

# 多屏异显应用开发文档

## 1. 功能介绍

video pbp功能: picture by picture, 多路解码, 双屏显示

画中画(PIP): 1个Display Engine+2个Video Engine。两路解码, 一路DE输出, 输出到同一个DE(4K或2K DE)。一路解码显示到主图层, 一路解码显示到辅图层

双屏异显: 2个Display Engine+2个Video Engine。两路解码, 两路DE输出((4K DE和2K DE)), 可以输出到主图层或辅图层。

## 2. 异显的设备组合

设备	备注
LVDS+LVDS	两个LVDS屏 CLOCK 要能一样
LVDS+MIPI	如果MIPI 是竖屏, MIPI 屏最好连到4KDE(通过DTS 配置)
RGB+MIPI	如果MIPI 是竖屏, MIPI 屏最好连到4KDE(通过DTS 配置)
HDMI+CVBS	HDMI 从4K DE 输出, CVBS 从2K DE 输出
HDMI+VGA	HDMI 从4K DE 输出, VGA 从2K DE 输出

理论上可以支持多路解码 (Video Engine) , 但限于性能问题, 目前只支持两路解码

解码数据源 (通道) :

- media player
- 同屏
- HDMI in
- CVBS in

**双路解码输出组合**, 第一行表示1路输出, 第一列表示另外1路数据, O 表示可以同时两路解码, X 表示不支持这种组合

	media player	同屏	HDMI in	CVBS in
media player	O	O	O	O
同屏	O	X	O	O
HDMI in	O	O	X	X
CVBS in	O	O	X	X

### 3. demo代码

- 我们使用控制台串口命令验证，测试多屏异显示。没有UI显示，这样可以灵活验证多种输出场景组合，客户可以参考对应sample code进行移植

后面控制台使用到的串口测试命令，相关代码在SDK中找到：

- 媒体播放器  
hcrtos\components\hc-examples\source\ffplayer\_examples.c
- 同屏器(无线/有线)  
hcrtos\components\hc-examples\source\hccast\_test\
- HDMI in  
hcrtos\components\cmds\source\hdmi\_rx\
- CVBS in  
hcrtos\components\cmds\source\tv\_decoder\
- 显示设置  
hcrtos\components\hc-examples\source\dis\_test.c

SDK的板级默认config中，已经有支持多路解码的configs:

```
hichip_hc16xx_db_a3300_v10_multi_display_bl_defconfig
hichip_hc16xx_db_a3300_v10_multi_display_defconfig
hichip_hc16xx_db_a3300_v10_projector_cast_multi_dec_bl_defconfig
hichip_hc16xx_db_a3300_v10_projector_cast_multi_dec_defconfig
hichip_hc16xx_db_a3300_v10_projector_cast_multi_dis_bl_defconfig
hichip_hc16xx_db_a3300_v10_projector_cast_multi_dis_defconfig
hichip_hc16xx_db_a5100_v31_multi_decode_bl_defconfig
hichip_hc16xx_db_a5100_v31_multi_decode_defconfig
hichip_hc16xx_db_d3000_v10_projector_cast_multi_dis_bl_defconfig
hichip_hc16xx_db_d3000_v10_projector_cast_multi_dis_defconfig
....
```

xxxx\_multi\_display\_xxx: 双路解码，双路输出

xxxx\_multi\_decode\_xxx: 双路解码，单路输出，画中画

这些demo defconfig编译出来的测试固件，是demo DE多路解码显示，无UI显示。

- 默认开机bootloader在2K DE显示开机logo(可以改为在4K DE显示开机logo)

- 进入控制台后不会有UI应用，2K/4K DE都没有任何UI
- 手动数据命令播放多媒体，同屏，hdmi rx等
- demo的是DE的解码显示，解码过程也无任何UI

使用demo代码测试，需要打开必要的menuconfig（上面的xxxx\_multi\_display\_xxx, xxxx\_multi\_decode\_xxx defconfig已经使能）

**使能测试代码sample code:**

**make menuconfig**

**hdmi in, cvbs in, wifi test sample code**

```
./output/.config - HCRITOS SDK Menu Configure
> Components > Cmts ----- Cmts
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu ----). Highlighted letters are
feature. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] feature is selected [ ] feature

-----^(-)-----
[ ] spi operations
[ ] persistent memory operations
[ ] pok test operations
[ ] uart test operations
[ ] watchdog test operations
[ ] Tfuse bits dump operations
[*] hdmi rx test operations --->
[*] tv decoder(cvbs in) test operations
[ ] IQ test operations
[ ] Display debugging
[ ] HCFOTA test operations
[ ] standby test operations
[ ] config bluetooth test
[ ] config rtwpriv
[ ] config iperf3
[ ] ws2811 test operations
[ ] sorting test operations
[ ] USB setting operations ----
[ ] vin dvp test operations
[ ] pwm test
[*] wifi test
[*] wifi mac addr updating test
[ ] Network test commands
```

**网络工具:**

默认应该已经enable

```
> Components > kernel > Net > Enable network ----- Enable network
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu ----). Highlighted le
<N> excludes a feature. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] feature

--- Enable network
[*] Enable Lwipsack
    Lwipsack (Enable Lwipsack (2.1.2)) --->
[ ] ppp support
[ ] IPv6 support
```

net的工具代码放在lwip shell命令中

```
components/kernel/source/net/lwip-2.1.2/enhancement/src/api_shell.c:81:CONSOLE_CMD(net, NULL, NULL, CONSOLE_CM
components/kernel/source/net/lwip-2.1.2/enhancement/src/api_shell.c:3504:CONSOLE_CMD(netstat, "net", (console_
t")
components/kernel/source/net/lwip-2.1.2/enhancement/src/api_shell.c:3892:CONSOLE_CMD(netdebug, "net", (console
ebug")
```

**同屏测试代码**

```
Components
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu ----). Highlighted letters are hotkeys. P
feature. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] feature is selected [ ] feature is excluded

() Prebuilt subdir
[*] prebuilts --->
Applications Configuration --->
[ ] Stack Smashing Protector(ssp)
-* Newlib --->
[*] kernel --->
[*] Cms --->
-* pthread
-* ffmpeg --->
[*] hc-examples
[ ] hmi wireless example
[*] video picture by picture mode test examples
[*] hcast test examples
[*] hcast wifi cast test examples
[*] hcast usb test examples
[ ] Opencore-amr library
[*] libusb
[ ] build libusb examples
[ ] libuvc
[*] hcfota
[*] liblvg1 --->
[*] Bluetooth --->
[ ] quicklz
v(+)
```

## 4. 双画面输出的数据类型定义

```
dis.h:
typedef enum DIS_TYPE {
    DIS_TYPE_SD = 0,
    DIS_TYPE_HD,    // 使用2K DE
    DIS_TYPE_UHD,  // 使用4K DE
} dis_type_e;

typedef enum DIS_LAYER {
    DIS_LAYER_MAIN = 0x1,
    DIS_LAYER_AUXP = 0x10,
    DIS_LAYER_GMAS,
    DIS_LAYER_GMAF,
} dis_layer_e;

vidmp.h:
typedef enum VIDEO_PBP_MODE {
    VIDEO_PBP_OFF,    // 单画面输出
    VIDEO_PBP_2P_ON,  // 双画面输出
} video_pbp_mode_e;

ffplayer.h:
typedef struct HCPlayerInitArgs
{
    ....
    video_pbp_mode_e  pbp_mode; //VIDEO_PBP_OFF: 单画面输出; VIDEO_PBP_2P_ON: 双画
面输出
    dis_type_e dis_type; //DIS_TYPE_HD: 输出到2K DE; DIS_TYPE_UHD: 输出到4K DE
    dis_layer_e dis_layer; //DIS_LAYER_MAIN: 输出到主图层; DIS_LAYER_AUXP: 输出到辅
图层
```

```
};

vidmp.h:
struct video_config {
    .....
    uint32_t      pbp_mode; //VIDEO_PBP_OFF: 单画面输出; VIDEO_PBP_2P_ON: 双画面输出
    int           dis_type; //DIS_TYPE_HD: 输出到2K DE; DIS_TYPE_UHD: 输出到4K DE
    int           dis_layer; //!< dis_layer_e
};
```

## 5. ffmpegplayer双画面输出相关配置

媒体播放器，DLNA同屏应用通过ffmpegplayer解码播放，可以配置输出的DE和图层

```
HcPlayerInitArgs init_args = {0};

init_args.preview_enable = 0;//0:全屏输出; 1: 缩放输出

init_args.pbp_mode = VIDEO_PBP_2P_ON; //VIDEO_PBP_OFF: 单画面输出;
VIDEO_PBP_2P_ON: 双画面输出
init_args.dis_type = DIS_TYPE_UHD; //DIS_TYPE_UHD: logo显示在4K DE; DIS_TYPE_HD:
logo显示在2K DE
init_args.dis_layer = DIS_LAYER_MAIN; //DIS_LAYER_MAIN: 输出到主图层;
DIS_LAYER_AUX: 输出到辅图层

void *player = hcplayer_create(&init_args);
```

## 6. Video Engine双画面输出配置

应用可以直接通过Video Engine解码，应用场景如解码logo，镜像同屏等。可以配置输出的DE和图层

```
struct video_config mvcfg;

mvcfg.pbp_mode = VIDEO_PBP_2P_ON; //VIDEO_PBP_OFF: 单画面输出; VIDEO_PBP_2P_ON: 双画面输出
mvcfg.dis_layer = DIS_LAYER_MAIN; //DIS_LAYER_MAIN: 输出到主图层; DIS_LAYER_AUX: 输出到辅图层
mvcfg.dis_type = DIS_TYPE_UHD; //DIS_TYPE_UHD: logo显示在4K DE; DIS_TYPE_HD: logo显示在2K DE

int g_logo_vfd = open("/dev/viddec", O_RDWR);

if (write(g_logo_vfd, (uint8_t *)&pkthd, sizeof(AvPktHd)) != sizeof(AvPktHd))
    .....
```

## 7. HDMI in 双画面输出配置

---

```
struct hdmi_rx_display_info dis_info = {0};
int rx_fd = open("/dev/hdmi_rx" , O_RDWR);

dis_info.dis_type = DIS_TYPE_UHD; //DIS_TYPE_UHD: 显示在4K DE; DIS_TYPE_HD: 显示在2K DE
dis_info.dis_layer = DIS_LAYER_MAIN; //DIS_LAYER_MAIN: 显示在主图层;
DIS_LAYER_AUX: 显示在辅图层
ioctl(rx_fd , HDMI_RX_SET_DISPLAY_INFO , &dis_info);

video_pbp_mode_e pbp_mode = VIDEO_PBP_2P_ON; //VIDEO_PBP_OFF: 单画面输出;
VIDEO_PBP_2P_ON: 双画面输出
ioctl(rx_fd , HDMI_RX_SET_PBP_MODE , pbp_mode);
```

## 8. CVBS in双画面输出配置

---

```
struct tvdec_display_info dis_info = { 0 };
int tv_dec_fd = open("/dev/tv_decoder" , O_WRONLY);

dis_info.dis_type = DIS_TYPE_HD;
dis_info.dis_layer = DIS_LAYER_MAIN;
ioctl(tv_dec_fd , TVDEC_SET_DISPLAY_INFO , &dis_info);

video_pbp_mode_e pbp_mode = VIDEO_PBP_2P_ON; //VIDEO_PBP_OFF: 单画面输出;
VIDEO_PBP_2P_ON: 双画面输出
ioctl(tv_dec_fd , TVDEC_SET_PBP_MODE , pbp_mode);
```

## 9. 视频缩放配置

---

```
int fd = open("/dev/dis", O_WRONLY);

struct dis_zoom dz = { 0 };
dz.src_area.x = s_x;
...
dz.dst_area.x = d_x;
...

dz.layer = DIS_LAYER_MAIN; //DIS_LAYER_MAIN:主图层缩放; DIS_LAYER_AUXP: 辅图层
缩放
dz.distype = DIS_TYPE_HD; //DIS_TYPE_UHD: 4K DE缩放; DIS_TYPE_HD: 2K DE缩放

ioctl(fd, DIS_SET_ZOOM, &dz);
```

## 10. 视频宽高比设置

```
int fd = open("/dev/dis", O_RDWR);
if ( fd < 0)
    return;
dis_aspect_mode_t aspect = {0};

aspect.distype = DIS_TYPE_UHD; //DIS_TYPE_UHD: 4K DE; DIS_TYPE_HD: 2K DE
aspect.layer = DIS_LAYER_MAIN; //DIS_LAYER_MAIN:主图层; DIS_LAYER_AUXP: 辅图层
缩放DE

aspect.tv_mode = DIS_TV_16_9; //16:9
aspect.dis_mode = DIS_PILLBOX;
aspect.active_mode = DIS_SCALE_ACTIVE_IMMEDIATELY;
ioctl(fd, DIS_SET_ASPECT_MODE, &aspect);
close(fd);
```

## 11. 屏幕信息

获取屏幕信息时，在双画面情况下，需要选择获取哪个DE的屏幕信息

```
int api_get_screen_info(dis_screen_info_t * dis_area)
{
    int fd;
    fd = open("/dev/dis", O_WRONLY);
    if(fd < 0){
        return -1 ;
    }
}
```

```

        dis_area->distype=DIS_TYPE_UHD; //DIS_TYPE_UHD:4K DE screen信息;
DIS_TYPE_HD:2K DE screen信息;
        ioctl(fd , DIS_GET_SCREEN_INFO, dis_area);
        close(fd);
        return 0;
    }

```

开关视频输出

```

int fd = -1;
struct dis_win_onoff winon = { 0 };
fd = open("/dev/dis" , O_WRONLY);
if(fd < 0)
{
    return -1;
}
winon.distype = DIS_TYPE_HD; //
winon.layer = DIS_LAYER_MAIN;
winon.on = on_off ? 1 : 0;
ioctl(fd , DIS_SET_WIN_ONOFF , &winon);
close(fd);

```

涉及到`open("/dev/dis" , O_WRONLY)`; DisplayEngine的操作，在使能双输出功能时，都需要注意要选择输出的DE。

## 12. 设备树双画面配置

我们默认SDK中已经有相应的配置双路解码的defconfig，

其中，

xxx-multi-display-xxx: 双路解码双路输出

xxx-multi-decode-xxx: 双路解码单路输出

DTS需要增加双画面相关配置，以hc16xx-db-d3000-v10-hcprojector.dts为例。详细看

```

#define CONFIG_PBP_MODE_SUPPORT (1)                //多解码支持。 PIP画中画单屏输出，打开这个
就可以

//下面是双屏输出需要配置
...
#define CONFIG_DE4K_OUTPUT_SUPPORT 1                //需要第二个4K DE输出
#if (CONFIG_DE4K_OUTPUT_SUPPORT == 1)
#define CONFIG_DUAL_DE_SUPPORT 1                    //异显输出（双DE）支持，后面dts需要配置两
个DE节点属性，否则不能启动；PIP画中画使用1个DE,可以设置为0
#endif

//配置4K DE， 可以和2K DE 配置不同的屏幕（包括点屏）

```



```

#if (CONFIG_DUAL_DE_SUPPORT == 1)
    DE4k: de4k-engine {
        /*

```

点屏双DE输出时，可以支持 mipi + lvds，lvds + lvds。需要修改一下屏参配置的dtsi文件

## 12.1 mipi + lvds

mipi\_xxxx.dtsi用作4k DE输出

```

&mipi {
    reg = <0xb884A000 0x300>, <0xb8800444 0x8>, <0xb8800080 0x8>;
    /*
    * 0: E_SRC_SEL_FXDE = 0
    * 1: E_SRC_SEL_4KDE
    * 2: E_SRC_SEL_HDMI_RX
    */
    src_sel = <CONFIG_DE4K_OUTPUT_SUPPORT>; //DE 选择为4K DE
};

//配置4K节点
&DE4k {
    tvtype = <15>;

```

lvds\_xxxx.dtsi用作2k DE输出

```

&lvds{
    reg = <0x18860000 0x200>, <0x18800000 0x500>;

    /*
    * 0: E_SRC_SEL_FXDE = 0
    * 1: E_SRC_SEL_4KDE
    * 2: E_SRC_SEL_HDMI_RX
    */
    src_sel = <0>; //DE 选择2K DE
    src1_sel = <0>; //DE 选择，LVDS一个插槽可以连两路LVDS 屏，但只有1个lvds dts节点。因此第二个lvds屏
    //要用这个参数选择DE。只有1个LVD屏， src1_sel和src_sel选择为0
    // status = "okay";
};

```

## 12.2 lvds + lvds

lvds + lvds是两个屏幕的插线并列插入板子一个插槽中。

注意：lvds + lvds使用一个lvds\_xxxx.dtsi配置文件，里面配置两个DE节点屏参。两个屏参参数应该接近，否则容易干扰

lcd的dtsi配置也有点不同

lvds\_xxxx.dtsi

```
&lvds{
.....
/*
 * 0: E_SRC_SEL_FXDE = 0
 * 1: E_SRC_SEL_4KDE
 * 2: E_SRC_SEL_HDMI_RX
 */
src_sel = <0>;           //DE 选择
src1_sel = <CONFIG_DE4K_OUTPUT_SUPPORT>;           //DE 选择，LVDS一个插槽可以
连两路LVDS 屏，但只有1个lvds dts节点。因此第二个lvds屏
                                                    //要用这个参数选择DE

    // status = "okay";
};

//2k DE的节点参数
&DE {
    tvtype = <15>;

//4k DE的节点参数
&DE4k {
    tvtype = <15>;
.....
}
```

## 13. bootloader双画面

bootloader需要显示开机logo/动画时，可以配置输出到2K/4K DisplayEngine

## 14. OSD/UI显示

OSD/UI是混合到DisplayEngine的，OSD/UI要输出在哪个DE，是DTS配置的

```
583 #if (CONFIG_DE4K_OUTPUT_SUPPORT == 1)
584 #define CONFIG_FB0_REG 0xb883a000    //输出到4K DE
585 #define CONFIG_FB1_REG 0xb883a080    //输出到4K DE
586 #else
```

```

587 #define CONFIG_FB0_REG 0xb8808000 //输出到2K DE
588 #define CONFIG_FB1_REG 0xb8808080 //输出到2K DE
589 #endif
590
591     fb0 {
592         bits_per_pixel = <32>;
593         xres = <1280>;
594         yres = <720>;
595         xres_virtual = <1280>;
596         yres_virtual = <720>;
597         xoffset = <0>;
598         yoffset = <0>;
599
600         scale = <1280 720 1920 1080>;
601
602         reg = <CONFIG_FB0_REG 0x1000>; //OSD/UI显示在4K DE的fb0/fb1或者
2K DE的fb0/fb1
603         /*

```

## 15. 测试方法和流程

海奇SDK中有测试多解码的sample code，在串口终端中数据命令进行测试

rtos和linux由于系统各不同，进入方式有些差异，但是进入测试模块后，具体的测试子命令参数流程是一致的，兼容的。

**注意：** 双路解码，单路输出，画中画功能测试项目，为了兼容以前应用没有配置PBP的功能，设置选择DE参数时，都是选择2K DE（但输出，选择2K DE为第一个输出）选择4K DE，没有输出显示！！！！

### 15.1 RTOS测试

要支持测试多个数据源测试，需要在config中打开对应配置：

```

> Components > hc-examples -> [*]    video picture by picture mode test examples
> Components > hc-examples -> [*]    bluetooth examples (D3000_V20 蓝牙板子)
> Components > hc-examples -> [*]    HDMI rx examples
> Components > hc-examples -> [*]    TV dec(CVBS rx) examples
> Components > hc-examples -> [*]    hccast test examples

```

开机后，先关闭所有视频图层，串口终端执行

```
pbp init
```

测试无线同屏时，我们用标准网络工具手动在串口终端联网，先必须保证平台处于station模式，并连接成功网络

## D3000\_V20

如果使用板子 D3000\_V20测试，板子的micro usb的电压（wifi dongle）是蓝牙控制的。开机进入控制台需要用蓝牙打开micro usb电压供电，否则micro usb的wifi dongle不能工作。同时，需要把蓝牙的音频通路选择为SPDIF in才能播放声音

dts的蓝牙节点：

```
bluetooth {
.....
    pinmux-lineout-det = <7 0>;        // <pinpad pinmux>
    pinmux-lcd-backlight = <13 2>;     // <pinpad pinmux>
    wifien-gpios = <22 1>;             // <pinpad pinpad_value>
    devpath = "/dev/uart0";
    status = "okay";
};
```

bluetooth audio\_channel 0

串口终端用命令拉高gpio（节点属性：wifien-gpios = <22 1>），执行如下蓝牙指令就可以初始化工作

```
bluetooth init
```

蓝牙还控制音频的传输：

bluetooth audio\_channel 0， 蓝牙模块选择接收line in输入，如CVBS-in的音频

bluetooth audio\_channel 1， 蓝牙模块选择接收SPDIF IN输入，如HDMI-in多媒体播放，同屏播放的音频

## 15.1.1 联网

### 15.1.1.1 station模式

```
# net ifconfig
p2p0 ip:0.0.0.0 netmask:0.0.0.0 gateway:0.0.0.0
    Hwaddr 62:1d:9d:8b:45:6e MTU:1500 Stop Link Down
wlan0 ip:0.0.0.0 netmask:0.0.0.0 gateway:0.0.0.0
    Hwaddr 60:1d:9d:8b:45:6e MTU:1500 Stop Link Down
lo ip:127.0.0.1 netmask:255.0.0.0 gateway:127.0.0.1
    Hwaddr 00 MTU:0 Running Link UP
```

2. 连接路由

联网:

station连接路由, 下面命令拷贝, 直接串口运行。如连接wifi: TP-LINK\_A422, 密码: hichip1234。

下面命令可以直接在串口终端输入窗口一次输入, 单注意保持单行命令不要被换车换行截断!

```
=====
=====
net ifconfig wlan0 up
net ifconfig p2p0 up
net dhclient wlan0
wifi wpa_supplicant -i wlan0 -Dwext -x 9877 -x 9881 -c /etc/wpa_supplicant.conf -
C /var/run/wpa_supplicant -N -i p2p0 -x 9890 -x 9883 -Dwext -c
/etc/wpa_supplicant.conf -C /var/run/wpa_supplicant
sleep 1000
wifi wpa_cli -i wlan0 -x 9877 add_network
wifi wpa_cli -i wlan0 -x 9877 set_network 0 ssid "\"TP-LINK_A422\""
wifi wpa_cli -i wlan0 -x 9877 set_network 0 psk "\"hichip1234\""
wifi wpa_cli -i wlan0 -x 9877 select_network 0
=====
=====
```

查询状态

```
=====
=====
wifi wpa_cli -i wlan0 -x 9877 status
bssid=a4:1a:3a:57:a4:22
freq=2462
ssid=TP-LINK_A422
id=0
mode=station
pairwise_cipher=CCMP
group_cipher=CCMP
key_mgmt=WPA2-PSK
wpa_state=COMPLETED
ip_address=192.168.1.138 => 连接成功,
p2p_device_address=60:1d:9d:8b:45:6e
address=60:1d:9d:8b:45:6e
uuid=a89d84df-c6b4-56ca-9add-94e1eda771b3
=====
=====
```

其他命令, 可供参考查询

获取已经联网wifi的ssid

```
# wifi wpa_cli -i wlan0 -x 9877 get_network 0 ssid
"TP-LINK_A422"
```

获取已经联网wifi的密码

```
# wifi wpa_cli -i wlan0 -x 9877 get_network 0 psk
"hichip1234"
```

如果wifi一直连接不上, 可以终止, 移除网路, 然后重新add 网络连接

```
# wifi wpa_cli -i wlan0 -x 9877 remove_network 0
```

wpa\_supplicant出错, 一般wpa服务进程会自动退出, 如果要强行退出, 也可以用命令

```
# wifi wpa_cli -i wlan0 -x 9877 terminate
Not owner
wpa_supplicant_thread:exit
```

注意：demo固定使用同屏设备名：my\_hccast\_test\_xxxx。可以在demo代码中修改：

```
hertos\components\hc-examples\source\hccast_test\hccast_test.h:
#define HCCAST_TEST_CAST_NAME    "my_hccast_test"
```

### 15.1.1.2 Host模式(不做测试)

```
# net
(net)# ifconfig wlan0 netmask 255.255.255.0
(net)# ifconfig wlan0 192.168.5.1 gateway 192.168.5.1
(net)# udhcpd start wlan0 192.168.5.2 10

#wifi
(net)# wpa_supplicant -H -i p2p0 -x 9890 -X 9883 -Dwext -c
/etc/wpa_supplicant.conf -C /var/run/wpa_supplicant
Create thread success.
hc1600a@dba5100v31(wifi)# [rtl8188fu__start_bss_network][1788]ch_to_set==6
[rtw_set_txq_params][7949]
[rtw_set_txq_params][7949]
[rtw_set_txq_params][7949]
[rtw_set_txq_params][7949]
[rtw_set_mac_acl][7912]
ioctl[SIOCSIWAP]: Not owner

从miracast退出后，可能需要重新设置hostap，才能发现hostap
#wifi
(wifi)# hostapd_cli -iwlan0 reload
如果上述命令不行，执行如下命令
(wifi)# hostapd_cli -iwlan0 disable
(wifi)# hostapd_cli -iwlan0 enable
```

这时板子创建成功名为SSID\_TO\_BE\_SET-234的热点，无密码

linux

```
/etc/./wifiprobe.sh
sleep 5
wpa_cli -i wlan0 add_network
wpa_cli -i wlan0 set_network 0 ssid '"TP-LINK_A422"'
wpa_cli -i wlan0 set_network 0 psk '"hichip1234"'
wpa_cli -i wlan0 enable_network 0
wpa_cli -i wlan0 select_network 0
udhcpc -i wlan0 -q
```

## 15.1.2 无线同屏

投影名称: my-hccast-test-xxxx

\* 启动参数

-d 0: 2K DE; 1: 4K DE

-l 0: 主图层; 1: 辅助层

-r rotate 0: 0度; 1: 90度; 2: 180度; 3: 270度。 ROTATE\_TYPE\_180...

-m mirror 0: 无翻转; 1: 左右镜像翻转

-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;

-o 设置音频路径: 1//HUDI\_AUDSINK\_TYPE\_I2SO(1<<0) | HUDI\_AUDSINK\_TYPE\_SPO(1<<2);

-g 使用audio dec解码: 1, enable; 0, disable. 可以和媒体播放器混音

-h 关闭音频: 1, 关闭; 0, 打开

\* 启动同屏(可以关闭音频-h)

2K DE, 主图:

cast\_wifi start -p 1 -d 0 -l 0

2K DE, 辅图:

cast\_wifi start -p 1 -d 0 -l 1

4K DE, 主图:

cast\_wifi start -p 1 -d 1 -l 0

4K DE, 辅图:

cast\_wifi start -p 1 -d 1 -l 1

\* 投屏模式

cast\_wifi rotate -f 1

\* 缩放

cast\_wifi preview 0 0 960 540

停止, 再启动同屏才生效

\* 停止同屏

cast\_wifi stop

在miracast同屏需要独占网卡, 断开连接后, 如果要重新同屏, 要stop/start miracast

1. 要dlna, aircast同屏, 需要重新联网, 旧有同屏服务要重新启动

cast\_wifi stop

rtos: wifi wpa\_cli -i wlan0 -x 9877 select\_network 0

net dhclient wlan0

cast\_wifi start -p 1

linux:

ctrl+z, multi\_dis切换到后台

# wpa\_cli -i wlan0 select\_network 0

# udhcpc -i wlan0 -q

# fg, multi\_dis切回到前台

multi\_dis:(cast\_wifi)# start -p 1

2. 继续miracast同屏, 要先停止miracast服务, 再继续启动服务, 当然, 直接第一步也可以

cast\_wifi stop

cast\_wifi start -p 1

dlna, aircast同屏结束, 无需重新启动服务

### 15.1.3 多媒体播放

**注意：目前图片播放只能主图层输出(-l 0)**

```
* 进入mdiea player测试
# mp
```

双路测试时，可以先关闭多媒体音频，否则可能和同屏等有声音干扰问题。这个是个开关，后面播放命令根据这个开关开启/关闭声音

```
# disav -v 0 -a 1
```

\* 注意：如果需要支持播放两路媒体播放器，可以先用命令multi 进入多路媒体播放模式， single退出多路媒体播放模式

```
# multi //使能多路多媒体播放
```

\*播放gongxiami.mp4到4K DE

```
# play /media/sda1/gongxiami.mp4 -z 1 -y 1 -l 0 -u 111
-u 111为自定义的播放文件名
```

\*播放test.mp4到2K DE

```
# play /media/sda1/test.mp4 -z 1 -y 0 -l 0 -u 222
-u 222为自定义的播放文件名
```

后续使用如下指定缩放111路缩放，对应pause命令也可以类似加入111文件名暂停该路播放

```
# preview 111 0,0,1920,1080,0,0,960,540
```

调整音量：

```
# volume 100
```

\* 播放参数

-y 0: 2K DE; 1: 4K DE

-l 0: 主图层; 1: 辅图层

-r: 1, 90; 2, 180; 3, 270

-m 0: 无镜像; 1: 左右镜像翻转; 2: 上下镜像翻转

-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;

-o 设置音频路径: 1//AUDSINK\_SND\_DEVBITS\_I2S0(1<<0) | AUDSINK\_SND\_DEVBITS\_SPO(1<<2);

-s freerun: 0, HCPLAYER\_FREERUN; 1, HCPLAYER\_FREERUN; 2, HCPLAYER\_AUDIO\_MASTER

双路播放时，可能需要media player关闭声音(只输出1路声音disav)，这时需要设置这个freerun。  
无声音的那路播放设置“-s 0”: freerun。

否则可能同步，卡顿。

多路多媒体播放: -u 指定名称, 如果单路媒体播放, 无需-u参数

2K DE, 辅图

```
play /media/sda1/gongxiami.mp4 -z 1 -y 0 -l 1 -u 111
```

2K DE, 主图

```
play /media/sda1/gongxiami.mp4 -z 1 -y 0 -l 0 -u 111
```

4K DE, 辅图

```
play /media/sda1/gongxiami.mp4 -z 1 -y 1 -l 1 -u 222
```

4K DE, 主图

```
play /media/sda1/gongxiami.mp4 -z 1 -y 1 -l 0 -u 222
```

\* 缩放 111的多媒体; 无该参数, 所有多媒体都缩放

```
preview 111 0,0,1920,1080,0,0,960,540
```



```
* 旋转 111的多媒体；无该参数，所有多媒体都选择
-r: 1, 90; 2, 180; 3, 270
-m 0: 无镜像; 1: 左右镜像翻转; 2: 上下镜像翻转
-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;
rotate 111 -f 1

* 停止 111的多媒体；无该参数，所有多媒体都停止
stop 111
```

## 15.1.4 有线同屏

```
* 启动参数
-d 0: 2K DE; 1: 4K DE
-l 0: 主图层; 1: 辅助层
-r rotate 0: 0度; 1: 90度; 2: 180度; 3: 270度。 ROTATE_TYPE_180...
-m mirror 0: 无翻转; 1: 左右镜像翻转
-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;

* 启动同屏
2K DE, 主图:
cast_usb start -p 1 -d 0 -l 0
2K DE, 辅图:
cast_usb start -p 1 -d 0 -l 1
4K DE, 主图:
cast_usb start -p 1 -d 1 -l 0
4K DE, 辅图:
cast_usb start -p 1 -d 1 -l 1

* 旋转
cast_usb rotate -r 1 -f 1

* 缩放
cast_usb preview 0 0 960 540
停止，再启动同屏才生效

* 停止
cast_usb stop
```

注意：播放全屏视频，会自动处理全屏旋转播放（不变形）， 旋转90，270，将不会改变全屏视频的旋转

## 15.1.5 HDMI in

**注意：HDMI in和CVBS in不能同时双路播放**

```
启动参数:
* 启动参数
```

```
-v 1: HDMI_RX_VIDEO_TO_DE; -v 4: HDMI_RX_VIDEO_TO_DE_ROTATE, 支持旋转
-t 0: 2K DE; 1: 4K DE
-l 0: 主图层; 1: 辅图层
-r rotate 0: 0度; 1: 90度; 2: 180度; 3: 270度。 ROTATE_TYPE_180...
-m mirror 0: 无翻转; 1: 左右镜像翻转
-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;
-g 使用audio dec解码: 1, enable; 0, disable. 可以和媒体播放器混音
-h 关闭音频: 1, 关闭; 0, 打开
-a 设置音频路径 0: HDMI_RX_AUDIO_BYPASS_TO_HDMI_TX, to HDMI_RX; 206:
HDMI_RX_AUDIO_TO_SPDIF_IN_AND_I2SO
```

A系列, hdmi tx输出显示, hdmi rx的音频是: -a 0: HDMI\_RX\_AUDIO\_BYPASS\_TO\_HDMI\_TX, 输出到 HDMI TX

C/D系列, LCD输出显示, hdmi rx的音频是: -a 206: HDMI\_RX\_AUDIO\_TO\_SPDIF\_IN\_AND\_I2SO, 输出到I2SO

\*进入hdmi in

# hdmi\_rx

\* 启动hdmi in

2K DE, 辅图

(hdmi\_rx) # start -v 4 -a 0 -p 1 -t 0 -l 1

2K DE, 主图

(hdmi\_rx) # start -v 4 -a 0 -p 1 -t 0 -l 0

4K DE, 辅图

(hdmi\_rx) # start -v 4 -a 0 -p 1 -t 1 -l 1

4K DE, 主图

(hdmi\_rx) # start -v 4 -a 0 -p 1 -t 1 -l 0

\* 投屏模式

-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;

(hdmi\_rx)# rotate -f 1

\* 缩放:

(hdmi\_rx) # preview 0 0 1920 1080 900 0 960 540

\* 停止

(hdmi\_rx) #stop

问题: hdmi rx和media player同时启动, midia player会没有声音; 停止hdmi rx或者拔出hdmi rx, media player恢复声音

## 15.1.6 CVBS in

**注意: HDMI in和CVBS in不能同时双路播放**

\* 启动参数

-v 0: TVDEC\_VIDEO\_TO\_DE; -v 1: TVDEC\_VIDEO\_TO\_DE\_ROTATE, 支持旋转

-y 0: 2K DE; 1: 4K DE

-l 0: 主图层; 1: 辅图层

-r 0: 0度; 1: 90度; 2: 180度; 3: 270度。 ROTATE\_TYPE\_180...

-m mirror 0: 无翻转; 1: 左右镜像翻转

-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;

\* 进入CVBS in

```
# tv_dec

* 启动
2K DE, 辅图
(tv_dec)# start -v 1 -p 1 -y 0 -l 1
2K DE, 主图
(tv_dec)# start -v 1 -p 1 -y 0 -l 0
4K DE, 辅图
(tv_dec)# start -v 1 -p 1 -y 1 -l 1
4K DE, 主图
(tv_dec)# start -v 1 -p 1 -y 1 -l 0

* 投屏模式
-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;
(tv_dec)# rotate -f 1

* 缩放:
(tv_dec) # preview 0 0 1920 1080 900 0 960 540

* 停止
(tv_dec)# stop
```

### 15.1.7 切换图层顺序

同一屏幕的PIP画中画输出测试时， 可以调整主图层和辅图层顺序

```
参数:
-d: 1, 2K DE; 2: 4K DE
-o: 主图层顺序, 3最顶层, 0最底层
-l: 辅图层顺序, 3最顶层, 0最底层
-r: fb1顺序, 3最顶层, 0最底层
-g: fb0顺序, 3最顶层, 0最底层

2K DE, 主图层在下, 辅图层在上
# dis layerorder -d 1 -o 0 -l 1 -r 2 -g 3
2K DE, 主图层在上, 辅图层在下
# dis layerorder -d 1 -o 1 -l 0 -r 2 -g 3
4K DE, 主图层在下, 辅图层在上
# dis layerorder -d 2 -o 0 -l 1 -r 2 -g 3
4K DE, 主图层在上, 辅图层在下
# dis layerorder -d 2 -o 1 -l 0 -r 2 -g 3
```

### 15.1.8 设置宽高比

```
参数
-d: 1, 2k DE; 2, 4K DE
-t: 0, 4:3; 1, 16:9; 2, auto
-l 0: 主图层; 1: 辅图层, 默认主图层
-m: 3, pill box(16:9, 原图黑边); 5, normal scale(full screen)
```

```
2K DE, pillbox
# dis aspect -d 1 -l 0 -t 1 -m 3
2K DE, normal scale
# dis aspect -d 1 -l 0 -t 1 -m 5
4K DE, pillbox
# dis aspect -d 2 -l 0 -t 1 -m 3
4K DE, normal scale
# dis aspect -d 2 -l 0 -t 1 -m 5
```

### 15.1.9 开关图层

图层的一些内存是共享的，输出到一个图层，可能会影响另外图层的显示。因此开机时，最好先将视频图层关闭。视频播放时，会自动打开图层

```
参数：
-d: 1, 2K DE; 2, 4K DE
-l: 1, 主图层; 16, 辅图层
-o: 0, 关闭; 1, 打开
关闭2K DE, 主图
dis winon -d 1 -l 1 -o 0
关闭2K DE, 辅图
dis winon -d 1 -l 16 -o 0
关闭4K DE, 主图
dis winon -d 2 -l 1 -o 0
关闭2K DE, 辅图
dis winon -d 2 -l 16 -o 0
```

## 15.2 Linux测试

要支持测试多个数据源测试，需要在config中打开对应配置：

```
> External options > hc-examples [*] video picture by picture mode test
examples
> External options > hc-examples [*] bluetooth test
> External options > hc-examples [*] HDMI rx test
> External options > hc-examples [*] TV decoder test
```

如果测试D3000\_V20板子，先把蓝牙初始化，给usb口供电

```
# bluetooth
bluetooth# init
```

测试无线同屏时，我们用标准网络工具手动在串口终端联网，先必须保证平台处于station模式，并连接成功网络

## D3000\_V20

如果使用板子 D3000\_V20测试，板子的micro usb的电压（wifi dongle）是蓝牙控制的。开机进入控制台需要用蓝牙打开micro usb电压供电，否则micro usb的wifi dongle不能工作。同时，需要把蓝牙的音频通路选择为SPDIF in才能播放声音

dts的蓝牙节点：

```
bluetooth {
.....
    pinmux-lineout-det = <7 0>;      // <pinpad pinmux>
    pinmux-lcd-backlight = <13 2>;    // <pinpad pinmux>
    wifien-gpios = <22 1>;            // <pinpad pinpad_value>
    devpath = "/dev/uart0";
    status = "okay";
};
```

蓝牙还控制音频的传输：

```
#bluetooth
bluetooth:# audio_channel 0      蓝牙模块选择接收line in输入，如CVBS-in的音频
bluetooth:# audio_channel 1      蓝牙模块选择接收SPDIF IN输入，如HDMI-in多媒体播放，
                                  同屏播放的音频
```

## 15.2.1 联网

### 15.2.1.1 station模式

```
/etc/./wifiprobe.sh
sleep 5
wpa_cli -i wlan0 add_network
wpa_cli -i wlan0 set_network 0 ssid '"TP-LINK_A422"'
wpa_cli -i wlan0 set_network 0 psk '"hichip1234"'
wpa_cli -i wlan0 enable_network 0
wpa_cli -i wlan0 select_network 0
udhcpc -i wlan0 -q
```

可以直接输出窗口一次输入

wpa\_cli -i wlan0 status 可以查询联网状态

和rtos测试不同，linux在一个应用中测试： multi\_dis。

进入应用：

```
# multi_dis
multi_dis:# help

Commands available:
  help          Show available cmds
  exit          Exit from current cmd set
  history       Show history cmds
  pbp           Entering pbp testing
  mplayer       Entering multi display for MP
  cast_wifi     Enter wifi cast multi display testing
  cast_usb      Enter usb cast multi display testing
  tv_dec        Enter tv decoder(CVBS rx) testing
  hdmi_rx       Enter HDMI in testing
```

可以看到multi\_dis下的各个测试子项目

- 继续敲入命令，可以进入下一级子测试项
- exit, 退到上一级目录
- 不要Ctrl+c, 会退出整个multi\_dis测试

进入测试:

```
# multi_dis
# multi_dis:# pbp
multi_dis:(pbp)# init
```

然后退出 (exit) 到multi\_dis下, 开始测试

## 15.2.2 无线同屏

投影名称: my-hccast-test-xxxx

\* 启动参数

```
-d 0: 2K DE; 1: 4K DE
-l 0: 主图层; 1: 辅助层
-r rotate 0: 0度; 1: 90度; 2: 180度; 3: 270度。 ROTATE_TYPE_180...
-m mirror 0: 无翻转; 1: 左右镜像翻转
-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;
-o 设置音频路径: 1//HUDI_AUDSINK_TYPE_I2SO(1<<0) | HUDI_AUDSINK_TYPE_SPO(1<<2);
-g 使用audio dec解码: 1, enable; 0, disable. 可以和媒体播放器混音
-h 关闭音频: 1, 关闭; 0, 打开
```

\* 进入同屏

```
multi_dis:# cast_wifi
```

开始同屏

2K DE, 主图:

```
start -p 1 -d 0 -l 0
```

2K DE, 辅图:

```
start -p 1 -d 0 -l 1
```

4K DE, 主图:

```
start -p 1 -d 1 -l 0
```

4K DE, 辅图:

```
start -p 1 -d 1 -l 1
```

\* 投屏模式

```
rotate -f 1
```

\* 缩放

```
preview 0 0 960 540
```

停止, 再启动同屏才生效

\* 停止同屏

```
stop
```

在miracast同屏需要独占网卡, 断开连接后, 如果要重新同屏, 要stop/start miracast

1. 要dlna, aircast同屏, 需要重新联网, 旧有同屏服务要重新启动

```
stop
```

ctrl+z, multi\_dis切换到后台

```
# wpa_cli -i wlan0 select_network 0
```

```
# udhcpc -i wlan0 -q
```

```
# fg, multi_dis切回到前台
```

```
multi_dis:(cast_wifi)# start -p 1
```

2. 继续miracast同屏, 要先停止miracast服务, 再继续启动服务, 当然, 直接第一步也可以

```
stop
```

```
start -p 1
```

dlna, aircast同屏结束, 无需重新启动服务

## 15.2.3 多媒体播放

**注意: 目前图片播放只能主图层输出(-l 0)**

\* 进入mdiea player测试

```
multi_dis:# mplayer
```

双路测试时, 可以先关闭多媒体音频, 否则可能和同屏等有声音干扰问题。这个是个开关, 后面播放命令根据这个开关开启/关闭声音

```
# disav -v 0 -a 1
```

\* 注意: 如果需要支持播放两路媒体播放器, 可以先用命令multi 进入多路媒体播放模式, single退出多路媒体播放模式

```
# multi //使能多路多媒体播放
```

\*播放gongxiami.mp4到4K DE

```
# play /media/sda1/gongxiami.mp4 -z 1 -y 1 -l 0 -u 111
```

-u 111为自定义的播放文件名

\*播放test.mp4到2K DE

```
# play /media/sda1/test.mp4 -z 1 -y 0 -l 0 -u 222
```

-u 222为自定义的播放文件名

后续使用如下指定缩放111路缩放，对应pause命令也可以类似加入111文件名暂停该路播放

```
# preview 111 0,0,1920,1080,0,0,960,540
```

调整音量：

```
# volume 100
```

\* 播放参数

```
-y 0: 2K DE; 1: 4K DE
```

```
-l 0: 主图层; 1: 辅图层
```

```
-r: 1, 90; 2, 180; 3, 270
```

```
-m 0: 无镜像; 1: 左右镜像翻转; 2: 上下镜像翻转
```

```
-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;
```

```
-o 设置音频路径: 1//AUDSINK_SND_DEVBITS_I2SO(1<<0) | AUDSINK_SND_DEVBITS_SPO(1<<2);
```

```
-s freerun: 0, HCPLAYER_FREERUN; 1, HCPLAYER_FREERUN; 2, HCPLAYER_AUDIO_MASTER
```

双路播放时，可能需要media player关闭声音(只输出1路声音disav)，这时需要设置这个freerun。  
无声音的那路播放设置“-s 0”：freerun。

否则可能同步，卡顿。

多路多媒体播放： -u 指定名称， 如果单路媒体播放，无需-u参数

2K DE， 辅图

```
play /media/hdd/gongxiami.mp4 -z 1 -y 0 -l 1 -u 111
```

2K DE， 主图

```
play /media/hdd/gongxiami.mp4 -z 1 -y 0 -l 0 -u 111
```

4K DE， 辅图

```
play /media/hdd/gongxiami.mp4 -z 1 -y 1 -l 1 -u 222
```

4K DE， 主图

```
play /media/hdd/gongxiami.mp4 -z 1 -y 1 -l 0 -u 222
```

\* 缩放 111的多媒体；无该参数，所有多媒体都缩放

```
preview 111 0,0,1920,1080,0,0,960,540
```

\* 投屏模式 111的多媒体；无该参数，所有多媒体都选择

```
-r: 1, 90; 2, 180; 3, 270
```

```
-m 0: 无镜像; 1: 左右镜像翻转; 2: 上下镜像翻转
```

```
-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;
```

```
rotate 111 -f 1
```

\* 停止 111的多媒体；无该参数，所有多媒体都停止

```
stop 111
```

## 15.2.4 有线同屏

\* 启动参数

```
-d 0: 2K DE; 1: 4K DE
```

```
-l 0: 主图层; 1: 辅助层
```

```
-r rotate 0: 0度; 1: 90度; 2: 180度; 3: 270度。 ROTATE_TYPE_180...
```

```
-m mirror 0: 无翻转; 1: 左右镜像翻转
```

```
-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;
```

\* 进入测试

```
multi_dis:# cast_usb
```



```
* 启动同屏
2K DE, 主图:
start -p 1 -d 0 -l 0
2K DE, 辅图:
start -p 1 -d 0 -l 1
4K DE, 主图:
start -p 1 -d 1 -l 0
4K DE, 辅图:
start -p 1 -d 1 -l 1
```

```
* 投屏模式
rotate -f 1
```

```
* 缩放
preview 0 0 960 540
停止, 再启动同屏才生效
```

```
* 停止
stop
```

注意: 播放满屏视频, 会自动处理全屏旋转播放(不变形), 旋转90, 270, 将不会改变全屏视频的旋转

## 15.2.5 HDMI in

**注意: HDMI in和CVBS in不能同时双路播放**

启动参数:

\* 启动参数

```
-v 1: HDMI_RX_VIDEO_TO_DE; -v 4: HDMI_RX_VIDEO_TO_DE_ROTATE, 支持旋转
-t 0: 2K DE; 1: 4K DE
-l 0: 主图层; 1: 辅图层
-r rotate 0: 0度; 1: 90度; 2: 180度; 3: 270度。 ROTATE_TYPE_180...
-m mirror 0: 无翻转; 1: 左右镜像翻转
-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;
-g 使用audio dec解码: 1, enable; 0, disable. 可以和媒体播放器混音
-h 关闭音频: 1, 关闭; 0, 打开
-a 设置音频路径 0: HDMI_RX_AUDIO_BYPASS_TO_HDMI_TX, to HDMI_RX; 206:
HDMI_RX_AUDIO_TO_SPDIF_IN_AND_I2SO
```

A系列, hdmi tx输出显示, hdmi rx的音频是: -a 0: HDMI\_RX\_AUDIO\_BYPASS\_TO\_HDMI\_TX, 输出到HDMI TX

C/D系列, LCD输出显示, hdmi rx的音频是: -a 206: HDMI\_RX\_AUDIO\_TO\_SPDIF\_IN\_AND\_I2SO, 输出到I2SO

\* 进入测试

```
multi_dis:# hdmi_rx
```

\* 启动hdmi in

```
2K DE, 辅图
start -v 4 -a 0 -p 1 -t 0 -l 1
2K DE, 主图
```

```
start -v 4 -a 0 -p 1 -t 0 -l 0
4K DE, 辅图
start -v 4 -a 0 -p 1 -t 1 -l 1
4K DE, 主图
start -v 4 -a 0 -p 1 -t 1 -l 0
```

\* 投屏模式

-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;  
rotate -f 1

\* 缩放:

```
preview 0 0 1920 1080 900 0 960 540
```

\* 停止

stop

问题: hdmi rx和media player同时启动, midia player会没有声音; 停止hdmi rx或者拔出hdmi rx, media player恢复声音

## 15.2.6 CVBS in

**注意: HDMI in和CVBS in不能同时双路播放**

\* 启动参数

-v 0: TVDEC\_VIDEO\_TO\_DE; -v 1: TVDEC\_VIDEO\_TO\_DE\_ROTATE, 支持旋转  
-y 0: 2K DE; 1: 4K DE  
-l 0: 主图层; 1: 辅图层  
-r 0: 0度; 1: 90度; 2: 180度; 3: 270度。 ROTATE\_TYPE\_180...  
-m mirror 0: 无翻转; 1: 左右镜像翻转  
-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;

\* 进入CVBS in

```
multi_dis:# tv_dec
```

\* 启动

```
2K DE, 辅图
start -v 1 -p 1 -y 0 -l 1
2K DE, 主图
start -v 1 -p 1 -y 0 -l 0
4K DE, 辅图
start -v 1 -p 1 -y 1 -l 1
4K DE, 主图
start -v 1 -p 1 -y 1 -l 0
```

\* 投屏模式

-f 投影模式 0: 桌面正投; 1: 吊装正投; 2: 桌面背投; 3: 吊装背投;  
rotate -f 1

\* 缩放:

```
preview 0 0 1920 1080 900 0 960 540
```

\* 停止

stop

## 15.2.7 设置宽高比

参数

-d: 1, 2k DE; 2, 4K DE

-t: 0, 4:3; 1, 16:9; 2, auto

-l 0: 主图层; 1: 辅图层, 默认主图层

-m: 3, DIS\_PILLBOX(16:9, 原图黑边); 5, DIS\_NORMAL\_SCALE(full screen)

#multi\_dis

(multi\_dis) #mplayer

multi\_dis:(mplayer)#dis\_test

2KDE, 主图, DIS\_TV\_16\_9, PILLBOX

aspect\_ratio -d 1 -l 0 -t 1 -m 3

2KDE, 辅图, DIS\_TV\_16\_9, PILLBOX

aspect\_ratio -d 1 -l 1 -t 1 -m 3

4KDE, 主图, DIS\_TV\_16\_9, normal scale

aspect\_ratio -d 2 -l 0 -t 1 -m 5

4KDE, 辅图, DIS\_TV\_16\_9, normal scale

aspect\_ratio -d 2 -l 1 -t 1 -m 5