所示, 至此, 输入命令 “sample” 回车, 即可以进入所烧 sample 用例入口函数 app_main。



图1-2 初始化完成打印信息

```
Do not support tasklet.
ISP Mod init!
load vpss.ko for Hi3518EV200...OK!
load vou.ko for Hi3518EV200...OK!
load rc.ko for Hi3518EV200...OK!
load venc.ko for Hi3518EV200...OK!
load chnl.ko for Hi3518EV200...OK!
load h264e.ko for Hi3518EV200...OK!
load jpege.ko for Hi3518EV200...OK!
mipi_init
init_phy power successful!
load hi_mipi driver successful!
acodec [acodec_device_init ,1091]: misc addr:0x20120000, acodec addr:0x201200c4

acodec [acodec_device_init ,1107]: crg addr:0x20030000, acodec crg addr:0x2003008c

SDK init ok...
Link is Up - 100Mbps/Full

Huawei LiteOS #
Huawei LiteOS # sample
sample task id 3

Huawei LiteOS # Usage :
    0) ai->ao(G726:SampleRate:8k).
    1) ai->aenc(AAC:SampleRate:8k).
```

1.3 Sensor 驱动的定义和配置

- Sensor 的适配具体见《Sensor 调试指南》，如适配正确，在 venc 用例中即可正常调用。
- MobileCam SDK 默认使用 OV9732 sensor，有两个地方需要修改为相应的值：
 - 项目编译文件 HuaweiLite.param
SENSOR_TYPE ?= OMNIVISION_OV9732_MIPI_720P_30FPS
 - sdk_init.c 文件
sensor_type= "ov9732";



2 Sample 用例

2.1 Audio 音频调试

2.1.1 概述

AUDIO 模块包括音频采集(ACAP)、音频编码 AENC、音频解码 ADEC、音频输出 AO 四个子模块，本用例调试可实现音频对讲 AI->AO、音频编码 AI->AENC。

- 用例目录：\mapi\sample\hi3518e\audio\HuaweiLite
- 编译目录：\mapi\sample\hi3518e 或 \mapi\sample\hi3518e\audio\HuaweiLite
- 编译命令：make clean;make
- 生成文件：sample_audio.bin
- 烧写脚本：
 - 烧写到内存（掉电不保存）：

```
tftp 0x80008000 sample_audio.bin;
go 0x80008000
```
 - 烧写到 flash（掉电保存）：

```
tftp 0x80008000 sample_audio.bin;
sf probe 0;
sf erase 0x40000 0x400000;
sf write 0x80008000 0x40000 0x400000;
setenv bootcmd 'sf probe 0;sf read 0x80008000 0x40000 0x400000;go 0x80008000';
saveenv;
```

烧写到 flash 完成后，需要复位开机，或者执行软件复位“reset”。
- 本用例可以实现两个场景：
 - 实时对讲，不编码，没有音频文件生成。
 - 音频编码，SD 卡生成 aac 格式的音频文件，可用电脑播放器播放。

2.1.2 实时对讲和编码流程

实时对讲场景说明，两对麦克风和喇叭分布如下：

Mic0

Mic1

Speaker0

Speaker1

- Speaker1 实时输出 Mic0 音频
- Speaker0 实时输出 Mic1 音频

mic 和 speaker 在单板的位置参见图 2-1 和图 2-2。

图2-1 mic 和 speaker 在单板中的位置

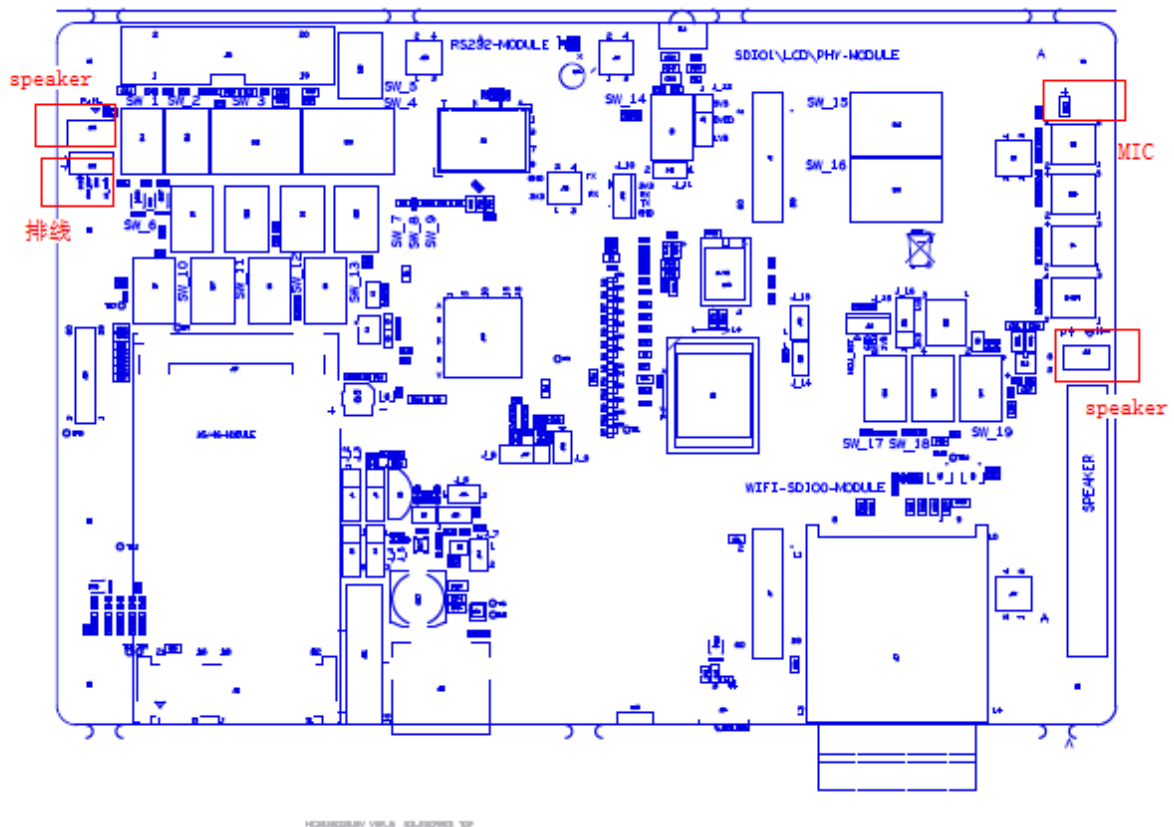
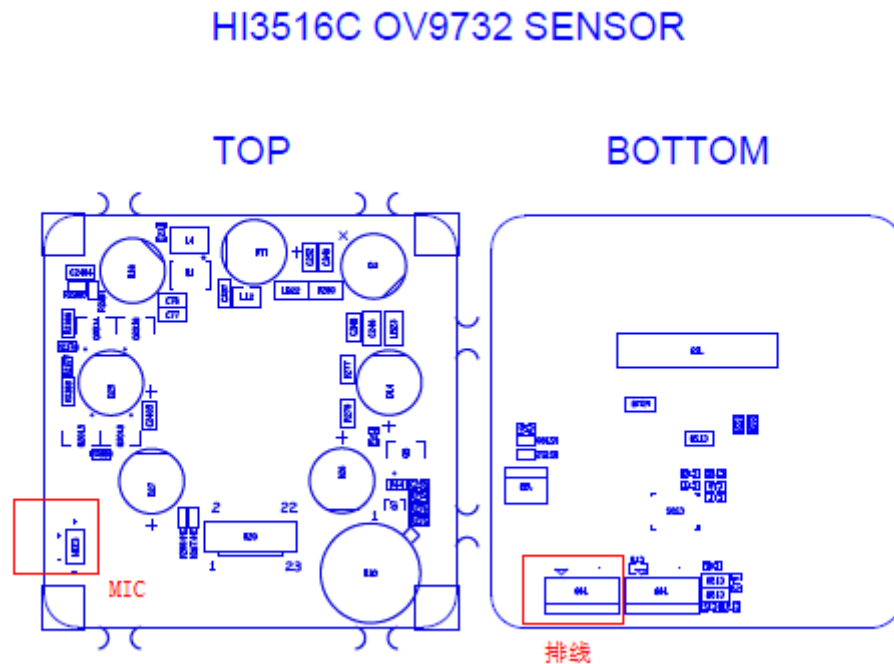


图2-2 mic 和 speaker 在 sensor 板的位置



音频编码主要实现创建编码通道、绑定音频输入通道及获取编码码流等功能，当前仅支持 AAC 格式编码。

实时对讲和音频编码在同一用例调试，可根据需要选择实时对讲或音频编码：

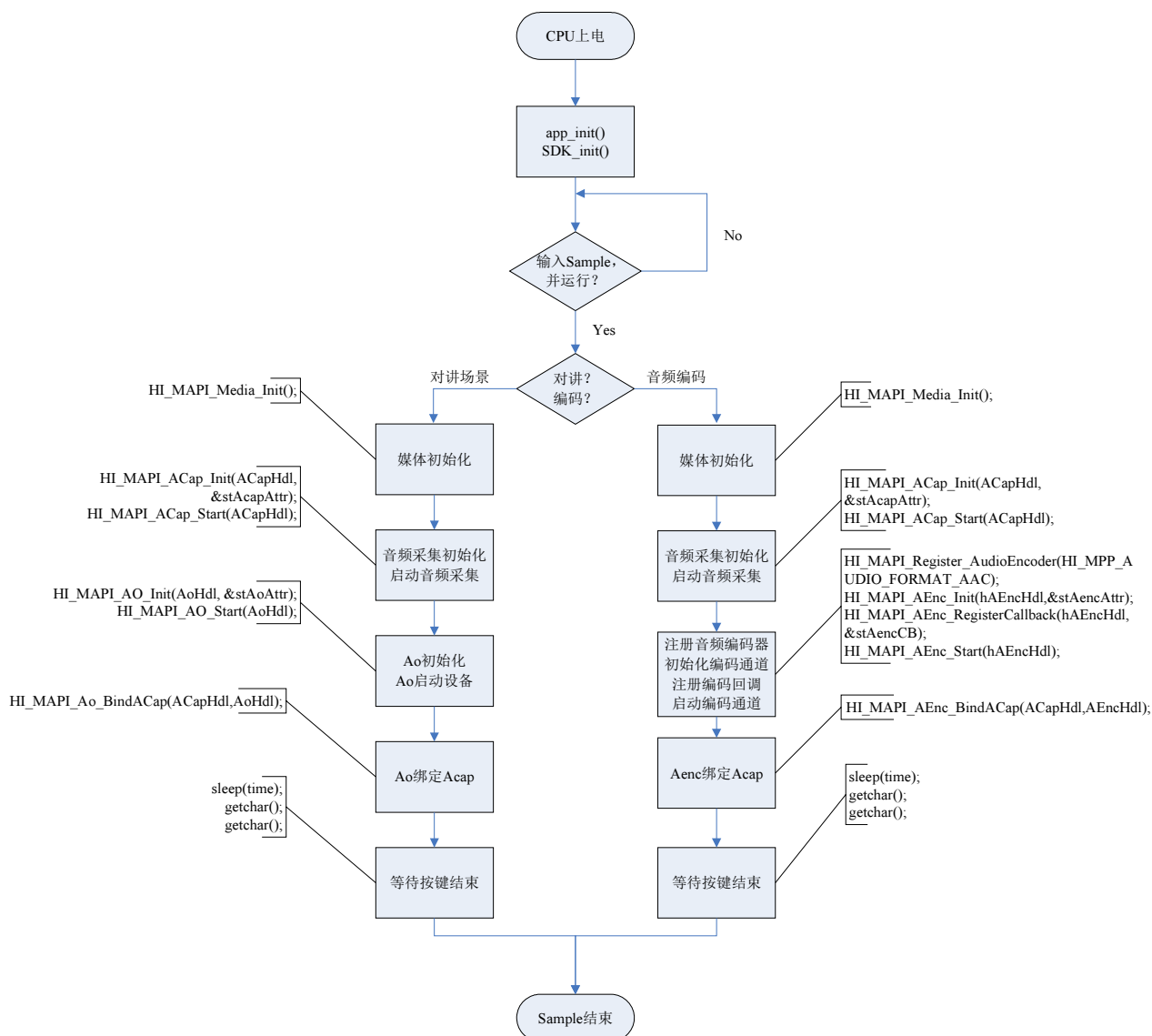
```
ai->ao(G726:SampleRate:8k).
```

```
ai->aenc(AAC:SampleRate:8k).
```

实时对讲和音频编码流程图如图 2-3 所示。



图2-3 音频对讲和编码流程图



2.1.3 结构体说明

HI_MPP_ACAP_ATTR_S: 音频采集结构体。

AENC_ATTR_AAC_S: 音频编码结构体。

HI_MPP_ACAP_ATTR_S

【定义】

```
typedef struct hiMPP_ACAP_ATTR_S
{
    HI_MPP_AUDIO_SAMPLE_RATE_E  enSampleRate;
    HI_MPP_AUDIO_BITWIDTH_E     enBitwidth;
```



```
HI_MPP_AUDIO_SOUND_MODE_E  enSoundMode;
HI_MPP_AUDIO_MODE_E  enAudioMode;
HI_U32                    u32PtNumPerFrm;
} HI_MPP_ACAP_ATTR_S;
```

结构体 HI_MPP_ACAP_ATTR_S，用于设置 ACAP 属性和 AO 属性，实时对讲场景的音频输出 AO 属性必须与音频采集 ACAP 属性一致，AO 可直接调用其设置。

【应用举例】

```
HI_MPP_ACAP_ATTR_S stAcapAttr;
HI_S32 ACapHdl = 0;
HI_S32 AoHdl = 0;

stAcapAttr.enAudioMode = HI_MPP_AUDIO_MODE_I2S_MASTER;
stAcapAttr.enBitwidth = HI_MPP_AUDIO_BITWIDTH_16;
stAcapAttr.enSampleRate = HI_MPP_AUDIO_SAMPLE_RATE_8;
stAcapAttr.enSoundMode = HI_MPP_AUDIO_SOUND_MODE_STEREO;
stAcapAttr.u32PtNumPerFrm = 320;

s32Ret = HI_MAPI_ACap_Init(ACapHdl, &stAcapAttr);
if(HI_SUCCESS != s32Ret)
{
    printf("HI_MAPI_ACap_Init fail s32Ret:%d\n", s32Ret);
    return HI_FAILURE;
}

s32Ret = HI_MAPI_ACap_Start(ACapHdl);
if(HI_SUCCESS != s32Ret)
{
    printf("HI_MAPI_ACap_Start fail s32Ret:%#x\n", s32Ret);
    return HI_FAILURE;
}

s32Ret = HI_MAPI_AO_Init(AoHdl, &stAcapAttr);
if(HI_SUCCESS != s32Ret)
{
    printf("HI_MAPI_AO_Start return ERR, %x\n", s32Ret);
    return HI_FAILURE;
}

s32Ret = HI_MAPI_AO_Start(AoHdl);
if(HI_SUCCESS != s32Ret)
{
    printf("HI_MAPI_AO_Start return ERR, %x\n", s32Ret);
    return HI_FAILURE;
}
```



```
s32Ret = HI_MAPI_Ao_BindACap(ACapHdl, AoHdl);
if(HI_SUCCESS != s32Ret)
{
    printf("HI_MAPI_AEnc_BindACap fail ");
    return HI_FAILURE;
}
```

AENC_ATTR_AAC_S

【定义】

```
typedef struct hiAENC_ATTR_AAC_S
{
    AAC_TYPE_E          enAACType;
    AAC_BPS_E           enBitRate;
    HI_MPP_AUDIO_SAMPLE_RATE_E enSmpRate;
    HI_MPP_AUDIO_BITWIDTH_E enBitWidth;
    HI_MPP_AUDIO_SOUND_TYPE_E enSoundType;
    HI_S16              s16BandWidth;
    AAC_TRANS_TYPE_E    enTransType;
} AENC_ATTR_AAC_S;
```

【应用举例】

```
HI_MPP_AENC_ATTR_S stAencAttr;
AENC_ATTR_AAC_S stAacAencAttr;
HI_S32 AEncHdl = 0;

HI_MPP_AENC_ATTR_S stAENCAAttr;
AENC_ATTR_AAC_S stAacAENCAAttr;
stAacAENCAAttr.enAACType = AAC_TYPE_AACLC;
stAacAENCAAttr.enBitRate = AAC_BPS_48K;
stAacAENCAAttr.enBitWidth = HI_MPP_AUDIO_BITWIDTH_16;
stAacAENCAAttr.enSmpRate = HI_MPP_AUDIO_SAMPLE_RATE_8;
stAacAENCAAttr.enSoundType = HI_MPP_AUDIO_SOUND_TYPE_STEREO;
stAacAENCAAttr.enTransType = AAC_TRANS_TYPE_ADTS;
stAacAENCAAttr.s16BandWidth = 0;
stAENCAAttr.enAENCFormat = HI_MPP_AUDIO_FORMAT_AAC;
stAENCAAttr.u32PtNumPerFrm = 1024;
stAENCAAttr.pValue = &stAacAENCAAttr;
stAENCAAttr.u32Len = sizeof(stAacAENCAAttr);

s32Ret = HI_MAPI_Register_AudioEncoder(HI_MPP_AUDIO_FORMAT_AAC);
if(HI_SUCCESS != s32Ret)
{
    printf("HI_MAPI_Register_AudioEncoder fail s32Ret:%d\n", s32Ret);
}
```




```
        return HI_FAILURE;
    }
    s32Ret = HI_MAPI_AEnc_Init(hAEncHdl, &stAencAttr);
    if(HI_SUCCESS != s32Ret)
    {
        printf("HI_MAPI_AEnc_Init fail s32Ret:%d\n", s32Ret);
        return HI_FAILURE;
    }
    HI_AENC_CALLBACK_S stAencCB;
    stAencCB.pfnDataCB = SAMPLE_AUDIO_DataProc;
    stAencCB.pPrivateData = HI_NULL;
    s32Ret = HI_MAPI_AEnc_RegisterCallback(hAEncHdl, &stAencCB);
    if(HI_SUCCESS != s32Ret)
    {
        printf("HI_MAPI_AEnc_RegisterCallback fail s32Ret:%d\n", s32Ret);
        return HI_FAILURE;
    }
    s32Ret = HI_MAPI_AEnc_Start(hAEncHdl);
    if(HI_SUCCESS != s32Ret)
    {
        printf("HI_MAPI_AEnc_Start fail s32Ret:%d\n", s32Ret);
        return HI_FAILURE;
    }

    s32Ret = HI_MAPI_AEnc_BindACap(ACapHdl, AEncHdl);
    if(HI_SUCCESS != s32Ret)
    {
        printf("HI_MAPI_AEnc_BindACap fail ");
        return HI_FAILURE;
    }
}
```

2.1.4 Proc 调试信息

2.1.4.1 AI

打印文本信息

ai 信息查看命令:cat /proc/umap/ai

Huawei LiteOS # cat /proc/umap/ai

Huawei LiteOS #

[AI] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov 2 2016, 09:47:44]

-----AI DEV ATTR-----



```

AiDev WorkMod SampR BitWid ChnCnt ClkSel SoundMod PoiNum ExFlag
FrmNum
0 i2s_mas 8kHz 16bit 2 0 stereo 320 0 30

-----AI DEV STATUS0-----
AiDev IntCnt fifoCnt buffInt FrmTime MaxFrmTime TranLen IsrTime
0 575 0 0 40000 40005 1280 374

-----AI DEV STATUS1-----
AiDev MaxIsrTime CBPhy CBSize ROffSet WOffSet
0 455 860f1000 2560 500 0

-----AI DEV EXTEND STATUS-----
AiDev enTrack bMute Volume
0 0 N 0

-----AI CHN STATUS-----
AiDev AiChn State Read Write BufFul UsrQueLost UsrFrmDepth
u32Data0 u32Data1 UserGet UserRls
0 0 enable 0 0 0 0 0 ffd3ffd2
ffd4ffd3 0 0
0 1 enable 0 0 0 0 0 ffffffff5
fffbffffb 0 0

-----AI CHN RESAMPLE STATUS-----
AiDev AiChn State bResmp PoiNum InSampR OutSampR
0 0 enable N 0 (null) (null)
0 1 enable N 0 (null) (null)

-----AI CHN VQE STATUS0-----
AiDev AiChn State bVqe workmod RATE PoiNum GainVol bAnr bAgc bEq
bHpf bAec bRnr bHdr bDrc bPeq WrFile
0 0 enable Y comm 8kHz 320 0 Y N N Y Y
N N N N N
0 1 enable N comm (null) 0 0 N N N N N
N N N N N

-----AI CHN VQE STATUS1-----
AiDev AiChn State bAnr bUsrmod NrIntensity NoiseDbThr SpProSwi
0 0 enable Y N 29604 -32677 -84
0 1 enable N N 0 0 0

-----AI CHN VQE STATUS2-----
AiDev AiChn State bHpf bUsrmod HpFReq

```



0	0	enable	Y	Y	80
0	1	enable	N	N	0

-----AI CHN VQE STATUS3-----

AiDev AiChn State bAec bUsrmod CngMode DTHnlStQTh NrAlPsEngy

NrClnSupEngy AecAo AecFail

0	0	enable	Y	N	close	-11692	115	91 (0, 0)
---	---	--------	---	---	-------	--------	-----	------------

0

0	1	enable	N	N	close	0	0	0 (-1,-1)
---	---	--------	---	---	-------	---	---	-----------

0

-----AI CHN VQE STATUS4-----

AiDev AiChn State bAec bUsrmod VcPrtctFrqL VcPrtctFrqL1 EcoBndLow

EcoBndHgh EcoBndLow2 EcoBndHgh2

0	0	enable	Y	N	29596	-32677	-32745	-11552
---	---	--------	---	---	-------	--------	--------	--------

-32710 29624

0	1	enable	N	N	0	0	0	0
---	---	--------	---	---	---	---	---	---

0 0

-----AI CHN VQE STATUS5-----

AiDev AiChn State bAec bUsrmod ERLBND[0] ERLBND[1] ERLBND[2]

ERLBND[3] ERLBND[4] ERLBND[5]

0	0	enable	Y	N	-32677	23041	16420	-14240
---	---	--------	---	---	--------	-------	-------	--------

-32694 8332

0	1	enable	N	N	0	0	0	0
---	---	--------	---	---	---	---	---	---

0 0

-----AI CHN VQE STATUS6-----

AiDev AiChn State bAec bUsrmod ERL[0] ERL[1] ERL[2] ERL[3] ERL[4]

ERL[5] ERL[6]

0	0	enable	Y	N	-32690	0	0	29560	-32677	18788
---	---	--------	---	---	--------	---	---	-------	--------	-------

-32733

0	1	enable	N	N	0	0	0	0	0	0
---	---	--------	---	---	---	---	---	---	---	---

0



图2-4 AI 调试截图

```
Huawei LiteOS #
Huawei LiteOS # cat /proc/umap/ai

Huawei LiteOS #
[AI] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov 2 2016, 09:47:44]

-----AI DEV ATTR-----
AiDev WorkMod SampR BitWid ChnCn ClkSel SoundMod PoiNum ExFlag FrmNum
0 i2s_mas 8kHz 16bit 2 0 stereo 320 0 30

-----AI DEV STATUS0-----
AiDev IntCnt fifoCnt buffInt FrmTime MaxFrmTime TranLen IsrTime
0 575 0 0 40000 40005 1280 374

-----AI DEV STATUS1-----
AiDev MaxIsrTime CBPhy CbSize ROffSet WOffSet
0 455 860f1000 2560 500 0

-----AI DEV EXTEND STATUS-----
AiDev enTrack bMute Volume
0 0 N 0

-----AI CHN STATUS-----
AiDev AiChn State Read Write BufFul UsrQueLost UsrFrmDepth u32Data0 u32Data1 UserGet UserRls
0 0 enable 0 0 0 0 0 ffd3ffd2 ffd4ffd3 0 0
0 1 enable 0 0 0 0 0 ffffffff fffbffff 0 0

-----AI CHN RESAMPLE STATUS-----
AiDev AiChn State bResmp PoiNum InSampR OutSampR
0 0 enable N 0 (null) (null)
0 1 enable N 0 (null) (null)

-----AI CHN VQE STATUS0-----
AiDev AiChn State bVqe workmod RATE PoiNum GainVol bAnr bAgc bEq bHpf bAec bRnr bHdr bDrc bPeg WrFile
0 0 enable Y comm 8kHz 320 0 Y N N Y Y N N N N N
0 1 enable N comm (null) 0 0 N N N N N N N N N

-----AI CHN VQE STATUS1-----
AiDev AiChn State bAnr bUsrmod NrIntensity NoiseDbThr SpProSwi
0 0 enable Y N 29604 -32677 -84
0 1 enable N N 0 0 0

-----AI CHN VQE STATUS2-----
AiDev AiChn State bHpf bUsrmod HpFFreq
0 0 enable Y Y 80
0 1 enable N N 0
```

图2-5 Proc 调试截图

```
-----AI CHN VQE STATUS3-----
AiDev AiChn State bAec bUsrmod CngMode DTHnlStQTh NrAlPsEngy NrClnSupEngy AecAo AecFail
0 0 enable Y N close -11692 115 91 (0,0) 0
0 1 enable N N close 0 0 0 (-1,-1) 0

-----AI CHN VQE STATUS4-----
AiDev AiChn State bAec bUsrmod VcPrtctFrqL VcPrtctFrqL1 EcoBndLow EcoBndHgh EcoBndLow2 EcoBndHgh2
0 0 enable Y N 29596 -32677 -32745 -11552 -32710 29624
0 1 enable N N 0 0 0 0 0

-----AI CHN VQE STATUS5-----
AiDev AiChn State bAec bUsrmod ERLBND[0] ERLBND[1] ERLBND[2] ERLBND[3] ERLBND[4] ERLBND[5]
0 0 enable Y N -32677 23041 16420 -14240 -32694 8332
0 1 enable N N 0 0 0 0 0

-----AI CHN VQE STATUS6-----
AiDev AiChn State bAec bUsrmod ERL[0] ERL[1] ERL[2] ERL[3] ERL[4] ERL[5] ERL[6]
0 0 enable Y N -32690 0 0 29560 -32677 18788 -32733
0 1 enable N N 0 0 0 0 0
```



参数说明

具体每个参数含义见《HiMPP IPC V2.0 媒体处理软件开发参考.pdf》，sample 用例中，音频属性设置是否成功，可查看 proc 调试信息的 AI DEV ATTR（参考数据结构 AIO_ATTR_S）。音频采集数据，可查看 proc 调试信息的 AI CHN STATUS（音频输入通道信息）的 u32Data0 和 u32Data1，如果采集成功数值有变化，如图 2-5 中可以看出，两个声道都采集数据。

2.1.4.2 AO

打印文本信息

ao 信息查看命令: cat /proc/umap/ao

Huawei LiteOS # cat /proc/umap/ao

Huawei LiteOS #

```
[AO] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov 2
2016, 09:47:44]
```

```
-----AO DEV ATTR-----
```

AoDev	WorkMod	SampR	BitWid	ChnCnt	ClkSel	SoundMod	PoiNum	ExFlag
0	i2s_mas	8kHz	16bit	2	0	stereo	320	2

```
-----AO DEV STATUS0-----
```

AoDev	IntCnt	fifoCnt	buffInt	FrmTime	MaxFrmTime	TranLen	IsrTime
0	339	0	0	39999	40008	1280	180

```
-----AO DEV STATUS1-----
```

AoDev	MaxIsrTime	CBPhy	CBSize	ROffSet	WOffSet
0	186	8611e000	5120	0	500

```
-----AO DEV EXTEND STATUS-----
```

AoDev	enTrack	bMute	Volume
0	0	N	5

```
-----AO CHN STATUS-----
```

AoDev	AoChn	State	Read	Write	BufEmp	u32Data0	u32Data1	bResmp
0	0	enable	0	1	1	ffc8ffc4	ffc3ffc4	N
0	(null)	(null)						
0	1	enable	0	1	1	fd00ff	f900f8	N
0	(null)	(null)						

```
-----AO CHN VQE STATUS0-----
```

AoDev	AoChn	State	bVqe	workmod	RATE	PoiNum	bAnr	bAgc	bEq	bHpf
-------	-------	-------	------	---------	------	--------	------	------	-----	------



```
WrFile
0 0 enable N comm (null) 0 N N N N N
0 1 enable N comm (null) 0 N N N N N
```

图2-6 AO 调试截图

```
Huawei LiteOS # cat /proc/umap/ao

Huawei LiteOS #
[AO] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov 2 2016, 09:47:44]

-----AO DEV ATTR-----
AoDev WorkMod SampR BitWid ChnCnt ClkSel SoundMod PoiNum ExFlag FrmNum
0 i2s_mas 8kHz 16bit 2 0 stereo 320 0 2

-----AO DEV STATUS0-----
AoDev IntCnt fifoCnt buffInt FrmTime MaxFrmTime TranLen IsrTime
0 339 0 0 39999 40008 1280 180

-----AO DEV STATUS1-----
AoDev MaxIsrTime CBPhy CbSize ROffset WOffset
0 186 8611e000 5120 0 500

-----AO DEV EXTEND STATUS-----
AoDev enTrack bMute Volume
0 0 N 5

-----AO CHN STATUS-----
AoDev AoChn State Read Write BufEmp u32Data0 u32Data1 bResmp PoiNum InSampR OutSampR
0 0 enable 0 1 1 ffc8ffca ffc3ffc4 N 0 (null) (null)
0 1 enable 0 1 1 fd00ff f900f8 N 0 (null) (null)

-----AO CHN VQE STATUS0-----
AoDev AoChn State bVqe workmod RATE PoiNum bAnr bAgc bEq bHpf WrFile
0 0 enable N comm (null) 0 N N N N N
0 1 enable N comm (null) 0 N N N N N
```

参数说明

AO 属性必须和 AI 属性一致，参数含义相同。AO 必须绑定 ACAP 才能接收到 AI 的数据，绑定成功 AO CHN STATUS（音频输入通道信息）的 u32Data0 和 u32Data1 可以看到有变化。

2.1.4.3 AENC

打印文本信息

音频采集信息查看命令: cat /proc/umap/aenc

```
Huawei LiteOS # cat /proc/umap/aenc
```

```
Huawei LiteOS #
```

```
[AENC] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov 2 2016, 09:47:44]
```

```
-----AENC CHN ATTR-----
```

```
ChnId PlType ADPCMTType PoiNum BufSize G726Rate
0 aac NULL 1024 50 NULL
```



```
-----AENC CHN STATUS-----
ChnId      RcvFrm  AiQueLost      EncOk      FrmErr      BufFull      GetStrm
RlsStrm    WtFile
0          52      0              52         0           0           52         52
N
```

表2-1 AENC 调试截图

```
Huawei LiteOS # cat /proc/umap/aenc

Huawei LiteOS #
[AENC] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov  2 2016, 09:47:44]

-----AENC CHN ATTR-----
ChnId  PlType  ADPCMType  PoiNum  BufSize      G726Rate
0      aac      NULL      1024    50           NULL

-----AENC CHN STATUS-----
ChnId  RcvFrm  AiQueLost      EncOk      FrmErr      BufFull      GetStrm      RlsStrm  WtFile
0      52      0              52         0           0           52         52      N
```

参数说明

【调试信息分析】

记录当前音频编码属性配置以及状态信息。

参数		描述
Attribution of AENC Channel（音频编码通道属性）	ChnId	AENC 通道号。
	PlType	编码协议类型。
	ADPCMType	ADPCM 编码时具体的类型： <ul style="list-style-type: none">ADPCM_TYPE_DVI4;ADPCM_TYPE_IMA;ADPCM_TYPE_ORG_DVI4
	PoiNum	编码通道允许的最大帧长中的采样点数目。
	BufSize	帧缓存数目。
	G726Rate	G726 协议编码时的编码码率。
Status of AENC Channel（音频编码通道状态）	ChnId	AENC 通道号。
	RcvFrm	接收的音频帧数目。
	AiQueLost	Ai-AENC 系统绑定时的丢帧数。
	EncOk	成功编码的音频帧数目。



	EncErr	编码失败的音频帧数目。
	BufFull	音频码流缓冲区满的次数（用户未及时获取音频码流数据会导致缓冲区满，进而会导致码流数据丢失。
	GetStrm	用户获取音频码流的次数。
	RlsStrm	用户释放音频码流的次数。
	WtFile	该通道是否在向文件存储通道数据。 enable: 启用向文件存储通道数据功能； disable: 关闭向文件存储通道数据功能。

2.2 Venc 视频编码及 LCD 调试

2.2.1 概述

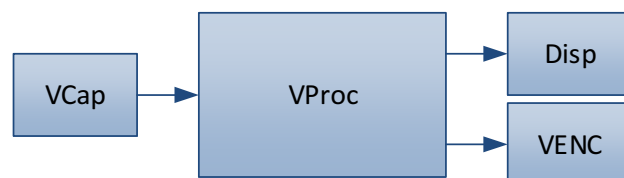
VENC 用例包括视频采集、视频处理、视频编码和显示。通过调用各模块的绑定接口，可与 VCAP 和 Disp/VENC 模块进行绑定，其中前者为 VProc 的输入源，后者为 VProc 处理后图像的接收者，如图 2-7 所示可实现视频编码，LCD 同步显示场景。

视频采集（VideoCapture，简称 VCAP）模块实现的功能：通过 MIPI Rx(含 MIPI 接口、LVDS 接口和 HISPI 接口)接收视频数据，将前端输入的原始数据送入到内部的图像信号处理单元(ISP)进行处理，并最终输出 YUV 数据给视频后处理模块（VideoProcess，简称 VPROC）。

VProc（Video Process）支持对输入的 VCap 图像进行处理，如去噪、拍照算法处理、缩放旋转等处理，同时一路图像输入可以输出多路处理后的不同图像。

VENC 模块，即视频编码模块。本模块支持多路实时编码，且每路编码独立，编码协议和编码 profile 可以不同。模块包含码率控制和帧率控制以及其它高级编码属性设置。

图2-7 VENC 用例场景



- 用例目录：\mapi\sample\hi3518e\venc\HuaweiLite
- 编译目录：\mapi\sample\hi3518e 或 \mapi\sample\hi3518e\venc\HuaweiLite
- 编译命令：make clean;make

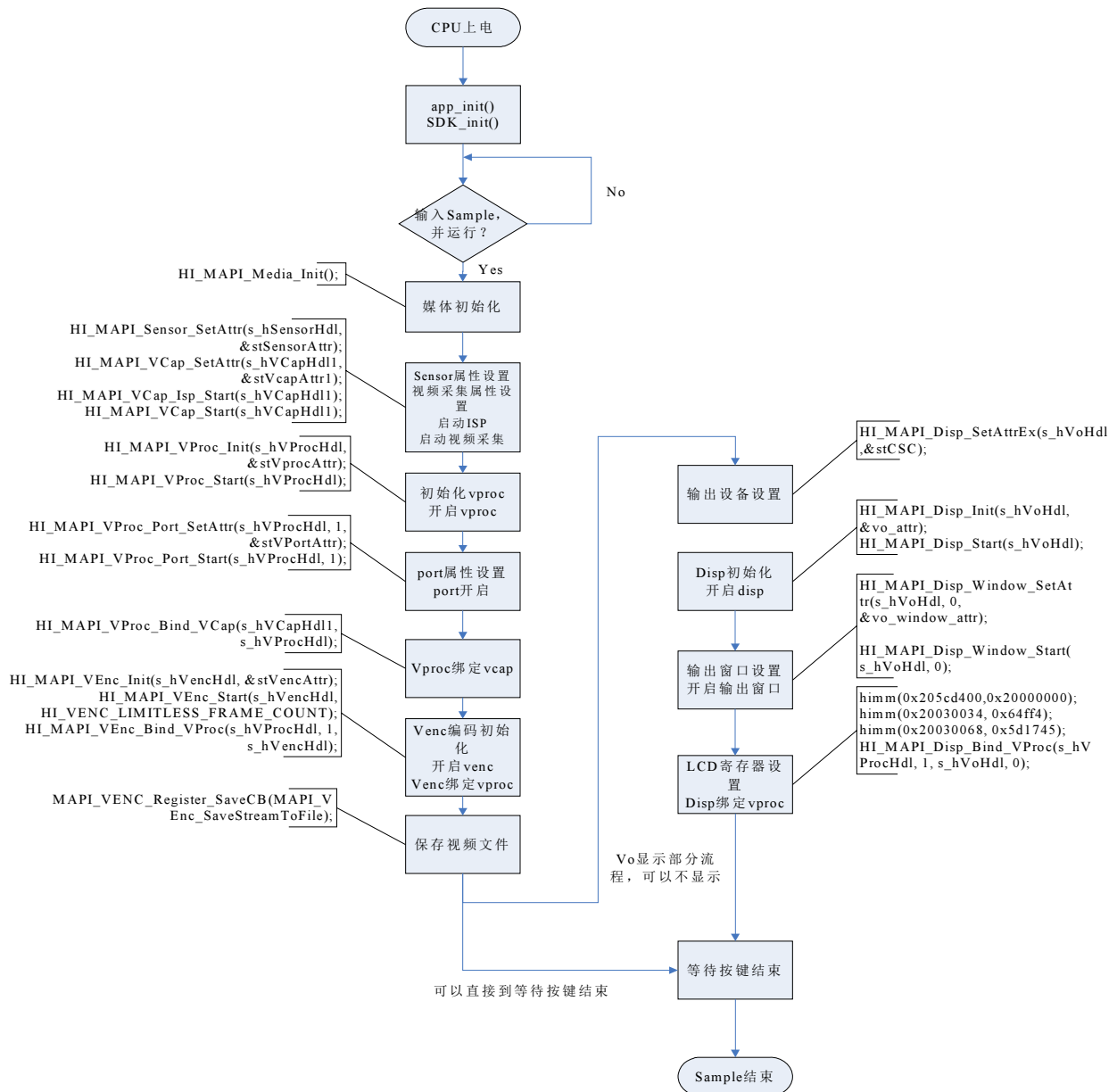


- 生成文件: sample_venc.bin
- 烧写脚本:
 - 烧写到内存（掉电不保存）:
tftp 0x80008000 sample_venc.bin;
go 0x80008000
 - 烧写到 flash（掉电保存）:
tftp 0x80008000 sample_venc.bin;
sf probe 0;
sf erase 0x40000 0x400000;
sf write 0x80008000 0x40000 0x400000;
setenv bootcmd 'sf probe 0;sf read 0x80008000 0x40000 0x400000;go 0x80008000';
saveenv;
烧写到 flash 完成后，需要复位开机，或者执行软件复位“reset”。

2.2.2 视频调试流程图

视频编码和 LCD 显示可同时进行，如果需要同时编码和显示，需要创建两个 vproc 通道，一个绑定编码，一个绑定 VO，详见 [HI_VPORT_ATTR_S](#)，视频调试流程图如图 2-8 所示。视频编码和 LCD 显示也可以分别单独调试。

图2-8 视频调试流程图



2.2.3 LCD 调试

- LCD 宏定义

宏定义文件 sdk_init.c

HI_CHAR* vo_type = "LCD"; //BT656 LCD

HI_CHAR* pin_mux_select = "vo"; //vo net

- 在板端初始化时，对 Hi3516CV200 进行有关 LCD 的引脚复用、频率、驱动能力等配置。LCD 调试流程图如图 2-8 所示。



注意

Hi3516CV200 跟 LCD 有关的寄存器，在初始化过后，有可能会被覆盖掉，调试过程中可通过命令查看寄存器值，方法如图 2-9 所示。如果初始化设置的值被覆盖，可在对 VO 属性设置后再对寄存器进行再赋值，本用例中对 3 个寄存器进行再赋值：

```
himm(0x205cd400,0x20000000); //bit[29] 1:serial; 0 parallel
himm(0x20030034, 0x64ff4);    //8 bit
himm(0x20030068, 0x5d1745);//27M
```

- 读寄存器命令：himd 寄存器地址；
比如 himd 0x205cc400
- 写寄存器命令：himm 寄存器地址；
比如 himm 0x205cc400 0xe0000010;

图2-9 读写寄存器命令

```
Huawei LiteOS #
Huawei LiteOS #
Huawei LiteOS #
Huawei LiteOS # himm 0x205cc400 0xe0000010;

0x205cc400: 0xE0000010 --> 0xe0000010
[END]

Huawei LiteOS #
Huawei LiteOS # himd 0x205cc400

The address begin 0x205cc400,length:0x80

205cc400:e0000010 0011321b 00bf077f 0000020f
205cc410:0021321b 00000000 00000000 00010001
205cc420:00000000 00000000 00000000 0000000e
205cc430:00000020 00000073 000d0303 00300118
205cc440:00000000 00000000 00000000 00000000
205cc450:00000000 00000000 00000000 00000000
205cc460:00000000 00000000 00000000 00000000
205cc470:00000000 00000000 00000000 00000000
The address end 0x205cc480
```

2.2.4 结构体说明

- [HI_MPP_SENSOR_ATTR_S](#): Sensor 属性结构体。
- [HI_MPP_VCAP_ATTR_S](#): 视频采集属性结构体。
- [HI_VPROC_ATTR_S](#): 定义 Vproc 属性结构体。
- [HI_VPORT_ATTR_S](#): 定义 Vport 属性结构体。
- [HI_MPP_VENC_ATTR_S](#): 视频编码结构体。



- [HI_MPP_DISP_ATTREX_S](#): 视频输出设备图像效果结构体。
- [HI_MPP_DISP_ATTR_S](#): 定义视频输出设备属性结构体。
- [HI_MPP_DISP_ATTR_S](#): 定义视频输出公共属性结构体。
- [HI_MPP_DISP_WINDOW_ATTR_S](#): 配置指定视频输出窗口的属性。

HI_MPP_SENSOR_ATTR_S

【定义】

```
/** sensor attribute*/
typedef struct hiHI_MPP_SENSOR_ATTR_S
{
    HI_S32 s32FrameRate;
    HI_MPP_RESOLUTION_S stResolution;
    HI_MPP_WDR_MODE_E enWdrMode;
} HI_MPP_SENSOR_ATTR_S;
```

【应用举例】

```
HI_MPP_SENSOR_ATTR_S stSensorAttr;
HI_S32 s_hSensorHdl=0;

stSensorAttr.s32FrameRate =30;
stSensorAttr.stResolution.u32Width = 1280;
stSensorAttr.stResolution.u32Height = 720;
stSensorAttr.enWdrMode = HI_MPP_WDR_MODE_NONE;
HI_MAPI_Sensor_SetAttr(s_hSensorHdl, &stSensorAttr);
```

HI_MPP_VCAP_ATTR_S

【定义】

```
/** video capture attribute*/
typedef struct hiHI_MPP_VCAP_ATTR_S
{
    HI_S32 s32FrameRate;
    HI_MPP_RESOLUTION_S stResolution;
    HI_MPP_WDR_MODE_E enWdrMode;
    HI_MPP_PIXEL_FORMAT_E enPixelFormat;
} HI_MPP_VCAP_ATTR_S;
```

【应用举例】

```
HI_MPP_VCAP_ATTR_S stVcapAttr1;
HI_S32 s_hVCapHdl1 = 0;
stVcapAttr1.enWdrMode = HI_MPP_WDR_MODE_NONE;
stVcapAttr1.s32FrameRate = 30;
stVcapAttr1.stResolution.u32Width = 1280;
```



```
stVcapAttr1.stResolution.u32Height = 720;
stVcapAttr1.enPixelFormat = HI_MPP_PIXEL_FORMAT_420;

HI_MAPI_VCap_SetAttr(s_hVCapHdl1, &stVcapAttr1);
```

HI_VPROC_ATTR_S

【定义】

```
typedef struct hiVPROC_ATTR_S
{
    HI_VPROC_TYPE_E enVProcType;
    HI_U32 u32MaxW;
    HI_U32 u32MaxH;
} HI_VPROC_ATTR_S;
```

【应用举例】

```
HI_VPROC_ATTR_S stVprocAttr;
HI_S32 s_hVProcHdl = 0;

stVprocAttr.enVProcType = VPROC_TYPE_VIDEO;
stVprocAttr.u32MaxW = 1280;
stVprocAttr.u32MaxH = 720;
HI_MAPI_VProc_Init(s_hVProcHdl, &stVprocAttr);
```

HI_VPORT_ATTR_S

【定义】

```
typedef struct hiVPORT_ATTR_S
{
    HI_S32 s32FrameRate;
    HI_MPP_RESOLUTION_S stResolution;
    HI_MPP_PIXEL_FORMAT_E enPixFormat;
} HI_VPORT_ATTR_S;
```

【应用举例】

```
HI_VPORT_ATTR_S stVPortAttr;
HI_S32 s_hVProcHdl=0;
HI_S32 s_hVPortHdl0=0;
HI_S32 s_hVPortHdl1=1;

stVPortAttr.stResolution.u32Width = 1280;
stVPortAttr.stResolution.u32Height = 720;
stVPortAttr.enPixFormat = HI_MPP_PIXEL_FORMAT_420;
```



```
stVPortAttr.s32FrameRate = 30;
HI_MAPI_VProc_Port_SetAttr(s_hVProcHdl, s_hVPortHdl0, &stVPortAttr);

s32Ret = HI_MAPI_VProc_Port_SetAttr(s_hVProcHdl, s_hVPortHdl0,
&stVPortAttr);
if (HI_SUCCESS != s32Ret)
{
    printf("HI_MAPI_VProc_Port_SetAttr fail s32Ret:%d\n",s32Ret);
    return HI_FAILURE;
}

s32Ret = HI_MAPI_VProc_Port_Start(s_hVProcHdl, s_hVPortHdl0);
if (HI_SUCCESS != s32Ret)
{
    printf("HI_MAPI_VProc_Port_Start fail s32Ret:%d\n",s32Ret);
    return HI_FAILURE;
}
stVPortAttr.stResolution.u32Width = 640;
stVPortAttr.stResolution.u32Height = 480;
stVPortAttr.enPixFormat = HI_MPP_PIXEL_FORMAT_420;
stVPortAttr.s32FrameRate = 30;
HI_MAPI_VProc_Port_SetAttr(s_hVProcHdl, s_hVPortHdl1, &stVPortAttr);

s32Ret = HI_MAPI_VProc_Port_SetAttr(s_hVProcHdl, s_hVPortHdl1,
&stVPortAttr);
if (HI_SUCCESS != s32Ret)
{
    printf("HI_MAPI_VProc_Port_SetAttr fail s32Ret:%x\n",s32Ret);
    return HI_FAILURE;
}

s32Ret = HI_MAPI_VProc_Port_Start(s_hVProcHdl, s_hVPortHdl1);
if (HI_SUCCESS != s32Ret)
{
    printf("HI_MAPI_VProc_Port_Start fail s32Ret:%x\n",s32Ret);
    return HI_FAILURE;
}
}
```

HI_MPP_VENC_ATTR_S

【定义】

```
/** the attribute of video encode*/
typedef struct hiMPP_VENC_ATTR_S
{
```



```
HI_VENC_TYPE_ATTR_S      stVencPloadTypeAttr;  
HI_MPP_VENC_RC_ATTR_S    stRcAttr;  
} HI_MPP_VENC_ATTR_S;
```

【应用举例】

```
HI_MPP_VENC_ATTR_S stVencAttr;  
HI_S32 s_hVencHdl=0;  
  
stVencAttr.stVencPloadTypeAttr.enType = HI_MPP_PAYLOAD_TYPE_H264;  
stVencAttr.stVencPloadTypeAttr.stResolution.u32Width = 640;  
stVencAttr.stVencPloadTypeAttr.stResolution.u32Height = 480;  
stVencAttr.stVencPloadTypeAttr.u32BufSize = 640*480;  
stVencAttr.stVencPloadTypeAttr.stAttrH264e.u32Profile = 0;  
stVencAttr.stVencPloadTypeAttr.stAttrH264e.stParamRef.u32Base = 1;  
stVencAttr.stVencPloadTypeAttr.stAttrH264e.stParamRef.u32Enhance = 0;  
stVencAttr.stVencPloadTypeAttr.stAttrH264e.stParamRef.bEnablePred = 0;  
stVencAttr.stRcAttr.enRcMode = HI_MPP_VENC_RC_MODE_CBR;  
stVencAttr.stRcAttr.stAttrCbr.u32BitRate = 2048;  
stVencAttr.stRcAttr.stAttrCbr.u32Gop = 30;  
HI_MAPI_VEnc_Init(s_hVencHdl, &stVencAttr);
```

HI_MPP_DISP_ATTREX_S

【定义】

```
typedef struct hiMPP_DISP_ATTREX_S  
{  
    HI_MPP_CSC_S stVideoCSC;  
} HI_MPP_DISP_ATTREX_S;
```

【应用举例】

```
HI_MPP_DISP_ATTREX_S stCSC= {0};  
HI_U32 s_hVoHdl;  
stCSC.stVideoCSC.enCscMatrix = VO_CSC_MATRIX_BT709_TO_RGB_PC;  
stCSC.stVideoCSC.u32Contrast = 50;  
stCSC.stVideoCSC.u32Hue = 50;  
stCSC.stVideoCSC.u32Luma = 50;  
stCSC.stVideoCSC.u32Satuature = 50;  
  
HI_MAPI_Dispatch_SetAttrEx(s_hVoHdl, &stCSC);
```

HI_MPP_DISP_ATTR_S

【定义】

```
typedef struct hiMPP_DISP_ATTR_S
```



```
{
    HI_U32 u32BgColor;
    HI_MPP_DISP_INTF_TYPE_E enIntfType;
    HI_MPP_DISP_INTF_SYNC_E enIntfSync;
    HI_MPP_DISP_SYNC_INFO_S stSyncInfo;
    HI_MPP_DISP_ROTATE_E enDispRorate;
} HI_MPP_DISP_ATTR_S;
```

【应用举例】

```
HI_MPP_DISP_ATTR_S vo_attr = {0};
HI_U32 s_hVoHdl=0;

vo_attr.u32BgColor = 0x00;
vo_attr.enIntfType = VO_INTF_LCD_8BIT;
vo_attr.enIntfSync = VO_OUTPUT_USER;

vo_attr.stSyncInfo.bSynm=1;
vo_attr.stSyncInfo.bIop=1;
vo_attr.stSyncInfo.u8Intfb=16;

vo_attr.stSyncInfo.ul6Vact=272;
vo_attr.stSyncInfo.ul6Vbb=8;
vo_attr.stSyncInfo.ul6Vfb=8;

vo_attr.stSyncInfo.ul6Hact=480;
vo_attr.stSyncInfo.ul6Hbb=40;
vo_attr.stSyncInfo.ul6Hfb=56;
vo_attr.stSyncInfo.ul6Hmid=1;

vo_attr.stSyncInfo.ul6Bvact=272;
vo_attr.stSyncInfo.ul6Bvbb=8;
vo_attr.stSyncInfo.ul6Bvfb=8;

vo_attr.stSyncInfo.ul6Hpw=1;
vo_attr.stSyncInfo.ul6Vpw=1;

vo_attr.stSyncInfo.bIdv=0;
vo_attr.stSyncInfo.bIhs=1;
vo_attr.stSyncInfo.bIvs=1;
s32Ret = HI_MAPI_Disp_Init(s_hVoHdl, &vo_attr);
```

HI_MPP_DISP_WINDOW_ATTR_S

【定义】



```
typedef struct hiMPP_DISP_WINDOW_ATTR_S
{
    HI_MPP_RECT_S stRect;
    HI_U32 u32Priority;
} HI_MPP_DISP_WINDOW_ATTR_S;
```

【应用举例】

```
HI_MPP_DISP_WINDOW_ATTR_S vo_window_attr;
s_hVoHdl=0;

vo_window_attr.stRect.u32X = 0;
vo_window_attr.stRect.u32Y = 0;
vo_window_attr.stRect.u32Width = 480;
vo_window_attr.stRect.u32Height = 272;
vo_window_attr.u32Priority = 0;
HI_MAPI_Disp_Window_SetAttr(s_hVoHdl, 0, &vo_window_attr);
```

2.2.5 Proc 调试信息

2.2.5.1 VI

打印文本信息

vi 信息查看命令: cat /proc/umap/vi

Huawei LiteOS # cat /proc/umap/vi

Huawei LiteOS #

[VIU] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov 2 2016, 09:47:44]

-----MODULE PARAM-----

detect_err_frame	drop_err_frame	stop_int_level
10	0	0

-----VI-VPSS ONLINE STATE-----

OnlineState
online

-----VI MODULE STATE-----

ModuleState
Started

-----VI DEV ATTR1-----

Dev	IntfM	WkM	ComMsk0	ComMsk1	ScanM	AD0	AD1	AD2
0	MIPI	1Mux	fff00000	0	P	-1	-1	-1



```

-----VI DEV ATTR2-----
      AD3      Seq  DPath  DType   DRev   CapX   CapY   CapW   CapH  BasW
BasH  Comp  HReph  VReph
      -1      N/A    ISP    RGB     N     0     0    1280   720  1280
720    N  NONE  NONE

-----VI HIGH DEV ATTR 1-----
      Dev  InputM      WkM      ComMsk0      ComMsk1      ScanM      AD0      AD1      AD2

-----VI HIGH DEV ATTR 2-----
      AD3      Seq  CombM  CompM   ClkM     Fix     FldP  DPath  DType
DRev

-----VI HIGH DEV ATTR 3-----
      CapX      CapY      CapW      CapH  BasW  BasH  Comp  HReph  VReph

-----VI PHYCHN ATTR 1-----
      PhyChn      CapX      CapY      CapW      CapH      DstW      DstH  CapSel  Mirror
Flip
           0          0          0    1280      720    1280      720   both      N      N

-----VI PHYCHN ATTR 2-----
      IntEn  PixFom  SrcRat  DstRat   Comp
           Y   SP420     30     30      N

-----VI PHYCHN STATUS 1-----
      PhyChn      Dev      IntCnt  VbFail      LosInt  TopLos  BotLos  BufCnt
IntT
           0          0        112      0          0       0       0       0      131

-----VI PHYCHN STATUS 2-----
      SendT   Field  Stride  MaxIntT      IntGapT      MaxGapT  LIntCnt  ThrCnt
AutoDis
           0    frm   1280    156      33328      34720      0       0       0

-----VI PHYCHN STATUS 3-----
      CasAutD  TmgErr      ccErrN      IntRat
           0       0          0        31

-----VI LDC ATTR-----
      PhyChn  ViewType  Ratio  minRatio  COffX  COffY  Enable

-----VI FLASH ATTR-----
      Dev   Mode  StartTime  DuraTime  InterVal  CapIdx  Enable  FlashedNum

```



```
0  Once      0      0      0      0      N      0

-----VI CSC ATTR-----
Dev  Type HueVal  ContrVal  LumaVal  StatuVal  TVMode
0    709   50     50     50     50      N

-----VI DCI ATTR-----
Dev  Enable BlackGain ContrGain LightGain  ManBlendEn  BlendRatio
BlackStretchEn
0    Y      32      32      32      NA      NA      NA

-----VI WDR ATTR-----
Dev  Mode BufNum  DstW  DstH  PoolId  VcNum  DesNum  State
bCompress
0    NONE  0      0      0      -1     0      0      NONE  N

-----VI WDR DES STATUS-----
Dev  Idx  IntGap  IntCnt  CcErrCnt

-----VI WDR SRC STATUS-----
Dev  Idx  IntGap  IntCnt  CcErrCnt

-----VI WDR COMBINE STATUS-----
Dev  IntGap  IntCnt  CcErrCnt
0    0      0      0
```



图2-10 VI 调试截图

```
cat /proc/umap/vi

Huawei LiteOS #
[VIU] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov 2 2016, 09:47:44]

-----MODULE PARAM-----
detect_err_frame  drop_err_frame  stop_int_level
                10                0                0

-----VI-VPSS ONLINE STATE-----
OnlineState
online

-----VI MODULE STATE-----
ModuleState
Started

-----VI DEV ATTR1-----
Dev  IntfM  WkM  ComMsk0  ComMsk1  ScanM  AD0  AD1  AD2
0    MIPI  1Mux  fff00000  0        P     -1   -1   -1

-----VI DEV ATTR2-----
AD3  Seq  DPath  DType  DRev  CapX  CapY  CapW  CapH  BasW  BasH  Comp  HReph  VReph
-1   N/A  ISP    RGB    N     0     0     1280  720  1280  720  N     NONE  NONE

-----VI HIGH DEV ATTR 1-----
Dev  InputM  WkM  ComMsk0  ComMsk1  ScanM  AD0  AD1  AD2

-----VI HIGH DEV ATTR 2-----
AD3  Seq  CombM  CompM  ClkM  Fix  FldP  DPath  DType  DRev

-----VI HIGH DEV ATTR 3-----
CapX  CapY  CapW  CapH  BasW  BasH  Comp  HReph  VReph

-----VI PHYCHN ATTR 1-----
PhyChn  CapX  CapY  CapW  CapH  DstW  DstH  CapSel  Mirror  Flip
0        0    0    1280  720  1280  720  both    N      N

-----VI PHYCHN ATTR 2-----
IntEn  PixFom  SrcRat  DstRat  Comp
Y      SP420  30      30      N

-----VI PHYCHN STATUS 1-----
PhyChn  Dev  IntCnt  VbFail  LosInt  TopLos  BotLos  BufCnt  IntT
0        0    503    0       0       0       0       0       131

-----VI PHYCHN STATUS 2-----
SendT  Field  Stride  MaxIntT  IntGapT  MaxGapT  LIntCnt  ThrCnt  AutoDis
0      frm  1280   152     33289   34732   0       0       0
```



```

-----VI PHYCHN STATUS 3-----
CasAutD  TmgErr  ccErrN  IntRat
    0      0      0      31

-----VI LDC ATTR-----
PhyChn  ViewType  Ratio minRatio  COffX  COffY  Enable

-----VI FLASH ATTR-----
Dev  Mode  StartTime  DuraTime  InterVal  CapIdx  Enable  FlashedNum
  0   Once      0      0      0      0      N      0

-----VI CSC ATTR-----
Dev  Type  HueVal  ContrVal  LumaVal  StatuVal  TVMode
  0   709   50    50      50      50      N

-----VI DCI ATTR-----
Dev  Enable  BlackGain  ContrGain  LightGain  ManBlendEn  BlendRatio  BlackStretchEn
  0    Y      32      32      32      NA      NA      NA

-----VI WDR ATTR-----
Dev  Mode  BufNum  DstW  DstH  PoolId  VcNum  DesNum  State  bCompress
  0  NONE    0      0      0     -1      0      0  NONE    N

-----VI WDR DES STATUS-----
Dev  Idx  IntGap  IntCnt  CcErrCnt

-----VI WDR SRC STATUS-----
Dev  Idx  IntGap  IntCnt  CcErrCnt

-----VI WDR COMBINE STATUS-----
Dev  IntGap  IntCnt  CcErrCnt
  0      0      0      0

```

参数说明

具体每个参数含义见《HiMPP IPC V2.0 媒体处理软件开发参考.pdf》，参数设置是否成功，可通过该打印信息查看。Proc 信息中 VI PHYCHN STATUS 的 IntCnt 表示中断次数，每次帧中断中取值增加 1，如果 VI 工作不正常，此值不会增加。

2.2.5.2 VENC

文本打印信息

venc 信息查看命令：cat /proc/umap/venc

Huawei LiteOS # cat /proc/umap/venc

Huawei LiteOS #

[VENC] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Aug 31 2016, 15:37:06]

```

-----MODULE PARAM-----
VencBufferCache
    0

-----VENC CHN ATTR 1-----
ID  Width  Height  Type  ByFrame  Timeout  Sequence  LeftBytes
LeftFrm  CurPacks  prio

```



```
0      640    480    96      Y      0      105      0
0      1      0

-----VENC CHN ATTR 2-----
VeStr  OsdStr  SrcFr  TarFr      Timeref  PixFmt  PicAddr
Y      N      -1     -1      220  YUV420  0x85aea000

-----VENC CHN RECEIVE STAT-----
ID      Start      StartEx  RecvLeft  EncLeft
0      1      0      0      0

-----VENC VPSS QUERY-----
ID      Query      QueryOk  QueryFR      Invld      Full      VbFail
QueryFail  InfoErr      Stop
0      0      0      0      0      0      0
0      0      0

-----VENC SEND1-----
ID      VpssSnd      VInfErr  OthrSnd      OInfErr      Send      Stop
Full      CropErr      DirectSnd  SizeErr
0      106      0      0      0      106      0
0      0      106      0

-----VENC SEND2-----
ID      SendVgs      StartOk  StartFail      IntOk      IntFail
SrcAdd      SrcSub      DestAdd      DestSub
0      0      0      0      0      0      0
0      0      0

-----VENC PIC QUEUE STATE-----
ID      Free      Busy      Vgs
0      5      1      0

-----VENC CHNL INFO-----
ID      Inq      InqOk      Start      StartOk      Config      VencInt
ChaResLost  OverLoad
0      457      106      106      106      106      105
0      0

-----VENC CROP INFO-----
ID  CropEn  StartX  StartY  Width  Height
0      N      0      0      0      0

-----VENC STREAM STATE-----
```



ID	FreeCnt	BusyCnt	UserCnt	UserGet	UserRls	GetTimes
Interval		FrameRate				
0	1	0	0	105	105	105
29997	30					

图2-11 VENC 调试截图

```
Huawei LiteOS # cat /proc/umap/venc

Huawei LiteOS #
[VENC] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Aug 31 2016, 15:37:06]

-----MODULE PARAM-----
VencBufferCache
0

-----VENC CHN ATTR 1-----
ID      Width  Height  Type  ByFrame  Timeout  Sequence  LeftBytes  LeftFrm  CurPacks  prio
0        640    480    96    Y         0        105        0         0        1        0

-----VENC CHN ATTR 2-----
VeStr   OsdStr   SrcFr   TarFr   Timeref  PixFmt  PicAddr
Y        N        -1      -1      220     YUV420  0x85aea000

-----VENC CHN RECEIVE STAT-----
ID      Start  StartEx  RecvLeft  EncLeft
0        1        0        0         0

-----VENC VPSS QUERY-----
ID      Query  QueryOk  QueryFR  Invld  Full  VbFail  QueryFail  InfoErr  Stop
0        0        0        0        0      0      0        0         0        0

-----VENC SEND1-----
ID      VpssSnd  VInfErr  OthrSnd  OInfErr  Send  Stop  Full  CropErr  DirectSnd  SizeErr
0        106      0        0        0        106  0      0      0        106      0

-----VENC SEND2-----
ID      SendVgs  StartOk  StartFail  IntOk  IntFail  SrcAdd  SrcSub  DestAdd  DestSub
0        0        0        0        0      0      0      0      0        0

-----VENC PIC QUEUE STATE-----
ID      Free  Busy  Vgs
0        5    1    0

-----VENC CHNL INFO-----
ID      Inq  InqOk  Start  StartOk  Config  VencInt  ChaResLost  OverLoad
0        457  106   106   106   106    105        0         0

-----VENC CROP INFO-----
ID      CropEn  StartX  StartY  Width  Height
0        N        0        0      0      0

-----VENC STREAM STATE-----
ID      FreeCnt  BusyCnt  UserCnt  UserGet  UserRls  GetTimes  Interval  FrameRate
0        1        0        0        105     105     105     29997     30
```

参数说明

具体每个参数含义见《HiMPP IPC V2.0 媒体处理软件开发参考.pdf》，红框圈出的表示 vpss 发送给 venc 的帧数据，以及启动编码次数和成功的次数。venc 编码正常，帧数据会递增，需要注意，venc 需绑定 vpss 才能接收到 vpss 的帧数据。

2.2.5.3 VO

文本打印信息

vo 信息查看命令：cat /proc/umap/vo



Huawei LiteOS # cat /proc/umap/vo

Huawei LiteOS #

[VOU] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov 2 2016, 09:47:44]

-----DEV CONFIG-----

DevId	DevEn	Mux1	Mux2	Mux3	InfSync	BkClr
DevFrt						
0	Y	LCD_8BIT			USER	0 60

-----MODULE PARAM-----

detectCycle	transparentTransmit	lowPowerMode
0	N	Y

-----DEV VDAC STATUS-----

DevId	VDAC
0	N

-----VIDEO LAYER STATUS-----

LayerId	VideoEn	ClustMode	PixFmt	ImgW	ImgH	DispW	DispH	DispFrt
DoubFrm	Toleration	Priority						
0	Y	N	SP420	480	272	480	272	60
10000000	0							N

-----VIDEO LAYER STATUS 2-----

layerId	VideoEn	EnChNum	Matrix	Luma	Cont	Hue	Satu
0	Y	1	4	50	50	50	50

-----VIDEO LAYER STATUS 3-----

layerId	DevId	SetBeg	SetEnd	PartitionMode	bSDVgsBypass	u32BufLen
0	0	N	N	Single	Y	4

-----CHN BASE INFO -----

LayerId	ChnId	ChnEn	Prio	DeFlk	ChnX	ChnY	ChnW	ChnH	DispX	DispY
bSnap	Field									
0	0	Y	0	N	0	0	480	272	-1	-1
									N	both

-----CHN PLAY INFO 1-----

LayerId	ChnId	Batch	Show	Pause	Step	Revrs	Refsh	Thrshd	ChnFrt	ChnGap
0	0	N	Y	N	N	N	2	60	16666	

-----CHN PLAY INFO 2-----

LayerId	ChnId	DisplayPts	PrePts	CurrPts
---------	-------	------------	--------	---------



```
ScalePts          SetPts          RecvCurPts
0      0          19562985          19562985          19562985
0              -1              0

-----BySingle CHN STATUS 1-----
LayerId  ChnId  Job    Task    LCnt    SCnt    ChRpt    DRpt  CBusy  DBusy
ShouD Dsped b2Scl  ChnAddr DispAddr
      0      0      0      0      0      364      292      1      0      0      1
1      N 85fbbc00 85fbbc00

-----BySingle CHN STATUS 2-----
LayerId  ChnId  bBorder  BorderWidth  Color  ChnFreeNum  ChnBusyNum
DispFreeNum DispBusyNum
0      0      N      0      0      7      0      13      0

-----CHN OTHER INFO-----
LayerId  ChnId  bZoom  ZmTyp  ZoomX  ZoomY  ZoomW  ZoomH  SrcW
SrcH
0      0      N      0      0      0      0      0      0

-----LAYER CSC PARAM-----
LAYERID  Matrix  Luma  Cont  Hue  Satu
0      5      50      50      50      50
```

图2-12 sys 调试截图

```
Huawei LiteOS # cat /proc/umap/v0
Huawei LiteOS #
[VOU] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov 2 2016, 09:47:44]

-----DEV CONFIG-----
DevId  DevEn  Mux1    Mux2    Mux3    InfSync  BkClr  DevFrt
0      Y      LCD_8BIT  USER    0      60

-----MODULE PARAM-----
detectCycle transparentTransmit lowPowerMode
0      N      Y

-----DEV VDAC STATUS-----
DevId  VDAC
0      N

-----VIDEO LAYER STATUS-----
LayerId VideoEn ClustMode PixFmt  ImgW  ImgH  DispW  DispH  DispFrt  DoubFrm  Toleration  Priority
0      Y      N      SP420  480   272   480   272   60      N      10000000  0

-----VIDEO LAYER STATUS 2-----
layerId VideoEn EnChNum  Matrix  Luma  Cont  Hue  Satu
0      Y      1      4      50   50   50   50

-----VIDEO LAYER STATUS 3-----
layerId  DevId  SetBeg  SetEnd  PartitionMode  bSDVgsBypass  u32BufLen
0      0      N      N      Single      Y      4

-----CHN BASE INFO -----
LayerId  ChnId  ChnEn  Prio  DeFlk  ChnX  ChnY  ChnW  ChnH  DispX  DispY  bSnap  Field
0      0      Y      0      N      0      0      480  272  -1    -1    N      both

-----CHN PLAY INFO 1-----
LayerId  ChnId  Batch  Show  Pause  Step  Revrs  Refsh  Thrshd  ChnFrt  ChnGap
0      0      N      Y      N      N      N      N      2      60      16666
```



```
-----CHN PLAY INFO 2-----
LayerId  ChnId  DisplayPts  PrePts  CurrPts  ScalePts  SetPts  RecvCurPts
   0      0    19562985  19562985  19562985      0      -1         0

-----BySingle CHN STATUS 1-----
LayerId  ChnId  Job  Task  LCnt  SCnt  ChRpt  DRpt  CBusy  DBusy  ShouD  Dsped  b2Scl  ChnAddr  DispAddr
   0      0   0    0    0    364   292    1     0     0     1     1     N  85fbbc00  85fbbc00

-----BySingle CHN STATUS 2-----
LayerId  ChnId  bBorder  BorderWidth  Color  ChnFreeNum  ChnBusyNum  DispFreeNum  DispBusyNum
   0      0     N         0         0         7         0         13         0

-----CHN OTHER INFO-----
LayerId  ChnId  bZoom  ZmTyp  ZoomX  ZoomY  ZoomW  ZoomH  SrcW  SrcH
   0      0     N     0     0     0     0     0     0     0

-----LAYER CSC PARAM-----
LAYERID  Matrix  Luma  Cont  Hue  Satu
   0      5    50   50   50   50
```

参数说明

每个参数具体含义见《HiMPP IPC V2.0 媒体处理软件开发参考.pdf》，vo 文本打印信息可以查看 vo 相关的属性设置，BySingle CHN STATUS 1 的 SCnt，表示通道接收其他模块发送帧计数。如果该数值为 0，表示 vo 还没有收到图像数据；如果该数值一直不增加，表示 vo 接收不到图像，可用于判断是否正常接收图像。需要注意，VO 绑定 VProc 才能接收图像数据。

2.2.5.4 sys 系统信息查看

sys 系统文本打印信息

sys 信息查看命令：cat /proc/umap/sys

Huawei LiteOS # cat /proc/umap/sys

Huawei LiteOS #

```
[SYS] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov 2
2016, 09:47:44]
```

```
-----MODULE PARAM-----
```

```
vi_vpss_online  sensor0  sensor1
                  1  ov9732  xtp
```

```
-----MODULE STATUS-----
```

```
Status
```

```
run
```

```
-----SCALE COEFF INFO-----
```

RangeLevel	RangeValue	HorLum	HorChr	VerLum	VerChr
RANGE_0	(0, 1/4)	LEVEL_0	LEVEL_0	LEVEL_0	LEVEL_0
RANGE_1	[1/4, 1/3)	LEVEL_1	LEVEL_0	LEVEL_1	LEVEL_0
RANGE_2	[1/3, 1/2)	LEVEL_2	LEVEL_1	LEVEL_2	LEVEL_1
RANGE_3	[1/2, 3/4)	LEVEL_3	LEVEL_2	LEVEL_3	LEVEL_2
RANGE_4	[3/4, 1)	LEVEL_3	LEVEL_2	LEVEL_4	LEVEL_2
RANGE_5	[1, 1]	LEVEL_4	LEVEL_3	LEVEL_5	LEVEL_3
RANGE_6	(1, MAX)	LEVEL_5	LEVEL_2	LEVEL_6	LEVEL_2



```
-----MEM TABLE-----
Mod          ModName    Dev    Chn          MmzName

-----BIND RELATION TABLE-----
  FirMod  FirDev  FirChn  SecMod  SecDev  SecChn  TirMod  TirDev  TirChn
SendCnt  rstCnt
vpss      0      1    venc      0      0    null      0      0      0      0
vpss      0      1    vou       0      0    null      0      0      0      0
viu       0      0    vpss      0      0    null      0      0      0      0
```

图2-13 sys 调试截图

```
Huawei LiteOS #
Huawei LiteOS #
Huawei LiteOS # cat /proc/umap/sys

Huawei LiteOS #
[SYS] Version: [Hi3518EV200_MPP_V1.0.0.0 B010 Release], Build Time[Nov  2 2016, 09:47:44]

-----MODULE PARAM-----
vi_vpss_online  sensor0  sensor1
                  1    ov9732  xtp

-----MODULE STATUS-----
Status
run

-----SCALE COEFF INFO-----
RangeLevel  RangeValue  HorLum  HorChr  VerLum  VerChr
RANGE_0     (0, 1/4)    LEVEL_0 LEVEL_0  LEVEL_0 LEVEL_0
RANGE_1     [1/4, 1/3]  LEVEL_1 LEVEL_0  LEVEL_1 LEVEL_0
RANGE_2     [1/3, 1/2]  LEVEL_2 LEVEL_1  LEVEL_2 LEVEL_1
RANGE_3     [1/2, 3/4]  LEVEL_3 LEVEL_2  LEVEL_3 LEVEL_2
RANGE_4     [3/4, 1]    LEVEL_3 LEVEL_2  LEVEL_4 LEVEL_2
RANGE_5     [1, 1]      LEVEL_4 LEVEL_3  LEVEL_5 LEVEL_3
RANGE_6     (1, MAX)   LEVEL_5 LEVEL_2  LEVEL_6 LEVEL_2

-----MEM TABLE-----
Mod          ModName    Dev    Chn          MmzName

-----BIND RELATION TABLE-----
FirMod  FirDev  FirChn  SecMod  SecDev  SecChn  TirMod  TirDev  TirChn  SendCnt  rstCnt
vpss     0      1    venc      0      0    null      0      0      0      0
vpss     0      1    vou       0      0    null      0      0      0      0
viu      0      0    vpss      0      0    null      0      0      0      0

Huawei LiteOS #
Huawei LiteOS #
Huawei LiteOS #
```

参数说明

cat /proc/umap/sys 主要用于查看各设备的绑定关系，SecMod 绑定 FirMod，图 2-13 中绑定关系为：

- vpss 绑定 viu，viu 指视频输入单元。



- venc 绑定 vpss
- vou 绑定 vpss

2.3 Wifi 调试

2.3.1 概述

MobileCam SDK 支持使用 6214 和 8801 wifi，两者都有 AP 模式和 STA 模式，其中 6214 支持镜像，实现快速启动，8801 不需要做镜像，本文以 6214 的 STA 模式为例做说明，如图 2-14 所示，选择“1”。

相关软件组件：板端 uboot 代码、uboot 环境变量设置、板端应用代码、MCU 代码、手机客户端。

说明

因为 WiFi 待机唤醒是一整套方案，与具体的电路紧密相关。

- 用例目录：\mapi\sample\hi3518e\ wifi_sta_6214\HuaweiLite
 - wifi 库文件：libmhd.a、libmhd_util.a
 - 编译目录： \mapi\sample\hi3518e\ wifi_sta_6214 \HuaweiLite
 - 编译命令：make liteos_image
 - 生成文件及目录：...\wifi_sta_6214\out\burn\sample.bin
 - 烧写脚本：
 - 烧写到内存（掉电不保存）：

```
tftp 0x80008000 sample.bin;
go 0x80008000
```
 - 烧写到 flash（掉电保存）：

```
tftp 0x80008000 sample.bin;
sf probe 0;
sf erase 0x40000 0x400000;
sf write 0x80008000 0x40000 0x400000;
setenv bootcmd 'sf probe 0;sf read 0x80008000 0x40000 0x400000;go 0x80008000';
saveenv;
```
 - 设置镜像启动脚本：

```
setenv bootimg 'sf probe 0;sf read 0x80008000 0x450000 0x300000;go 0x80008000';
saveenv;
```
- 烧写到 flash 完成后，需要复位开机，或者执行软件复位“reset”。



注意

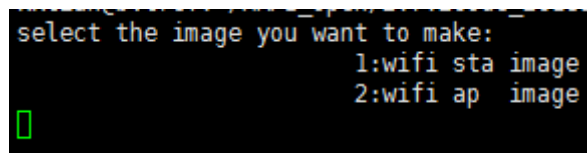
wifi 用例编译需添加两个库文件，6214 wifi 支持做镜像，需要设置镜像启动脚本“bootimg”。烧写完成后可通过命令“pri”查看启动脚本，“bootimg”设置起始地址跟代码两个宏定义有关：

```
#define FLASH_APP_IMG_ADDR (0x40000)
#define FLASH_WIFI_IMG_ADDR (0x450000)
```

- FLASH_APP_IMG_ADDR 表示正常启动 app 业务地址。
- FLASH_WIFI_IMG_ADDR 表示 wifi 唤醒，镜像启动地址。

6214 做镜像编译 make liteos_image 之后，必须要执行命令 make liteos_image_clean 清除镜像改动过的脚本，才能再编译其他用例，否则编译报错。

图2-14 镜像编译模式选择



```
select the image you want to make:
1:wifi sta image
2:wifi ap image
█
```

2.3.2 Wifi 待机

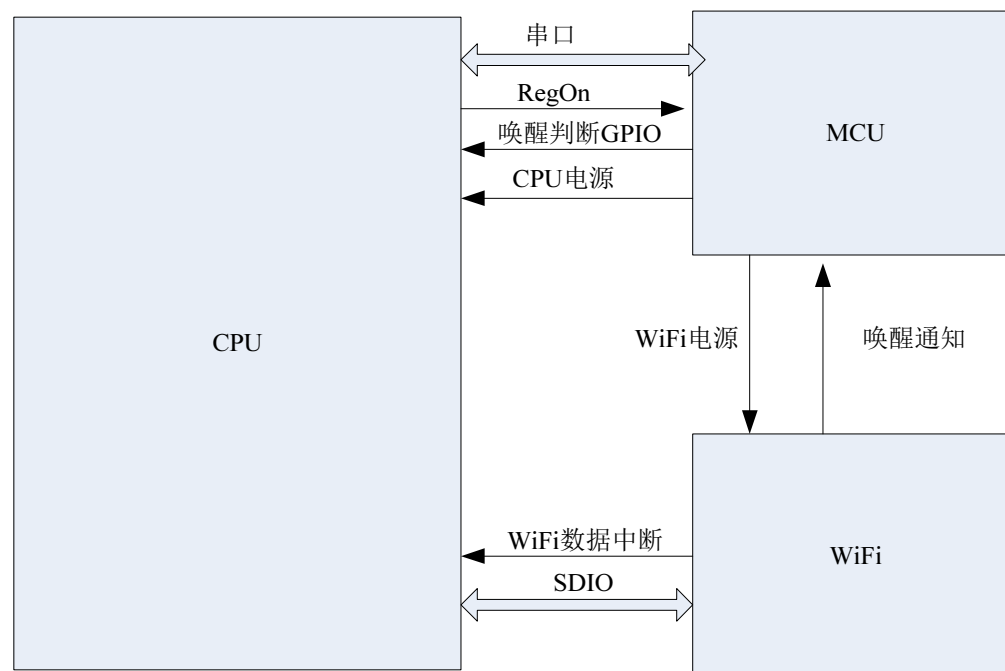
WiFi 待机指的是板端 CPU 下电，WiFi 模组不下电的工作状态。具备 WiFi 待机功能的电路结构如[图 2-15](#) 所示。



注意

在整个文档里都有对[图 2-15](#) 的引用。MCU 控制了 WiFi 和 CPU 的电源，待机状态时，MCU 本身不下电。

图2-15 Sample 电路框图



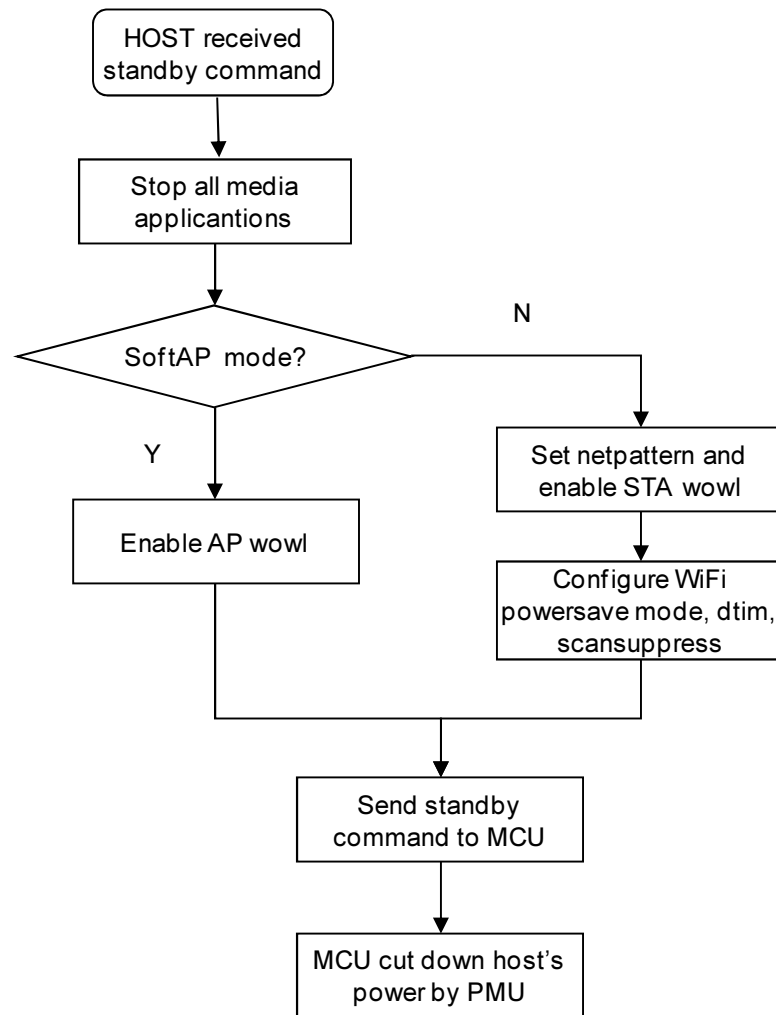
WiFi 待机的具体过程如下：

1. 手机客户端发送特定的指令到板端。
2. 板端接收到待机请求后，CPU 发送指令，使 WiFi 模组进入到低功耗的待机模式。
3. Cpu 向 MCU 发送关机请求，接着被 MCU 下电。

此时 WiFi 模组并未掉电，WiFi 工作在 STA 模式时，WiFi 模组与路由器的连接并未断开。此时 CPU 被 MCU 控制下电，MCU 自身也进入低功耗模式，此时板端的整体功耗很低。

STA 模式时，WiFi 模组定时与路由器通信，时间间隔一般设置为 1s，设置的最小单位为 100ms，在待机时设置给 WiFi 模组。时间间隔缩小，唤醒时的速度会加快，但是功耗会增加。

图2-16 待机流程图



应用举例：

```
void hi_wifi_standby(int mode)
{
    if (mode == SOFTAP) {
        mhd_wowl_ap_enable();
    } else {
        mhd_wowl_sta_add("0x983B16F8F39C", 54);
        mhd_wowl_sta_enable();
        mhd_wifi_set_scansuppress(1);
        mhd_sta_set_powersave(1, 0);
        mhd_sta_set_bcn_li_dtim(5);
    }
    // TODO: Send standby command to MCU
}
```



2.3.3 Wifi 唤醒

WiFi 唤醒是指 WiFi 模组接收外部信号（由手机客户端发送），传递信号给 MCU，从而控制 CPU 上电开机的过程。

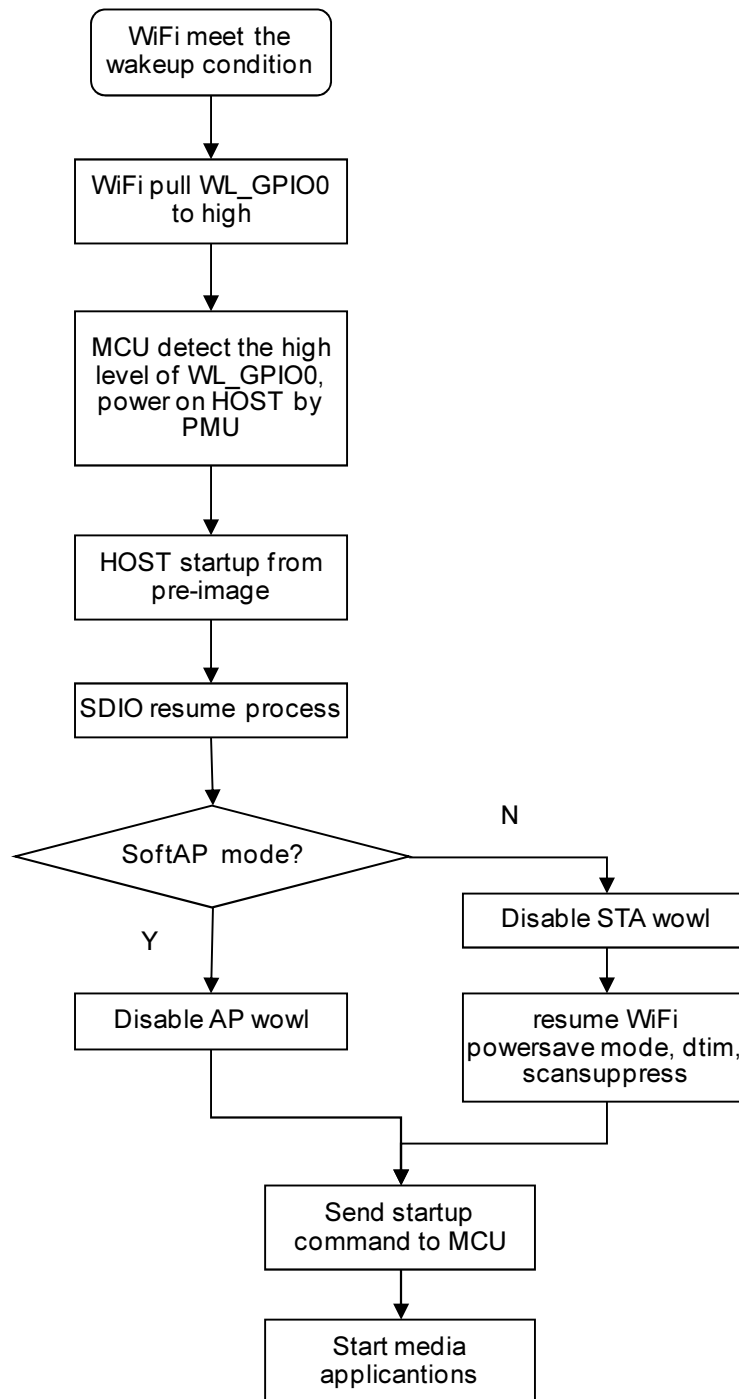
唤醒过程如下：

- 步骤 1. 点击手机客户端唤醒按钮，发送唤醒指令到 WiFi 模组。
- 步骤 2. Wifi 模组收到信号，给 MCU 传递一个脉冲信号，MCU 退出低功耗模式，对 CPU 上电。
- 步骤 3. Cpu 上电后，根据被 MCU 拉高的某个 GPIO 引脚（此引脚状态用来区分正常上电开机和唤醒开机），由 uboot 读取之前保存的 WiFi 镜像启动。

因为 WiFi 模组与路由器的连接一直保持着，在唤醒后几百毫秒内，就可以把业务数据发送出去，这也正是 WiFi 唤醒相当于正常开机的最大优点。

----结束

图2-17 唤醒流程图



应用举例：

```

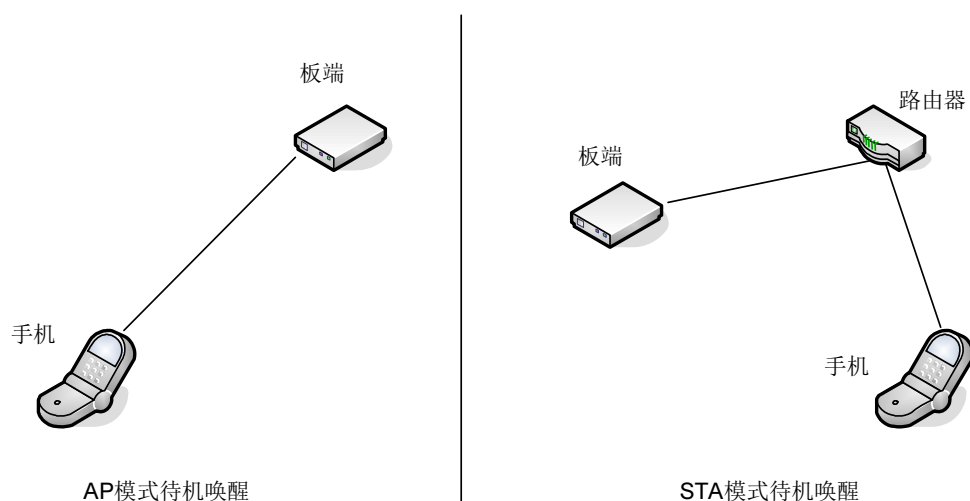
void hi_wifi_wakeup(int mode)
{
    mmc_host_resume(NULL);
    if (mode == SOFTAP) {
        mhd_wowl_ap_disable();
    }
}
  
```

```
else {  
    mhd_wowl_sta_disable();  
    mhd_wifi_set_scansuppress(0);  
    mhd_sta_set_powersave(2, 200);  
    mhd_sta_set_bcn_li_dtim(0);  
}  
// TODO: Send startup command to MCU  
}
```

2.3.4 AP 模式和 STA 模式待机唤醒区别

AP 模式和 STA 模式待机唤醒的网络拓扑图如图 2-18 所示，AP 模式只需要手机直接连接到板端的热点即可，STA 模式需要一个无线路由器作为网络的中心节点。STA 模式下，因为手机和板端不是直接相连，所以为了让板端和手机可以发现对方，使用飞鸽协议来做设备发现。飞鸽协议详细信息请查阅相关资料。

图2-18 AP 模式和 STA 模式网络拓扑图

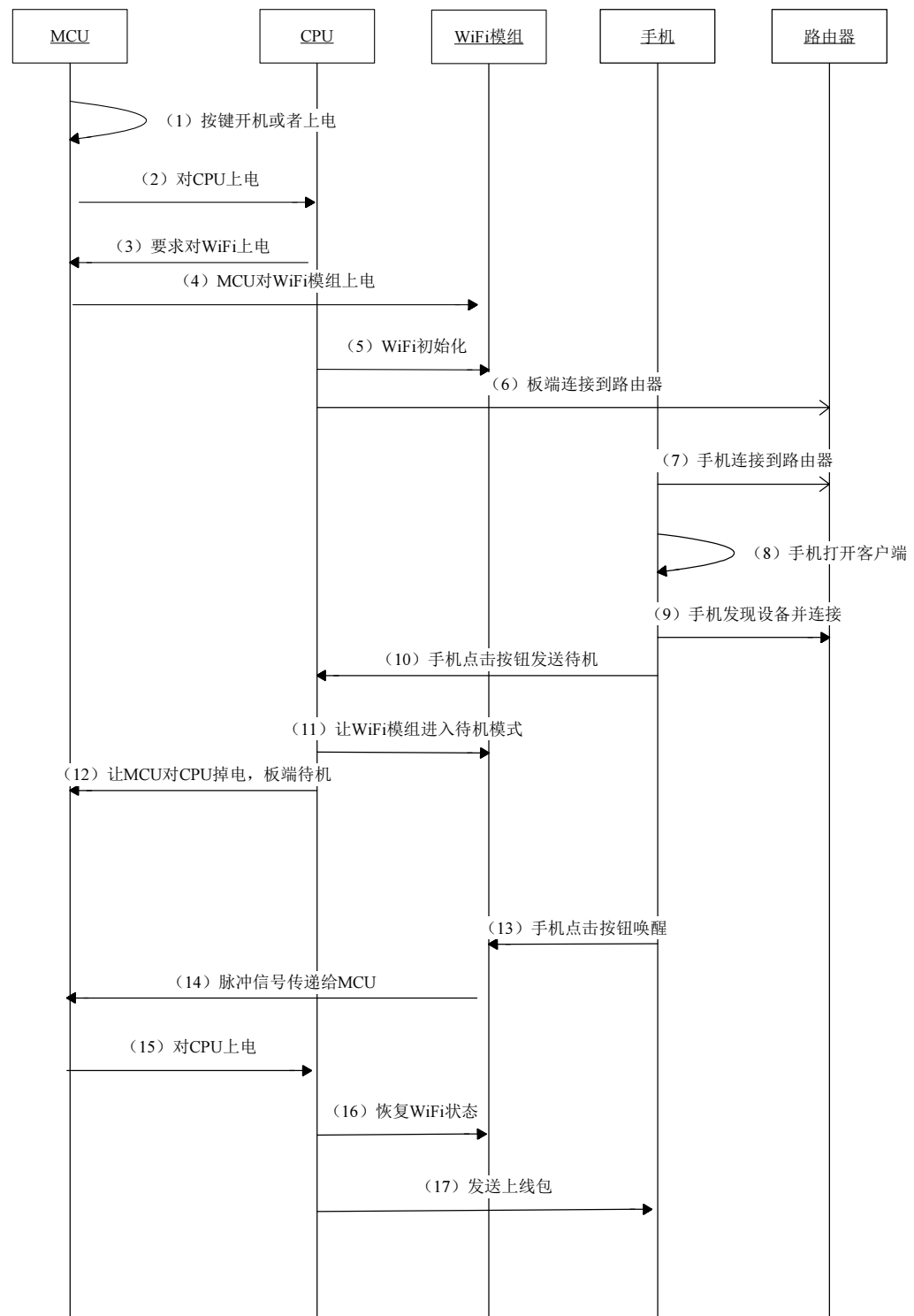


2.3.5 待机唤醒时序步骤

一次完整的 STA 模式从板端上电到 WiFi 待机再到 WiFi 唤醒的完整过程如图 2-19 所示。如果是 AP 模式，则没有路由器那一部分的相关操作。



图2-19 STA 模式唤醒时序图



各个步骤的说明如下：

步骤 1. (1) 按键开机或者上电。Sample 所用电路板如果带电池，则需要用按键开机，否则上电自动开机。



- 步骤 2. (2) MCU 对 CPU 上电。MCU 控制了 CPU 的电源，板端上电后，先是 MCU 运行，MCU 再控制 CPU 的上电。
- 步骤 3. (3) CPU 要求 MCU 对 WiFi 模组上电。这个信号是通过图 2-15 RegOn 传递给 MCU 的。如果这个引脚电平为高，则 MCU 对 WiFi 模组上电。
- 步骤 4. (4) MCU 对 WiFi 模组上电。因为 MCU 控制了 WiFi 的电源。
- 步骤 5. (5) WiFi 初始化。
- 步骤 6. (6) 板端连接到路由器。STA 模式需要这一步，AP 模式则不需要。
- 步骤 7. (7) 手机连接到路由器。STA 模式需要这一步，AP 模式为手机连接到板端。
- 步骤 8. (8) 手机打开客户端。STA 模式打开 WakeDemo 客户端，AP 模式则打开 StandbyWake 客户端。
- 步骤 9. (9) 手机发现设备并连接。AP 模式在步骤 (7) 已经连接好了。STA 模式需要在 WakeDemo 客户端连接设备。
- 步骤 10. (10) 手机点击按钮发送待机。此时手机会发送一组待机命令给板端的 WiFi 模组。
- 步骤 11. (11) CPU 让 WiFi 模组进入待机模式。CPU 通过 SDIO 接口，向 WiFi 传递一组命令，让 WiFi 进入待机模式。
- 步骤 12. (12) CPU 让 MCU 对自身掉电。CPU 向 MCU 发送串口命令，要求 MCU 对自身掉电。MCU 接到命令后，控制 CPU 的电源，让 CPU 进入掉电状态。此时板端进入 WiFi 待机的状态。
- 步骤 13. (13) 手机点击唤醒按钮。这个时候，用户想要唤醒板端，则点击手机客户端上的唤醒按钮，手机端发送一组特定指令到板端的 WiFi 模组。
- 步骤 14. (14) WiFi 模组传递脉冲信号给 MCU。因为 MCU 控制 CPU 的电源，所以唤醒时必须先通知 MCU。
- 步骤 15. (15) MCU 对 CPU 上电。MCU 接到 WiFi 通知的唤醒消息后，就对 CPU 上电。
- 步骤 16. (16) CPU 恢复 WiFi 状态。CPU 向 WiFi 模组发送一系列指令，让 WiFi 模组退出待机模式，进入正常工作模式。
- 步骤 17. (17) 发送上线包。STA 模式需要，AP 模式不需要。

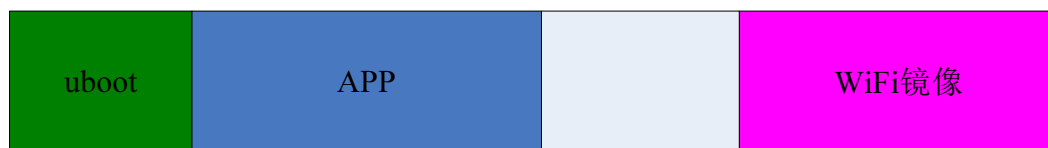
----结束

2.3.6 Wifi 镜像生成

生成 WiFi 镜像是指在 WiFi 启动完成后，将此时的 DDR 中的内容，包括 TEXT 段、DATA 段、BSS 段以及分配的堆区内容都写入到 Flash 里的指定位置的过程。生成的 WiFi 镜像是 WiFi 唤醒能够工作的基础。

生成 WiFi 镜像后，Flash 的内容如图 2-20 所示。

图2-20 生成 WiFi 镜像后的 Flash 内容分布



正常上电启动时，读取 APP 执行。

WiFi 唤醒时，读取 WiFi 镜像来恢复 WiFi 连接，然后启动。

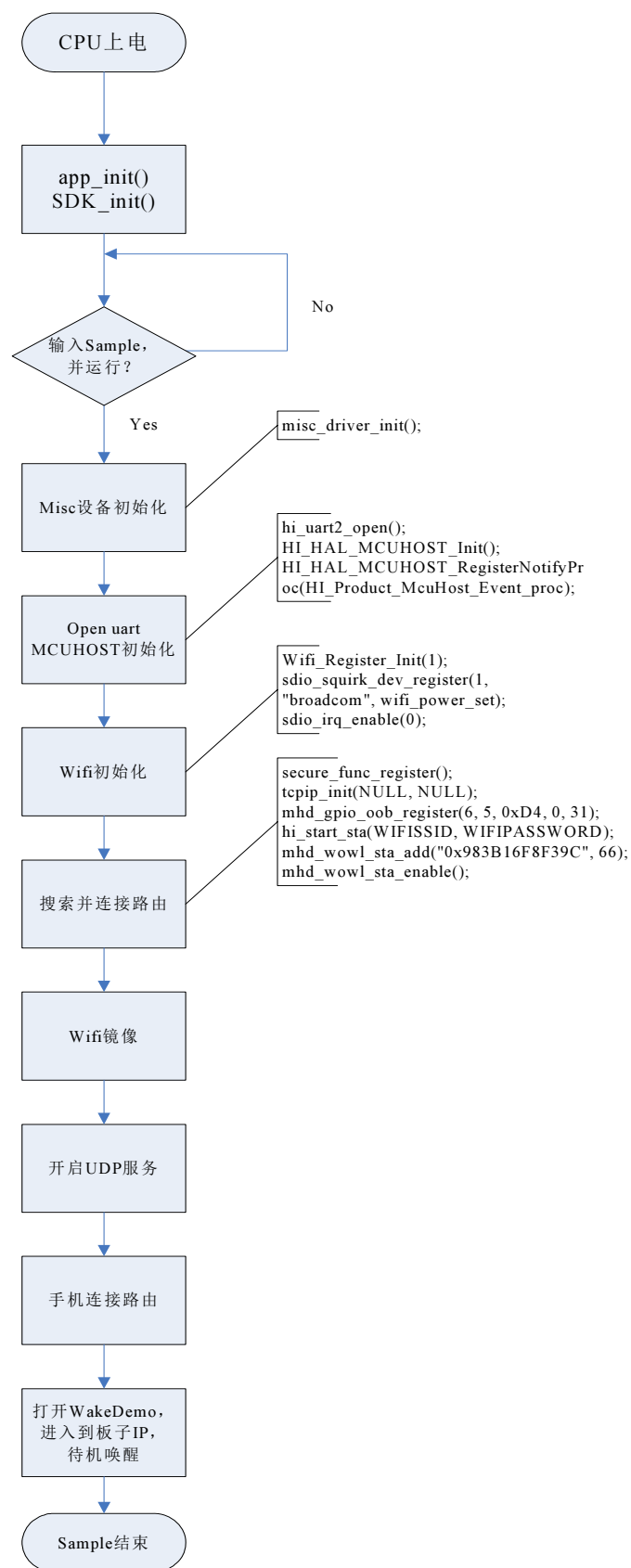
生成 WiFi 镜像是比较耗时的操作，且会擦写 Flash，在 APP 的内容没有改动时，不需要重新生成。Sample 提供一种判断机制，来决定是否要重新生成 WiFi 镜像。机制的工作过程如下：

- 用编译时获取到当前的日期和时间，与 Flash 里某个地址读取内容比较。
 - 如果比较内容一致，则说明 APP 没有修改，不需要重新生成镜像。
 - 如果比较内容不一致，则进入生成镜像的代码分支，且在生成完镜像之后，把编译日期和时间写入到指定的 Flash 位置。

2.3.7 Wifi 待机唤醒场景实现

- 软硬件说明
 - 用例可实现 STA 模式待机唤醒场景。
 - Wifi 配置名称、密码文件：wifi_info.h。
 - Wifi 模块和 SD 卡共用 SDIO0，因此 wifi 和 SD 不能同时使用。

图2-21 Wifi 待机唤醒流程





2.3.8 接口说明

2.3.8.1 驱动接口

- `int mt_wifi_start(struct netif **pnetif)`
 - 【功能】
开启 MT7601U 设备，接口内部会初始化 MT7601U，注册网络设备。
 - 【参数】
`pnetif`: 出参，用于返回向 lwip 注册的网络设备数据结构。
 - 【返回值】
0 表示成功，-1 表示失败。
- `int mt_wifi_close(void)`
 - 【功能】
关闭 MT7601U 设备，这个接口内部去初始化 MT7601U，删除网络设备。
 - 【参数】
无。
 - 【返回值】
0 表示成功，-1 表示失败。
- `int mt_msc_start(void)`
 - 【功能】
开启零配置功能，STA 模式下使用，调用前要和 AP 断开连接。函数会立即返回，结果通过 `mt_msc_get_result` 获取。
 - 【参数】
无。
 - 【返回值】
0 表示成功，-1 表示失败。
- `int mt_msc_stop(void)`
 - 【功能】
停止零配置。
 - 【参数】
无。
 - 【返回值】
0 表示成功，-1 表示失败。
- `int mt_msc_get_result(struct msc_result *result)`
 - 【功能】
获取零配置的结果，调用 `mt_msc_start` 后请循环调用，并通过参数的 `ssid` 成员非空来判断零配置成功。
 - 【参数】



result: 零配置获取到的 AP 的 SSID、认证、密码等。

【返回值】

0 表示成功, -1 表示失败。

2.3.8.2 SoftAP 模式接口

- int hostapd_start(char *ifname, struct hostapd_conf *hconf)

【功能】

开启 SoftAP, 接口内部会配置 SoftAP 参数, 启动 hostapd task。

【参数】

- ifname: 网络接口名, 可以传入 pnetif->name
- hconf: SoftAP 参数, 包括 SSID、认证、密码等, 认证模式支持 OPEN、WPA-PSK、WPA2-PSK、WPA-PSK/WPA2-PSK

【返回值】

0 表示成功, -1 表示失败。

- int hostapd_stop(void)

【功能】

关闭 SoftAP。

【参数】

无。

【返回值】

0 表示成功, -1 表示失败。

编程实例:

```
#include "hostapd_if.h"
struct netif *pwifi = NULL;
void hi_start_softap(struct netif *pnetif)
{
    UINT32 uwRet;
    struct hostapd_conf hapd_conf;
    memset(&hapd_conf, 0, sizeof(struct hostapd_conf));
    /* set WiFi driver */
    strcpy(hapd_conf.driver, "mediatek");
    /* set ssid */
    strcpy(hapd_conf.ssid, "hisiAP");
    /* set channel */
    hapd_conf.channel_num = 1;
    /* set auth */
    hapd_conf.auth_algs = 0;
    hapd_conf.wpa_key_mgmt = WPA_KEY_MGMT_NONE;
    /* start SoftAP */
    uwRet = hostapd_start(pnetif->name, &hapd_conf);
    if (!uwRet)
```




```
{
    /*config static ip for server*/
    ip_addr_t ipaddr, netmask, gw;
    IP4_ADDR(&gw, 192, 168, 1, 1);
    IP4_ADDR(&ipaddr, 192, 168, 1, 1);
    IP4_ADDR(&netmask, 255, 255, 255, 0);
    msleep(2000);
    netifapi_netif_set_addr(pwifi, &ipaddr, &netmask, &gw);

    dprintf("start dhcpd...\n");
    netifapi_dhcpd_start(pwifi);
}

void hi_stop_softap(struct netif *pnetif)
{
    netifapi_dhcpd_stop(pnetif);
    hostapd_stop();
}

void hi_start_wifi(void)
{
    if (mt_wifi_start(&pwifi) != 0) {
        printf("start wifi failed\n");
        return;
    }
    hi_start_softap(pwifi);
}
```

上述样例是 OPEN 模式的认证，其它认证模式设置如下：

1) WPA-PSK:

```
hapd_conf.wpa_key_mgmt = WPA_KEY_MGMT_PSK;
hapd_conf.wpa = 1;
hapd_conf.wpa_pairwise = WPA_CIPHER_TKIP | WPA_CIPHER_CCMP;
```

2) WPA2-PSK:

```
hapd_conf.wpa_key_mgmt = WPA_KEY_MGMT_PSK;
hapd_conf.wpa = 2;
hapd_conf.wpa_pairwise = WPA_CIPHER_TKIP | WPA_CIPHER_CCMP;
```

3) WPA-PSK/WPA2-PSK:

```
hapd_conf.wpa_key_mgmt = WPA_KEY_MGMT_PSK;
hapd_conf.wpa = 3;
hapd_conf.wpa_pairwise = WPA_CIPHER_TKIP | WPA_CIPHER_CCMP;
```



2.3.8.3 STA 模式接口

- `int wpa_supplicant_start(char *ifname, char *driver, struct wpa_supplicant_conf *conf)`
 - 【功能】
开启 STA，接口内部会配置 STA 参数，启动 wpa_supplicant task。
 - 【参数】
 - ifname: 网络接口名，可以传入 pnetif->name
 - driver: WiFi 驱动名，Mediatek 的 WiFi 传入 “mediatek”
 - conf: wpa_supplicant 配置参数，请传入 NULL
 - 【返回值】
0 表示成功，-1 表示失败。
- `int wpa_supplicant_stop (void)`
 - 【功能】
关闭 STA。
 - 【参数】
无。
 - 【返回值】
0 表示成功，-1 表示失败。
- `int wpa_register_event_cb(wpa_event_cb func)`
 - 【功能】
注册 wpa_supplicant 事件的接收回调函数，由于扫描、连接操作的结果都是异步上报的，所以需要注册回调函数接收这些事件。启动 wpa_supplicant 后需要立即调用该函数。
 - 【参数】
func: 事件回调函数，接收有新的扫描结果、连接成功、连接断开事件
 - 【返回值】
0 表示成功，-1 表示失败。
- `int wpa_cli_scan (void)`
 - 【功能】
发起扫描，函数会立即返回，扫描结束后会通过注册的回调函数发送 WPA_EVT_SCAN_RESULTS 事件给应用。
 - 【参数】
无。
 - 【返回值】
0 表示命令发送成功，-1 表示失败。
- `int wpa_cli_scan_results(struct wpa_ap_info *results, unsigned int *num)`
 - 【功能】
获取扫描结果，通过回调函数获取 WPA_EVT_SCAN_RESULTS 事件后再调用。
 - 【参数】



- results: 出参, 扫描到的 AP 列表, 一个 wpa_ap_info 结构体保存一个 AP 的信息
- num: 入参和出参, 入参表示传入的 results 数组的长度, 出参表示赋值给 results 数组的 AP 数量

【返回值】

0 表示成功, -1 表示失败。

- int wpa_cli_connect(struct wpa_assoc_request *assoc)

【功能】

连接 AP, 函数会立即返回, 连接结果会通过注册的回调函数发送 WPA_EVT_CONNECTED 或者 WPA_EVT_DISCONNECTED 事件给应用。

【参数】

assoc: 连接的 AP 的信息, 包括 SSID、认证、密码、是否隐藏 SSID

【返回值】

0 表示命令发送成功, -1 表示失败。

- int wpa_cli_wps_pbc(char *bssid)

【功能】

开始 WPS PBC 过程, 成功后可以收到 WPA_EVT_CONNECTED 事件。

【参数】

bssid: AP 的 MAC 地址, 传入 NULL 表示不限制 AP, 同时进入 WPS PBC 过程的 AP 都可以连接

【返回值】

0 表示命令发送成功, -1 表示失败。

- int wpa_cli_wps_pin(char *pin, char *bssid)

【功能】

开始 WPS PIN 过程, 成功后可以收到 WPA_EVT_CONNECTED 事件。

【参数】

- pin: WPS 的 pin 码, 与 AP 上设置的需要保持一致
- bssid: AP 的 MAC 地址, 传入 NULL 表示不限制 AP, 同时进入 WPS PIN 过程的 AP 都可以连接

【返回值】

0 表示命令发送成功, -1 表示失败。

编程实例:

```
#include "mt_wifi.h"
#include "wpa_supPLICANT.h"

struct netif *pwifi = NULL;
void wifi_event_cb(enum wpa_event event)
{
    switch(event) {
        case WPA_EVT_SCAN_RESULTS:
```



```
        printf("Scan results available\n");
        // TODO: get scan results
        break;
    case WPA_EVT_CONNECTED:
        printf("WiFi: Connected\n");
        dprintf("start dhcp...\n");
        netifapi_dhcp_start(pwifi);
        break;
    case WPA_EVT_DISCONNECTED:
        printf("WiFi: disconnect\n");
        netifapi_dhcp_stop(pwifi);
        break;
    default:
        break;
    }
}

void hi_sta_connect(void)
{
    int ret;
    struct wpa_assoc_request reqInfo;

    memset(&reqInfo, 0, sizeof(struct wpa_assoc_request));
    strcpy(reqInfo.ssid, "hisiAP");
    reqInfo.auth = WPA_SECURITY_OPEN;
    wpa_cli_connect(&reqInfo);
}

void hi_stop_sta(void)
{
    wpa_supplicant_stop();
}

void hi_start_wifi(void)
{
    if (mt_wifi_start(&pwifi) != 0) {
        printf("start wifi failed\n");
        return;
    }
    ret = wpa_supplicant_start(pwifi->name, "mediatek", NULL); /*init
wpa_supplicant*/
    if (ret)
        return;
    wpa_register_event_cb(wifi_event_cb);
}
```



```
        hi_sta_connect();  
    }
```



3 MCU 部分

3.1 MCU 烧录

MCU 负责给主控芯片供电，在调试用例之前，首先烧录 MCU 程序。MCU 的烧录方法有两种：

- 生成好的 hex 文件烧录。
- 有源码的仿真烧录。

3.1.1 MCU 文件烧录

准备工作

- 可用于烧录的 hex 文件。
- STM8 连接器（ST-Link/v2）。
- ST Visual Programmer 烧录软件。

烧录步骤

- 步骤 1. 连接 STM8（ST-Link/v2）。
- 步骤 2. 打开“ST Visual Programmer”，MCU 选择如[图 3-1](#)所示。
- 步骤 3. 打开 MCU 烧录文件。
- 步骤 4. 点击烧录，如[图 3-2](#)和[图 3-3](#)所示。
- 步骤 5. 关闭“ST Visual Programmer”软件，重新上电。

图3-1 MCU 类型选择

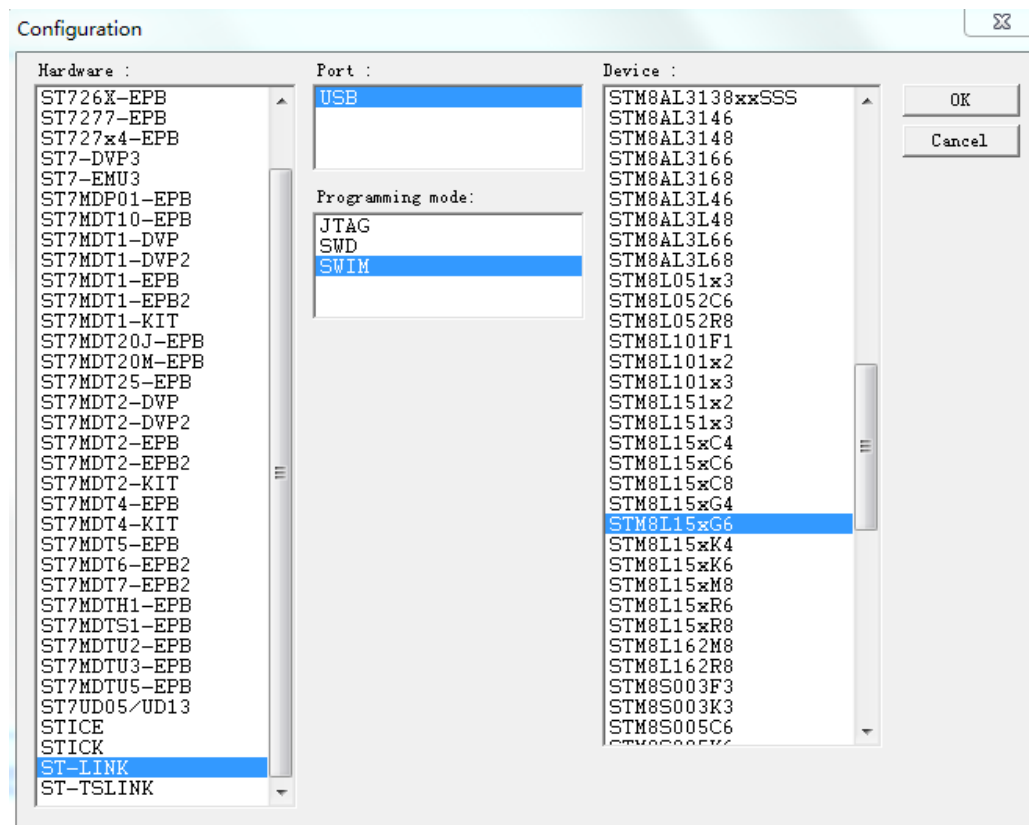


图3-2 点击烧录

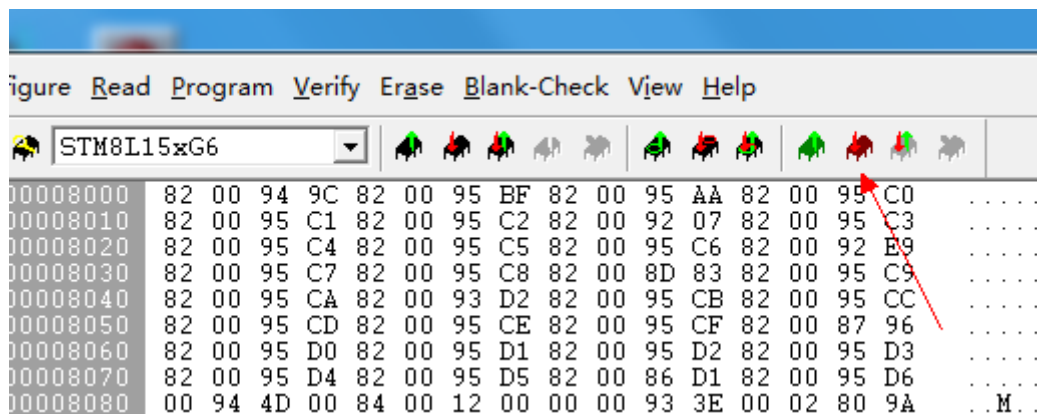


图3-3 验证成功

```
> Programming DATA MEMORY area...
Cut Version and Revision of device: 2.1
< DATA MEMORY programming completed.
> Verifying DATA MEMORY area...
Cut Version and Revision of device: 2.1
< DATA MEMORY successfully verified.
> Programming OPTION BYTE area...
Cut Version and Revision of device: 2.1
< OPTION BYTE programming completed.
> Verifying OPTION BYTE area...
Cut Version and Revision of device: 2.1
< OPTION BYTE successfully verified.
```

3.1.2 MCU 源码仿真烧录

准备工作

- MCU 源码
- STM8 连接器 (ST-Link/v2)
- IAR Embedded Workbench 仿真软件

烧录步骤

- 步骤 1. 连接 STM8 (ST-Link/v2)
- 步骤 2. 用 IAR Embedded Workbench 打开 MCU 源码
- 步骤 3. 烧录, 如图 3-4 所示。
- 步骤 4. 烧录成功跳转到调试模式, 如图 3-5 所示。

调试模式可以增加断点, 查看程序的运行情况。

图3-4 源码仿真烧录

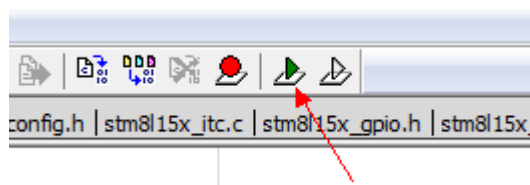
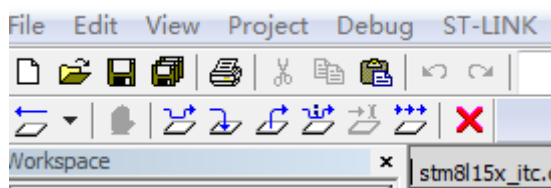


图3-5 调试模式





4 RTSP 说明

4.1 概述

RTSP 协议定义了一对多应用程序如何有效地通过 IP 网络传送多媒体数据，在体系结构上位于 RTP 和 RTCP 之上，它使用 TCP 或 RTP 完成数据传输。目前在流媒体传输技术中使用最多的就是基于 RTSP/RTP 的流媒体传输，在智能网络摄像机上也需要实现基于 RTSP/RTP 的 H.264 实时流的传输。RTSP 协议基于 TCP 完成 RTSP 请求报文和响应报文的传输，RTP 协议基于 UDP 协议完成流媒体数据的实时传输，RTCP 协议基于 UDP 协议提供客户端和服务端有关当前网络拥塞和以及实时流传输质量等信息。

4.2 RTSP 消息格式

RTSP 消息主要分两种：请求(request)和回应(response)。

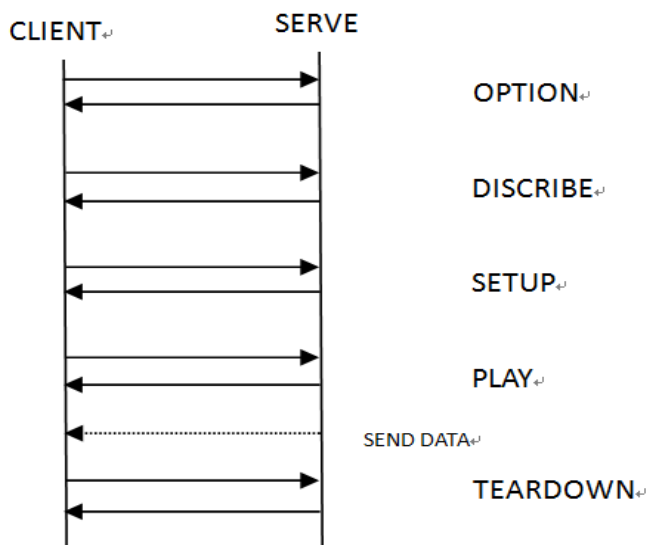
- 请求消息
方法 URI RTSP 版本 CR LF
消息头 CR LF CR LF
消息体 CR LF
- 回应消息
RTSP 版本 状态码 解释 CR LF
消息头 CR LF CR LF
消息体 CR LF

CR LF 表示回车换行；消息头需要两个 CR LF；URI 表示接受方的地址，如 192.168.3.13；RTSP 版本一般是 RTSP/1.0；状态码表示对消息回应的一个情况，具体看状态码值(RTSP response statue-code)对应的文本描述，200 表示成功。

4.3 RTSP 交互过程

RTSP 交互过程与 TCP 握手相似，主要经过 option、describe、setup、play、teardown 主要是这五步(借阅：

<http://wenku.baidu.com/view/25f2c81cfad6195f312ba600.html?from=search>)



说明

- OPTION: client 请求服务器提供可用方法；
- DISCRIBE: client 请求 server 提供会话的描述信息；
- SETUP: client 请求与 server 建立会话，确定传输模式；
- PLAY: client 向 server 发送播放请求；
- SEND DATA: server 向 client 发送流数据；
- TEARDOWN: client 向 server 发送关闭请求；

4.4 文件结构

Protocol 目录下的 rtsp 包含如下两个文件：

- Include 包含头文件。
- src 包含源文件：源文件包含 rtsp 发包协议以及 rtp 码流发送方式等一系列源码。

4.5 编译方式

将 protocol 文件放置与 osdrv 同一层目录；在 linux 编译环境进入到 rtsp 目录下，执行 Makefile 脚本即可成功编译生成 librtsp.a 的库文件。

include	2016/10/29 14:46	文件夹	
src	2016/11/11 14:46	文件夹	
librtsp.a	2016/11/11 14:46	A 文件	121 KB
Makefile	2016/11/11 14:46	文件	1 KB



4.6 主要 API 接口

- [rtsp_create](#): 创建 rtsp 通道
- [rtsp_finish](#): 结束 rtsp
- [rtp_send_h264](#): 发送.h264 码流数据

`rtsp_create(unsigned char max_con, int priority, Thr_Exit_Fun fun)`

说明:

Max_con	最大链接数
Priority	优先级
fun	回调函数

描述:

- `Rtsp_create` 函数中主要进行一些准备工作, 创建线程池、链接池、发送 buff 缓存池;
- `Max_con` 决定了最大链接个数, 即链接池和缓存池的个数;
- `Fun` 可自己定义输入, 当前的作用是在主线程退出时通知上层引用者;
- `Rtsp_create` 创建成功返回 `rtsp_handle` 句柄; 失败返回 `NULL`

`rtsp_finish(rtsp_handle rtsp_h)`

说明:

Rtsp_h	由 <code>rtsp_create</code> 创建成功返回的句柄
--------	--------------------------------------

描述:

`Rtsp_finish` 退出将释放所有创建的线程池、连接池以及缓存池等资源, 释放所有链表, 无返回值

`rtp_send_h264(rtsp_handle rtsp_h, signed char *buf, size_t len, struct timeval *p_tv)`

说明:

Rtsp_h	由 <code>rtsp_create</code> 创建成功返回的句柄
buf	发送 buf 流串
len	Buf 流串的长度
P_tv	时间戳, 结构体类型

描述:

- Rtp_send_h264 提供发送视频流接口;
- P_tv 时间戳采用的.h264 本身所带时间计算获取, 结构体如下:

```
{  
    long tv_sec;    /**< second */  
    long tv_usec;  /**< microseconds */  
}
```

以上三个函数是调用 LIBRTSP.A 库文件的三个 API 接口, 具体实现可查看 src 目录下的源码。

4.7 APK

- 为方便 rtsp 协议测试, 目录下附带有 APK 源码和安装包。



- vlc_for_rtsp.apk 是 android 版本; 手机安装之后如图:



- 在输入框输入 IP, 如: 172.16.32.180;
- 点击“链接”, 视频会在下面灰色区域部分显示;
- 针对网络上对视频流延时的修改方法, 提供的源码在 medialist.java 文件的 getmediaoptions 函数当中:

```
options.add(":file-caching=700");//100 文件缓存
```

```
options.add(":network-caching=700");//200 网络缓存
```



```
options.add(":codec=mediacodec,iomx,all");//硬解码方式
```

可对以上几个数值进行调整，其中网络缓存值对延时的时长相对影响较大，可根据实际调试效果进行调整。

4.8 RTSP response statue-code

"100"	Continue
"200"	OK
"201"	Created
"250"	Low on Storage Space
"300"	Multiple Choices
"301"	Moved Permanently
"302"	Moved Temporarily
"303"	See Other
"304"	Not Modified
"305"	Use Proxy
"400"	Bad Request
"401"	Unauthorized
"402"	Payment Required
"403"	Forbidden
"404"	Not Found
"405"	Method Not Allowed
"406"	Not Acceptable
"407"	Proxy Authentication Required
"408"	Request Time-out
"410"	Gone
"411"	Length Required
"412"	Precondition Failed
"413"	Request Entity Too Large
"414"	Request-URI Too Large
"415"	Unsupported Media Type
"451"	Parameter Not Understood
"452"	Conference Not Found
"453"	Not Enough Bandwidth



| "454" Session Not Found
| "455" Method Not Valid in This State
| "456" Header Field Not Valid for Resource
| "457" Invalid Range
| "458" Parameter Is Read-Only
| "459" Aggregate operation not allowed
| "460" Only aggregate operation allowed
| "461" Unsupported transport
| "462" Destination unreachable
| "500" Internal Server Error
| "501" Not Implemented
| "502" Bad Gateway
| "503" Service Unavailable
| "504" Gateway Time-out
| "505" RTSP Version not supported
| "551" Option not supported
| extension-code
extension-code = 3DIGIT
Reason-Phrase = *<TEXT, excluding CR, LF>



5 UBOOT 烧写方式

5.1 概述

Uboot 烧写方式使用的是 python 脚本，在使用该脚本烧写之前，须安装 python 的运行环境，请查阅《python 安装步骤.docx》，或者可根据自己电脑本身的配置在网络上下载安装。

5.2 烧写步骤

请按以下步骤进行烧写 uboot:

步骤 1. 将需要烧写的文件与 burn-boot.py 脚本同一目录。

burn-boot.py	2016/10/24 17:15	Python File	14 KB
README.md	2016/10/24 17:01	MD 文件	1 KB
u_boot_hi3516c.bin	2016/7/19 15:15	BIN 文件	188 KB

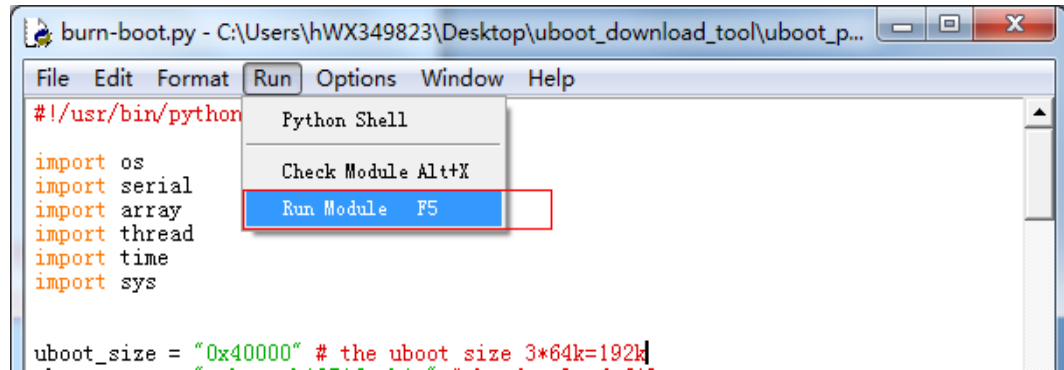
步骤 2. 使用 IDLE 编辑方式打开 burn-boot.py，修改文件名和大小。

```
uboot_size = "0x40000" # the uboot size 3*64k=192k
uboot_name = "u_boot_hi3516c.bin" #the download-filename
```

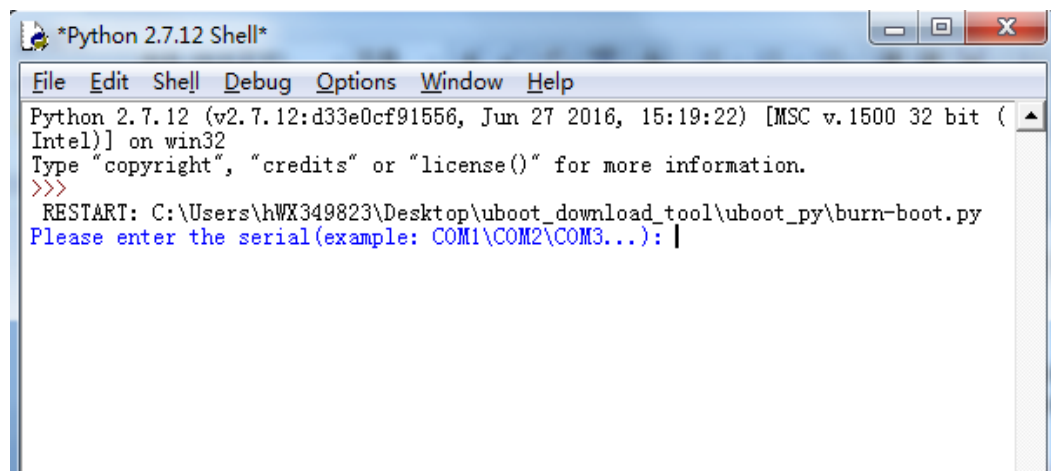
步骤 3. 修改之后，进行烧写

方法一：

- 可直接选择在 IDLE 编辑界面 RUN 菜单下的 Run Module 或者直接按下快捷键 F5，即可运行脚本。
- 按提示输入 PC 端与板端连接的端口号，如 COM1。
- 设备重新上电，uboot 开始进行下载。



- 脚本运行..提示输入串口号..



- 输入串口后，提示设备重新上电。

```
RESTART: C:\Users\hWX349823\Desktop\uboot_download_tool\uboot_py\burn-boot.py
Please enter the serial(example: COM1\COM2\COM3...): COM1
SerialPort has been connencted, Please power off, then power on the device.
If it doesn't work, please try to repower on.
```

- 设备上电之后，开始烧写，直到烧写结束。

方法二：(必须保证 python 脚本放在不包含中文字符的路径下)

- 打开 windows 的 cmd 界面,进入到 python 脚本目录
cd C:\Users\Desktop\uboot_download_tool\uboot_py
- 在输入 python burn-boot.py 回车。
- 按提示输入 PC 端与板端连接的端口号，如 COM1 回车。
- 设备重新上电，uboot 开始下载。



```
C:\Users\hWX349823\Desktop\uboot_download_tool\uboot_py>python burn-boot.py
Please enter the serial(example: COM1\COM2\COM3...): COM1
SerialPort has been connencted, Please power off, then power on the device.
If it doesn't work, please try to repower on.
#####10%
#####20%
#####30%
#####40%
#####50%
#####60%
#####70%
#####80%
#####90%
```

----结束

5.3 注意事项

Python 脚本的一些特殊性，为能正常烧写 uboot，请注意以下事项：

- 此 python 脚本仅支持 hi3516cv200 芯片。
- 必须保证 python 脚本放在不包含中文字符的路径下。
- 脚本在 15 秒内未收到板端信息，提示烧写失败。

```
****
RESTART: C:\Users\hWX349823\Desktop\uboot_download_tool\uboot_py\burn-boot.py
Please enter the serial(example: COM1\COM2\COM3...): COM1
SerialPort has been connencted, Please power off, then power on the device.
If it doesn't work, please try to repower on.
over time! recev anything in 15s
```

- 在修改烧写大小时，为确保擦写 flash 正常，size 大小最好以 0x10000(64K)为最小单位。
- 在选择编辑方式上，最好选择 python 安装自带的 IDLE 编辑器，因 python 脚本对格式要求非常严格，若采用其余打开方式编辑，或许会引起格式不对导致脚本异常出错。