密级状态： 绝密(　　) 秘密(　　) 内部资料(　　) 公开(　√)
Security Class: Top-Secret (　　) Secret (　　) Internal (　　) Public (√)

Rockchip_基于 DRM 框架的 HDMI 开发指南

Rockchip_HDMI_Based_on_DRM_Framework_Developer_Guide

（第二系统产品部）

(Technical Department, R & D Dept. II）

文件状态： **Status:** [　] 草稿 [　] Draft [　] 正在修改 [　] Modifying [√] 正式发布 [√] Released	**文件标识：** **File No.:**	RK-KF-YF-213
	当前版本： **Current Version：**	V1.3
	作　　者： **Author:**	陈顺庆 Chen Shunqing
	完成日期： **Finish Date:**	2019-11-27
	审　　核： **Auditor:**	
	审核日期： **Finish Date:**	

福州瑞芯微电子股份有限公司
Fuzhou Rockchip Electronics Co., Ltd.

免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

Disclaimer

This document is provided “as is” and Fuzhou Rockchip Electronics Co. Ltd (“the company”) makes no express or implied statement or warranty as to the accuracy, reliability, completeness, merchantability, specific purpose and non-infringement of any statement, information and contents of the document. This document is for reference only.

This document may be updated without any notification due to product version upgrades or other reasons.

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

Brand Statement

Rockchip, Rockchip™ icon, Rockchip and other Rockchip trademarks are trademarks of Fuzhou Rockchip Electronics Co., Ltd., and are owned by Fuzhou Rockchip Electronics Co., Ltd.

All other trademarks or registered trademarks mentioned in this document are owned by their respective owners.

福州瑞芯微电子股份有限公司
Fuzhou Rockchip Electronics Co., Ltd.
地址： 福建省福州市铜盘路软件园 A 区 18 号
网址： www.rock-chips.com
客户服务电话： +86-4007-700-590
客户服务传真： +86-591-83951833
客户服务邮箱： fae@rock-chips.com

Fuzhou Rockchip Electronics Co., Ltd.
Address: No. 18 Building, A District, No.89,software Boulevard Fuzhou,Fujian,PRC
Website: www.rock-chips.com
Customer service tel.: +86-4007-700-590
Customer service fax: +86-591-83951833
Customer service e-mail: fae@rock-chips.com

版 本 历 史 Revision History

版本号 Version no.	作者 Author	修改日期 Revision Date	修改说明 Revision description	备注 Remark
V1.0	陈顺庆 Chen Shunqing	2017-12-14	发布初始版本 Initial version release	
V1.1	陈顺庆 Chen Shunqing	2018-3-21	补丁 5.2 设置 HDMI 旋转说明 Add HDMI rotation setting instruction in 5.2	
V1.2	陈顺庆 Chen Shunqing	2019-2-26	更新 2.23、5.2、5.4 Update 2.23, 5.2, 5.4	
V1.3	陈顺庆 Chen Shunqing	2019-11-27	更新 hwc 属性 Update hwc property	

目 录 Contents

1 简介 Overview

DRM 全称是 Direct Rendering Manager，是 DRI(Direct Rendering Infrastructure)框架的一个组件；Android 新版本逐渐从 Framebuffer 框架迁移到 DRM 上，从内核 4.4 开始，RK 的显示框架逐渐迁移到 DRM 上；本文档介绍如何使用新的显示框架，适用于以下 SDK：

The full name of DRM is Direct Rendering Manager, which is a component of DRI (Direct Rendering Infrastructure) framework. Android new versions gradually switch the framework from Framebuffer to DRM. Starting from kernel 4.4, RK display framework is gradually changed to DRM. This document introduces how to use new display framework, which is suitable for the following SDK:

- RK3399 Android7.1 SDK
- RK3368H Android7.1 SDK/ RK3368H Android8.1 SDK
- RK3126C Android8.1 SDK
- RK3288 Android7.1 SDK
- RK3399 Linux SDK

2 变量定义 Variable definition

- Android7.1/Android8.1
  #define PROPERTY_TYPE "sys"
- Android9.0 and after
  #define PROPERTY_TYPE "vendor"

For example format as follows:

Android8.1		Android9.0
hwc.	->	vendor.hwc.
sys.	->	vendor.
persist.sys	->	persist.vendor

3 HDMI/DP 相关配置 HDMI/DP related configuration

3.1 HDMI 配置 dts HDMI configuration dts

3.1.1 使能对应显示设备节点 Enable the corresponding display device node

打开显示设备执行相关 hdmi 的 probe 函数，注册显示设备驱动，如打开 HDMI 需要添加：

Open the display device, execute hdmi related probe function, and register the display device driver.

For example, to enable HDMI, need to add:

```
&hdmi {
        status = "okay";
};
```

3.1.2 使能显示接口组件 Enable the display interface component

display-subsystem 注册会把所有打开的设备以组件的形式加在一起，等所有的组件加载完毕后，统一进行 bind/unbind。

display-subsystem registration will add all the devices enabled together in the form of component, and do bind/unbind after all the components are loaded completely.

3.1.3 绑定 VOP Bind VOP

如果平台存在两个 VOP（RK3288、RK3399）：vopb（支持 4K）、vopl（只支持 2K），当显示设备节点打开时，显示接口对应 vopb 和 vopl 的 ports 都会打开，需要关闭用不到的那个 VOP。

If the platform has two VOP (RK3288, RK3399): vopb (support 4K), vopl (only support 2K). When the display device node is enabled, the display interface will enable ports of vopb and vopl accordingly, and need to close that VOP not used.

比如 hdmi 绑定到 vopb 需要添加：

For example, to bind hdmi to vopb, need to add:

```
&hdmi_in_vopl {
        status = "disabled";
};
```

反之若绑定到 vopl 则添加：

Otherwise, if bind to vopl, need to add:

```
&hdmi_in_vopb {
        status = "disabled";
};
```

如果平台只有一个 VOP，可以跳过。

If the platform only has a VOP, you can skip it.

3.1.4 开机 LOGO Boot up LOGO

如果 uboot logo 未开启，那 kernel 阶段也无法显示开机 logo，只能等到 android 启动后才能看到显示；在 dts 里面将对应的 route 使能即可打开 uboot logo 支持，比如打开 hdmi 的 uboot logo 显示:

If uboot logo is not enabled, it also cannot display boot up logo in kernel stage, and the display can be seen only after entering android. Enable the corresponding route in dts can enable uboot logo support, such as enabling the hdmi uboot logo display:

```
&route_hdmi {
    status = "okay"
};
```

3.1.5 绑定 PLL（只有 RK3399 需要） Bind PLL (RK3399 only)

rk3399 的 hdmi 所绑定的 vop 时钟需要挂载到 vpll 上，若是双显，需将另一个 vop 时钟挂到 cpll，这样可以分出任意 dclk 的频率；如当 hdmi 绑定到 vopb 时配置：

The vop clock bound by RK3399 hdmi should be loaded to vpll. For dual display, need to load the other vop clock to cpll, in this way it can split any dclk frequency. For example, when hdmi is bound to vopb, the configuration is as below:

```
&vopb {
    assigned-clocks = <&cru DCLK_VOP0_DIV>;
    assigned-clock-parents = <&cru PLL_VPLL>;
};
&vopl {
    assigned-clocks = <&cru DCLK_VOP1_DIV>;
    assigned-clock-parents = <&cru PLL_CPLL>;
} ;
```

当 hdmi 绑定到 vopl 时配置：

When hdmi is bound to vopl, the configuration is as below:

```
&vopb {
    assigned-clocks = <&cru DCLK_VOP0_DIV>;
    assigned-clock-parents = <&cru PLL_CPLL>;
};
&vopl {
    assigned-clocks = <&cru DCLK_VOP1_DIV>;
    assigned-clock-parents = <&cru PLL_VPLL>;
```

```
};
```

3.2 HDMI 相关配置说明 HDMI related configuration instruction

3.2.1 信号强度配置(RK3288/RK3368/RK3399) Signal strength configuration (RK3288/RK3368/RK3399)

由于硬件走线差异，不同板子有可能需要不同的驱动强度配置，当遇到电视兼容性问题时，可以尝试修改这个看是否有改善。

Different boards may need different drive strength configuration due to different hardware layout. When encountering the television compatibility issue, you can try to modify this configuration.

hdmi 信号强度可通过 dts 的 rockchip.phy-table 属性配置，格式定义:<PIXELCLOCK PHY_CKSYMTXCTRL PHY_TXTERM PHY_VLEVCTRL>.

Hdmi signal strength can be configured by rockchip.phy-table property in dts. The format definition: <PIXELCLOCK PHY_CKSYMTXCTRL PHY_TXTERM PHY_VLEVCTRL>.

PIXELCLOCK 表示所在行参数所对应的最大 pixelclock 频率。

PIXELCLOCK means the max pixelclock frequency corresponding to the line parameter.

PHY_CKSYMTXCTRL 寄存器(0x09)值用于调整 HDMI 信号的预加重和上升斜率，加大预加重或 sloop boost，可以提升 Data 信号的上升斜率，但会降低信号的上升/下降时间。

PHY_CKSYMTXCTRL register (0x09) value is used to adjust the pre-emphasis and rising slope of HDMI signal. Increase the pre-emphasis or sloop boost can enhance the rising slope of Data signal and reduce the rise/fall time of the signal.

Bit[0]：Clock 信号使能。

Bit[0]: Clock signal enable

Bit[3:1]：Data 信号预加重，定义如下 rockchip,phy-table。

Bit[3:1]: Data signal pre-emphasis, defined as rockchip,phy-table below

Bit[4:5]：Data 信号 sloop boost。

Bit[4:5]: Data signal sloop boost

PHY_TXTERM 寄存器(0x19)值用于调整端接电阻值。

PHY_TXTERM register (0x19) value is used to adjust the terminal resistance value.

Bit[0:2]：值越大，端接电阻值越大。

Bit[0:2]: the larger the value, the larger the terminal resistance value

PHY_VLEVCTRL 寄存器(0x0e)值用于调整 HDMI 的信号幅度，具体定义如下 rockchip,phy-table：

PHY_VLEVCTRL register (0x0e) value is used to adjust the signal amplitude of HDMI, defined as rockchip,phy-table below

Bit[0:4] : tmds_clk +/- 信号幅度，值越低，驱动能力越强。

Bit[0:4]: the signal amplitude of tmds_clk +/-, the smaller the value, the stronger the drive capability

Bit[5:9]: tmds_data +/- 信号幅度，值越低，驱动能力越强。

Bit[5:9]: the signal amplitude of tmds_data +/-, the smaller the value, the stronger the drive capability

如：

For example:

```
 &hdmi {
            rockchip,phy-table =
                    <74250000 0x8009 0x0004 0x0272>,
                    <165000000 0x802b 0x0004 0x0209>,
                    <297000000 0x8039 0x0005 0x028d>,
                    <594000000 0x8039 0x0000 0x019d>,
                    <000000000 0x0000 0x0000 0x0000>;
};
```

其中<74250000 0x8009 0x0004 0x0272>，表示 pixeclock 为 74.25M(720p 分辨率)以下时，PHY_CKSYMTXCTRL 寄存器值为 0x8009，PHY_TXTERM 值为 0x0004，PHY_VLEVCTRL 值为 0x0272。修改后也可用 cat /d/dw-hdmi/phy 命令查看对应的寄存器值，确认是否有修改成功。

<74250000 0x8009 0x0004 0x0272> means when the pixelclock is below 74.25M (720p resolution), PHY_CKSYMTXCTRL register value is 0x8009, PHY_TXTERM value is 0x0004, and PHY_VLEVCTRL value is 0x0272. After modification, you can also use cat /d/dw-hdmi/phy command to check the corresponding register value to confirm if it is modified successfully.

3.2.2 DDC 的 I2C 速率配置 DDC I2C rate configuration

目前 i2c 速率通过 clk 高电平和低电平的时间来定义，如下为实测 i2c 速率为 50k 时候的配置，调整 i2c 速率只需将这 2 个值按对应的比例修改即可。

Currently i2c rate is defined by the time of clk high level and low level. Below is the configuration when the actually measured i2c rate is 50k. To adjust i2c rate, only need to modify these two values with the corresponding ratio.

```
&hdmi {
        ddc-i2c-scl-high-time-ns = <9625>;
        ddc-i2c-scl-low-time-ns = <10000>;
}
```

3.2.3 使能 HDCP1.4 Enable HDCP1.4

```
&hdmi {
```

```
        hdcp1x-enable = <1>;
}
```

使能 HDCP1.4 后还需要通过工具烧录对应 key，当前工具版本为“ProvisioningTool20170526.zip”，具体操作详见工具内的使用文档。

After enabling HDCP1.4, need to flash the corresponding key by the tool. Current tool version is “ProvisioningTool20170526.zip”. For the detailed operation, please refer to the tool instruction document.

3.2.4 打开音频 Enable audio

RK3399 目前 HDMI 声卡和 DP 公用：

Currently RK3399 HDMI sound card and DP are reused:

```
&hdmi_dp_sound {

        status = "okay";

};
```

3.3 DP 配置说明（RK3399）DP configuration instruction (RK3399)

3.3.1 DP 检测 DP detection

RK3399 的 typec 支持 dp/usb3/usb2，sdk 默认使用 fusb302 来检测接入的设备类型；当设备接入时，fusb302 通过 extcon 传递给 usb 驱动；fusb302 是通过 i2c 外挂的芯片，下面配置是挂到 i2c4 上时打开的配置，若挂在其他 i2c 上则需要对应修改。

RK3399 typec supports dp/usb3/usb2. Sdk uses fusb302 to detect the connected device type by default. When the device is connected, fusb302 transmits to usb driver through extcon. Fusb302 is the chip externally connected through i2c, and the following is the enable configuration when it is connected to i2c4. If it is connected to other i2c, need to make the modification accordingly.

```
&i2c4 {

        status = "okay";

        fusb0: fusb30x@22 {

                status = "okay";

        };

};
```

3.3.2 绑定 typec 口 Bind typec port

RK3399 有两个功能相同的 typec 口，都支持 dp 输出，不过由于 dp 控制器只有一个，所以同一时刻最多只能有一个 typec 口输出 dp 信号。

RK3399 has two typec ports with the same function. Both of them support dp output, but there is only one dp controller, so only one typec port can output dp signal at the same time.

typec0 口包括 usb 控制器(&usbdrd3_0); usb3phy(&tcphy0)和 usb2phy (&u2phy0);

typec0 port includes usb controllers (&usbdrd3_0); usb3phy(&tcphy0) and usb2phy (&u2phy0);

typec1 口包括 usb 控制器(&usbdrd3_1);usb3phy(&tcphy1)和 usb2phy (&u2phy1);

typec1 port includes usb controllers (&usbdrd3_1);usb3phy(&tcphy1) and usb2phy (&u2phy1);

若 fusb302 接到 typec0 口时，需配置如下：

If fusb302 is connected to typec0 port, need to configure as below:

```
&tcphy0 {
        extcon = <&fusb0>;
        status = "okay";
};
&u2phy0 {
        status = "okay";
        extcon = <&fusb0>;
};
&usbdrd3_0 {
        extcon = <&fusb0>;
        status = "okay";
};
```

若 fusb302 接到 typec1 口时，需配置如下：

If fusb302 is connected to typec1 port, need to configure as below:

```
&tcphy1{
        extcon = <&fusb0>;
        status = "okay";
};
&u2phy1 {
```

```
        status = "okay";
        extcon = <&fusb0>;
};
&usbdrd3_1 {
        extcon = <&fusb0>;
        status = "okay";
};
```

3.3.3 注册 DP 驱动 Register DP driver

打开 dp 的 dts 同时绑定 extcon 配置：

Bind extcon configuration when enabling the dts of dp:

```
&cdn_dp {
        status = "okay";
        extcon = <&fusb0>;
}
```

3.3.4 不使用 fusb302 将 typec 口固定作 DP 输出 Fix typec port as DP output when not using fusb302

有些产品不接 fusb302，而将其中一个 typec 口固定做 dp 输出，这时需要自己添加一个 extcon 驱动，如 vpd0（目前补丁 vpd0.patch 还未提交），然后将 dp 和 usb 的 extcon 设置成 vpd0 如下：

Some products don’t connect fusb302, and one typec port is fixed as dp output, then you need to add an extcon driver, such as vpd0 (current patch vpd0.patch is not submitted yet), and then set the extcon of dp and usb as vpd0 as below:

```
vpd0: virtual-pd0 {
        status = "okay";
}
&cdn_dp {
        status = "okay";
        extcon = <&vpd0>;
}
```

```
&tcphy0 {
        extcon = <&vpd0>;
        status = "okay";
};
//以上为接到 typec0 口时的配置，若是接到 typec1 口则需将 tcphy0 改为 tcphy1 如下：Above is the configuration when it is connected to typec0 port. If it is connected to typec1 port, need to replace tcphy0 with tcphy1 as below:
/*&tcphy1 {
        extcon = <&vpd0 >;
        status = "okay";
};*/
```

3.4 参考配置 Reference configuration

3.4.1 EDP(vopb) + HDMI(vopl)

```
/* 打开 edp 设备节点 Enable the node of edp device*/
&edp {
    status = "okay";
};
/* 绑定 edp 到 vopb  Bind edp to vopb*/
&edp_in_vopl {
    status = "disabled";
};
/* 开启 edp 的 uboot  logo 显示 Enable uboot logo display for edp*/
&route_edp {
    status = "okay"
};
```

```
/* 打开 hdmi 设备节点 Enable the node of hdmi device*/
&hdmi {
	status = "okay";
};
/* 绑定 hdmi 到 vopl	Bind hdmi to vopl*/
&hdmi_in_vopb {
	status = "disabled";
};
```

RK3399 还需将 hdmi 绑定的 vopl 时钟挂到 vpll 上，如下（其他芯片不需要配置）：

Besides, RK3399 needs to load the vopl clock bound by hdmi to vpll as below (no need to configure for other chips):

```
&vopb {
	assigned-clocks = <&cru DCLK_VOP0_DIV>;
	assigned-clock-parents = <&cru PLL_CPLL>;
};
&vopl {
	assigned-clocks = <&cru DCLK_VOP1_DIV>;
	assigned-clock-parents = <&cru PLL_VPLL>;
};
```

3.4.2 LVDS(vopl) + HDMI(vopb)

```
//使能 LVDS	Enable LVDS
&lvds {
	status = "okay";
};
//绑定 LVDS 到 vopl	Bind LVDS to vopl
&lvds_in_vopb {
	status = "disabled";
```

```
};
//LVDS 屏显示 uboot logo      LVDS panel displays uboot logo
&route_lvds{
        connect = <&vopl_out_lvds>;
        status = "okay"
};
//使能 HDMI      Enable HDMI
&hdmi {
        status = "okay";
};
//绑定 HDMI 到 vopb      Bind HDMI to vopb
&hdmi_in_vopl {
        status = "disabled";
};
```

RK3399 还需将 hdmi 绑定的 vopb 时钟挂到 vpll 上，如下（其他芯片不需要配置）：

Besides, RK3399 needs to load the vopb clock bound by hdmi to vpll as below (no need to configure for other chips):

```
&vopb {
        assigned-clocks = <&cru DCLK_VOP0_DIV>;
        assigned-clock-parents = <&cru PLL_VPLL>;
};
&vopl {
        assigned-clocks = <&cru DCLK_VOP1_DIV>;
        assigned-clock-parents = <&cru PLL_CPLL>;
};
```

3.4.3 MIPI(vopb) + HDMI(vopl)

```
&mipi {
```

```
        status = "okay";
};
&mipi_in_vopl {
        status = "disabled";
};
&hdmi {
        status = "okay";
};
&hdmi_in_vopb {
        status = "disabled";
};
&route_mipi{
        status = "okay"
};
```

RK3399 还需将 hdmi 绑定的 vopl 时钟挂到 vpll 上，如下（其他芯片不需要配置）：

Besides, RK3399 needs to load the vopl clock bound by hdmi to vpll as below (no need to configure for other chips):

```
&vopb {
        assigned-clocks = <&cru DCLK_VOP0_DIV>;
        assigned-clock-parents = <&cru PLL_CPLL>;};
&vopl {
        assigned-clocks = <&cru DCLK_VOP1_DIV>;
        assigned-clock-parents = <&cru PLL_VPLL>;
};
```

3.4.4 HDMI(vopb) + DP(vopl)

```
&hdmi {
        status = "okay";
```

```
};
&hdmi_in_vopl {
	status = "disabled";
};
&cdn_dp {
	status = "okay";
};
&dp_in_vopl {
	status = "disabled";
};
&route_hdmi {
	status = "okay"
};
```

RK3399 还需将 hdmi 绑定的 vopb 时钟挂到 vpll 上，如下（其他芯片不需要配置）：

Besides, RK3399 needs to load the vopb clock bound by hdmi to vpll as below (no need to configure for other chips):

```
&vopb {
	assigned-clocks = <&cru DCLK_VOP0_DIV>;
	assigned-clock-parents = <&cru PLL_VPLL>;
};
&vopl {
	assigned-clocks = <&cru DCLK_VOP1_DIV>;
	assigned-clock-parents = <&cru PLL_CPLL>;
};
```

4 显示框架配置 Display framework configuration

当前 SDK 的显示框架增加了一些系统属性，用于帮助客户能够根据需求配置显示。

Current SDK display framework adds some system properties, aiming to help customers configure display according to the requirements.

4.1 主副显示器配置 Main/secondary display configuration

表 4-1 主副显示器分辨率设置

Table 4-1 Main/secondary display resolution setting

属性 Property	功能说明 Function description
PROPERTY_TYPE.hwc.device.primary	设置显示接口做为主显 Set the display interface as main display
PROPERTY_TYPE.hwc.device.extend	设置显示接口做为副显 Set the display interface as secondary display

Node: PROPERTY_TYPE define in <2.变量定义 Variable definition>

以上两个属性的配置可加在产品配置目录下的 system.prop 里（如 device/rockchip/rk3368/rk3368_mid/system.prop）。

The configurations of above two properties can be added in system.prop under the product configuration directory (such as device/rockchip/rk3368/rk3368_mid/system.prop).

默认情况下(即以上属性未配置时)，不支持热拔插设备（如 CVBS/MIPI/LVDS 等）会作为主显，支持热插拔设备（如 HDMI/DP 等）会作为次显。

By default (when above properties are not configured), it doesn't support to use hot-plug devices (such as CVBS/MIPI/LVDS, etc.) as main display, but supports to use hot-plug devices (such as HDMI/DP, etc.) as secondary display.

通常主、副显只配置一个显示接口，例如 RK3399 MID SDK 默认采用的配置，HDMI 作为主显示，EDP 作为副显示。

Generally it only configures one display interface for the main and secondary display. For example, RK3399 MID SDK default configuration is to use HDMI as main display and EDP as secondary display.

```
PROPERTY_TYPE.hwc.device.primary=HDMI-A

PROPERTY_TYPE.hwc.device.extend=eDP
```

当主/副显配置多个显示接口时，优先使用支持热拔插的设备；例如：

When main/secondary display is configured with multiple display interfaces, prefer to use the device supporting hot-plug. For example:

```
PROPERTY_TYPE.hwc.device.primary=HDMI-A,eDP
```

当 HDMI 插入时，主显使用 HDMI 作为显示，HDMI 拔出时，主显使用 CVBS 作为显示。

When HDMI is plugged, main display will use HDMI as display. When HDMI is unplugged, main display will use CVBS as display.

注意：由于主显的 framebuffer 分辨率无法动态更改，所以有两个或以上设备作为主显时，最好设定一个主显的 framebuffer 分辨率；设置方法见章节 3.3。

Note: The framebuffer resolution of main display cannot be changed dynamically, so when there are two or more devices used as main display, you'd better set a framebuffer resolution for main display. For the setting method, refer to chapter 3.3.

关于接口名称可以参见 hardware/rockchip/hwcomposer/drmresources.cpp 里的定义：

For the interface name, you can refer to the definition in hardware/rockchip/hwcomposer/drmresources.cpp:

```
struct type_name connector_type_names[] = {
    { DRM_MODE_CONNECTOR_Unknown, "unknown" },//未知接口  unknown interface
    { DRM_MODE_CONNECTOR_VGA, "VGA" },      //VGA
    { DRM_MODE_CONNECTOR_DVII, "DVI-I" },//DVI，暂不支持  currently unsupported
    { DRM_MODE_CONNECTOR_DVID, "DVI-D" },//DVI，暂不支持 currently unsupported
    { DRM_MODE_CONNECTOR_DVIA, "DVI-A" },//DVI，暂不支持 currently unsupported
    { DRM_MODE_CONNECTOR_Composite, "composite" },//不支持  unsupported
    { DRM_MODE_CONNECTOR_SVIDEO, "s-video" },//S 端子  S port
    { DRM_MODE_CONNECTOR_LVDS, "LVDS" },//LVDS
    { DRM_MODE_CONNECTOR_Component, "component" },//分量信号 YPbPr Component signal YPbPr
    { DRM_MODE_CONNECTOR_9PinDIN, "9-pin DIN" },//不支持  unsupported
    { DRM_MODE_CONNECTOR_DisplayPort, "DP" },//DP
    { DRM_MODE_CONNECTOR_HDMIA, "HDMI-A" },//HDMI A 型口  HDMI A type
    { DRM_MODE_CONNECTOR_HDMIB, "HDMI-B" },//HDMI B 型口，不支持  HDMI B type, unsupported
    { DRM_MODE_CONNECTOR_TV, "TV" },// CVBS
    { DRM_MODE_CONNECTOR_eDP, "eDP" },//EDP
    { DRM_MODE_CONNECTOR_VIRTUAL, "Virtual" },//不支持  unsupported
    { DRM_MODE_CONNECTOR_DSI, "DSI" },//MIPI
};
```

4.2 主副显示器接口查询 Main/secondary display interface query

可以通过以下两个只读属性来分别查询主副显示器的输出接口的名称。

The following two read only properties can be used to separately inquire the output interface name of main/secondary display.

表 4-2 主副显示器查询

Table 4-2 Main/secondary display query

属性 Property	功能说明 Function description
PROPERTY_TYPE.hwc.device.main	查询当前主显的输出接口 Inquire current output interface of main display
PROPERTY_TYPE.hwc.device.aux	查询当前副显的输出接口 Inquire current output interface of secondary display

Node: PROPERTY_TYPE define in <2.变量定义 Variable definition>

4.3 FrameBuffer 分辨率配置 FrameBuffer resolution configuration

可以通过配置以下属性来设置 FrameBuffer 的分辨率：

The resolution of FrameBuffer can be set by configuring the following property:

```
persist.PROPERTY_TYPE.framebuffer.main=1920x1080
```

Node: PROPERTY_TYPE define in <2.变量定义 Variable definition>

4.4 分辨率过滤配置 Resolution filter configuration

因为初始获取到的全部分辨率过多，有些分辨率对用户来说并不需要，因此在 SDK 的 HWC 模块中对分辨率进行了过滤。

Because there are too many initially acquired resolutions, and some resolution are not useful for users, the resolutions are filtered in HWC module of SDK.

位于 device/rockchip/common/resolution_white.xml 路径的配置文件定义了能够通过过滤的白名单，HWC 中会根据该配置文件对初始的分辨率进行过滤筛选后再传递给上层，该 XML 文件的每一个<resolution>块定义了一个能够通过过滤的分辨率，其中详细项的定义如下：

The configuration file in the path of device/rockchip/common/resolution_white.xml defines the white list of resolution. HWC will filter and sort the initial resolutions according to the configuration file and then transmit to upper layer. Each <resolution> block in this XML file defines a resolution which can pass the filter. For the detailed item definition, refer to below:

表 4-3 分辨率过滤项定义说明

Table 4-3 Resolution filter item definition description

项定义 Item definition	说明 Description
clock	时钟 Clock
hdisplay	行有效像素，见图 4-1 的标示 Line effective pixel, refer to the mark in picture 4-1
hsync_start	行同步起始像素，见图 4-1 的标示 Line sync start pixel, refer to the mark in picture 4-1
hsync_end	行同步结束像素，见图 4-1 的标示 Line sync end pixel, refer to the mark in picture 4-1
htotal	一行总像素，见图 4-1 的标示 Total pixels of a line, refer to the mark in picture 4-1
hskew	行偏差 Line deviation
vdisplay	帧有效行，见图 4-1 的标示 Frame effective lines, refer to the mark in picture 4-1
vsync_start	帧同步开始行，见图 4-1 的标示 Frame sync start line, refer to the mark in picture 4-1
vsync_end	帧同步结束行，见图 4-1 的标示 Frame sync end line, refer to the mark in picture 4-1
vtotal	一帧总行数，见图 4-1 的标示 Total lines of a frame, refer to the mark in picture 4-1
vscan	帧扫描信号 Frame scan signal
vrefresh	显示设备帧率 The frame rate of the display device
flags	flags 的定义如下： flages definition is as below: DRM_MODE_FLAG_PHSYNC (1<<0) DRM_MODE_FLAG_NHSYNC (1<<1) DRM_MODE_FLAG_PVSYNC (1<<2) DRM_MODE_FLAG_NVSYNC (1<<3) DRM_MODE_FLAG_INTERLACE (1<<4)
vic	HDMI 标准对应定义的 VIC 值，如 HDMI 标准中未定义置 0 VIC values defined corresponding to HDMI standard, set 0 if it is not defined in HDMI standard

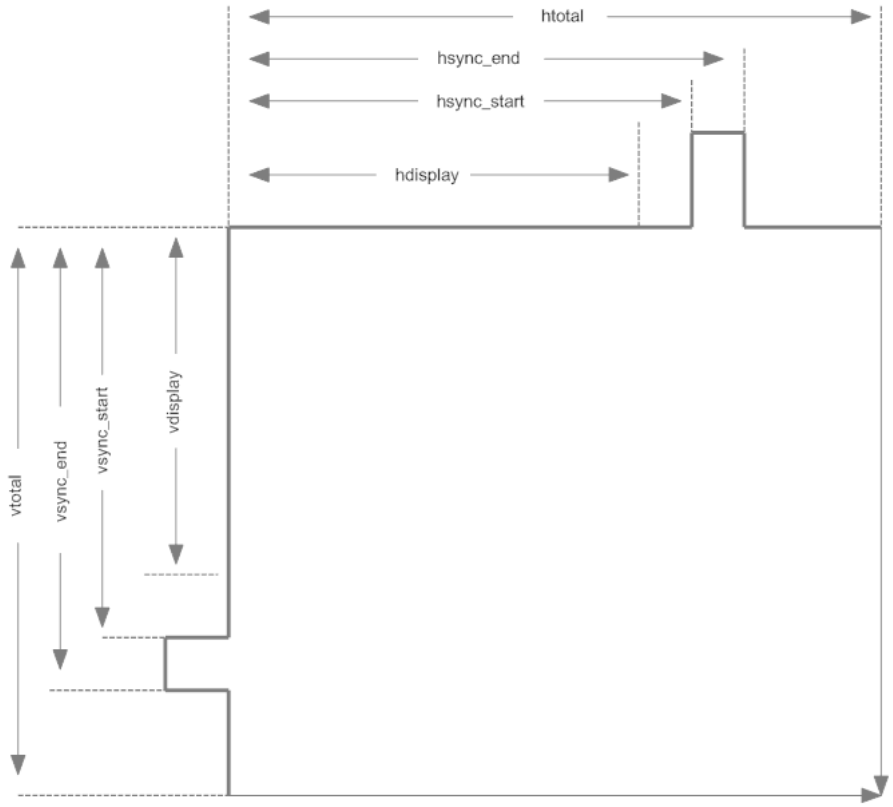

图 4-1 分辨率项定义示意图

Picture 4-1 Resolution item definition view

5 常用调试方法 Commonly used debugging method

5.1 查看 VOP 状态 Check VOP status

```
cat /d/dri/0/summary
```

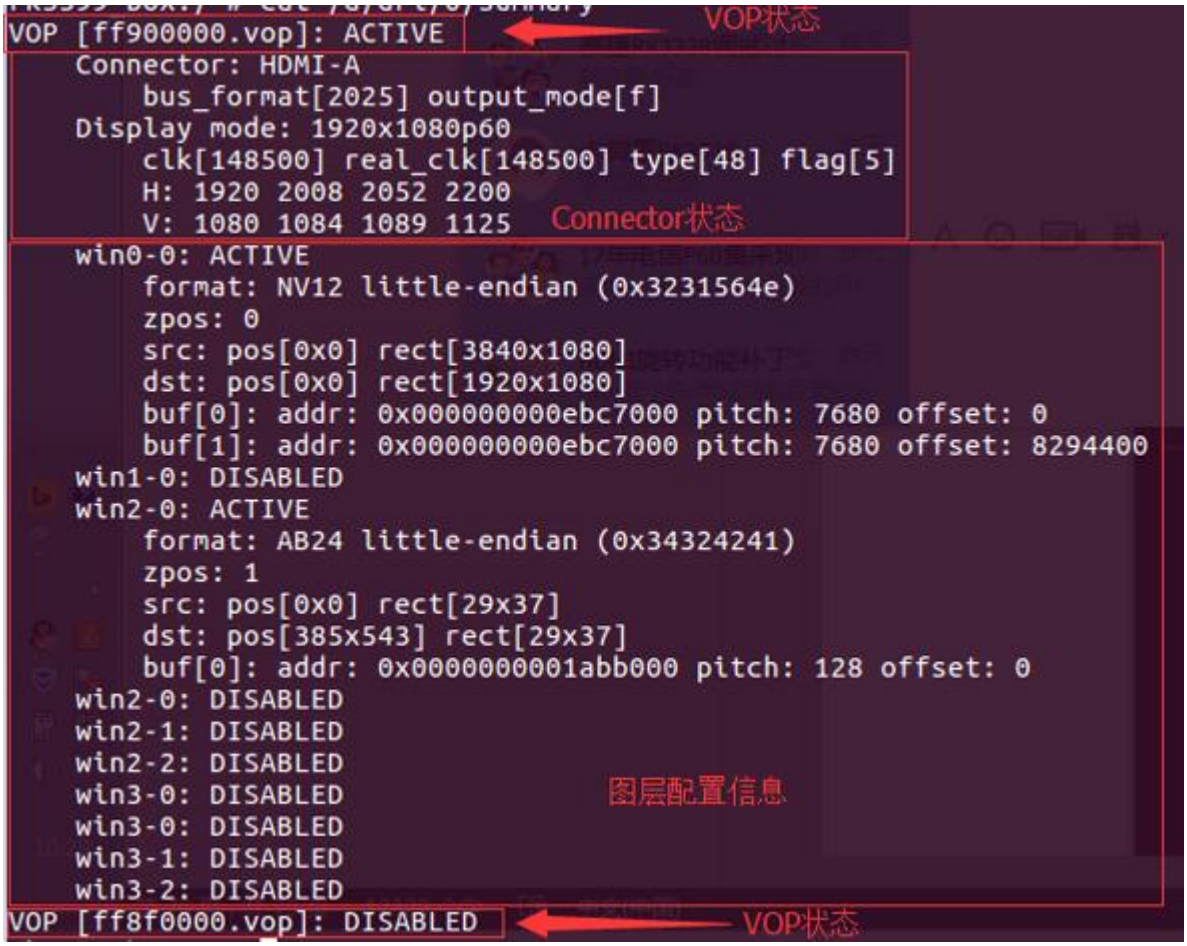

图 5-1

Picture 5-1

图 5-1 是 RK3399 连接 HDMI 时上述命令输出的 Log，可以提供三种信息：

Picture 5-1 shows the log output by the above command when RK3399 is connected with HDMI. It provides three kinds of information:

- VOP 状态： VOPB 处于使能状态，VOPL 处于禁用状态。
  VOP status: VOPB is enabled, VOPL is disabled.

- VOP 对应的 Connector 状态：VOPB 输出信号给 HDMI，bus_format = 0x2025 表示 YUV444 8bit，output_mode = 0x0f 表示 VOP 输出总线为 ROCKCHIP_OUT_MODE_AAAA，输出 1920x1080P60。
  The corresponding Connector status of VOP: VOPB outputs signal to HDMI, bus_format = 0x2025 means YUV444 8bit, output_mode = 0x0f means VOP output bus is ROCKCHIP_OUT_MODE_AAAA, output 1920x1080P60.

常用的 bus_format 由内核 uapi/linux/media-bus-format.h 定义：

The commonly used bus_format is defined by kernel uapi/linux/media-bus-format.h:

```
#define MEDIA_BUS_FMT_RGB888_1X24          0x100a    //RGB888

#define MEDIA_BUS_FMT_RGB101010_1X30       0x1018    //RGB101010
```

```
#define MEDIA_BUS_FMT_YUV8_1X24                 0x2025  //YUV444 8bit
#define MEDIA_BUS_FMT_YUV10_1X30                0x2016  //YUV444 10bit
#define MEDIA_BUS_FMT_UYYVYY8_0_5X24            0x2026  //YUV420 8bit
#define MEDIA_BUS_FMT_UYYVYY10_0_5X30           0x2027  //YUV420 10bit
```

常用的 output_mode 由内核 drivers/gpu/drm/rockchip/rockchip_drm_vop.h 定义：

The commonly used output_mode is defined by kernel drivers/gpu/drm/rockchip/rockchip_drm_vop.h:

```
#define ROCKCHIP_OUT_MODE_P888          0
#define ROCKCHIP_OUT_MODE_P666          1
#define ROCKCHIP_OUT_MODE_P565          2
#define ROCKCHIP_OUT_MODE_S888          8
#define ROCKCHIP_OUT_MODE_S888_DUMMY    12
#define ROCKCHIP_OUT_MODE_YUV420        14
/* for use special outface */
#define ROCKCHIP_OUT_MODE_AAAA          15
```

- 图层配置信息：win0 和 win2 使能，win2 buffer 格式为 ARGB，buffer 大小为 29x37；目标窗口为 29x37，窗口左上角坐标（385，543）。Win0 buffer 格式为 NV12，大小为 3840x2160；目标窗口大小为 1920x1080，窗口左上角坐标（0，0）。
  Image layer configuration information: win0 and win2 are enabled, win2 buffer format is ARGB, and buffer size is 29x37. Current window size is 29x37, and top-left coordinate of window is (385, 543). Win0 buffer format is NV12, size is 3840x2160. Current window size is 1920x1080, and top-left coordinate of window is (0, 0).

5.2 查看 Connector 状态 Check Connector status

/sys/class/drm 目录下可以看到驱动注册的各个输出接口，表 4-1 列出了 RK 平台上常见的输出名称。

You can see each output interface registered by driver in /sys/class/drm directory. Table 4-1 lists the common output interface names on RK platforms.

表 5-1

Table 5-1

名称 Name	类型 Type
card0-DP-1	DP

card0-eDP-1	EDP
card0-HDMI-A-1	HDMI
card0-TV-1	CVBS
card0-LVDS-1	LVDS
card0-DSI-1	MIPI DSI

/sys/class/drm 是 RK3399 平台 drm 目录结构，可以看到注册了 card0-HDMI-A-1 和 card0-DP-1 两种输出，分别表示 HDMI 和 DP。

/sys/class/drm is drm directory structure of RK3399 platform. You can see card0-HDMI-A-1 and card0-DP-1 two output interfaces are registered, which separately represent HDMI and DP.

以 card0-HDMI-A-1 为例，其目录下有以下文件：

Take card0-HDMI-A-1 as example, its directory includes the following files:

- enabled 使能状态 enable status
- status 连接状态 connection status
- mode 当前输出分辨率 current output resolution
- modes 连接设备支持的分辨率列表 the resolution list supported by the device connected
- audioformat 连接设备支持的音频格式 the audio format supported by the device connected
- edid 连接设备的 EDID，可以通过命令 cat edid > /data/edid.bin 保存下来。EDID of the device connected, which can be saved through the command cat edid > /data/edid.bin

5.3 查看 HDMI 状态 Check HDMI status

■ 查看当前输出状态

Check current output status

```
cat /d/dw-hdmi/status
```

HDMI 状态打印如图 5-3 所示：

Print HDMI status as picture 5-3:

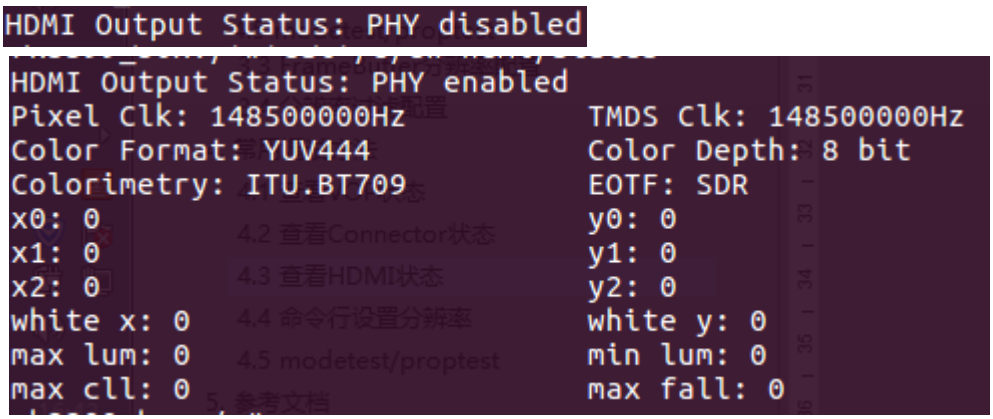

图 5-3

Picture 5-3

- HDMI Output Status 表示当前 PHY 状态，只有当 PHY 使能的时候才会有后续打印。
  HDMI Output Status means current PHY status, it will print further only when PHY is enabled
- Pixel Clk 表示当前输出的像素时钟。
  Pixel Clk means currently output pixel clock.
- TMDS Clk 表示当前输出 HDMI 符号率。
  TMDS Clk means currently output HDMI symbol rate.
- Color Format 表示输出的颜色格式，取值 RGB、YUV444、YUV422、YUV420。
  Color Format means the output color format, with the values RGB, YUV444, YUV422 and YUV420
- Color Depth 表示输出的颜色深度，取值 8bit、10bit、12bit、16bit。
  Color Depth means the output color depth, with the values 8bit, 10bit, 12bit and 16bit.
- Colorimery 表示输出的颜色标准，取值 ITU.BT601、ITU.BIT709、ITU.BT2020
  Colorimery means the output color standard, with the values ITU.BT601, ITU.BIT709 and ITU.BT2020.
- EOTF 表示输出的 HDR 电光转换曲线方式，有如下取值：
  EOTF means the conversion method of the output HDR electro-optic curve, with the following values:

EOTF	**含义 Meaning**
Unsupported	HDMI 不支持发送 HDR 信息 HDMI doesn’t support to send HDR information
Not Defined	未定义 Not defined
Off	不发送 HDR 信息 Not to send HDR information
SDR	采用 SDR 曲线 Use SDR curve
ST2084	采用 ST2084 EOTF 曲线 Use ST2084 EOTF curve
HLG	采用 HLGEOTF 曲线 Use HLGEOTF curve

- (x0，y0)、(x1，y1)、(x2，y2)、(white x，white y)、max lum、min lum、max cll、maxfall 为静态 HDR 描述子信息，只有 EOTF 值为 SDR、ST2084、HLG 值时才会存在。
  (x0,y0), (x1,y1), (x2,y2), (white x,white y), max lum, min lum, max cll and maxfall are static description sub information of HDR, and only exist when EOTF value is SDR, ST2084 or HLG.

5.4 命令行设置分辨率 Use command line to set resolution

- 当前可用显示分辨率列表：
  Currently available display resolution list:

  cat /sys/devices/platform/display-subsystem/drm/card0/card0-HDMI-A-1/modes

- 查看当前显示分辨率：
  Check current display resolution:

  cat /sys/devices/platform/display-subsystem/drm/card0/card0-HDMI-A-1/mode

- 通过 persist.PROPERTY_TYPE.resolution.main 以及 persist.PROPERTY_TYPE.resolution.aux 设置主副屏分辨率，每次设置完更新 PROPERTY_TYPE.display.timeline(每次加 1)使分辨率生效，例子如下：
  Set main/secondary panel resolution through persist.PROPERTY_TYPE.resolution.main and persist.PROPERTY_TYPE.resolution.aux. After each setting, update PROPERTY_TYPE.display.timeline (add 1 each time) to make the resolution take effect. For example:

  Node: PROPERTY_TYPE define in <2.变量定义 Variable definition>

  - 设置 4k60：
    Set 4k60:

    setprop persist.PROPERTY_TYPE.resolution.main 3840x2160@60
    setprop PROPERTY_TYPE.display.timeline 1
  - 设置 1080p60：
    Set 1080p60:

    setprop persist.PROPERTY_TYPE.resolution.main 1920x1080@60
    setprop PROPERTY_TYPE.display.timeline 2
  - 设置 720P60：
    Set 720P60:

    setprop persist.PROPERTY_TYPE.resolution.main 1280x720@60

    setprop PROPERTY_TYPE.display.timeline 3
  - 设置 480P60：
    Set 480P60:

    setprop persist.PROPERTY_TYPE.resolution.main 720x480@60

    setprop PROPERTY_TYPE.display.timeline 4

6 Q&A

6.1 EDID 没有读到的情况下怎么设置默认分辨率 How to set the default resolution without EDID

```
Commit 727e0fe68d8f422698f4e257cb7c04f90b8692c0

Author: xuhuicong xhc@rock-chips.com

Date:   Tue Sep 26 17:32:56 2017 +0800



drm/edid: output common tv resolution and hdmi mode if no read the correct edid



Change-Id: Ib7379340e8c1d59382553d21b60165fe5fb371e8

Signed-off-by: xuhuicong xhc@rock-chips.com
```

在有上面的提交基础上，修改 def_modes 的值，对应的是 vic 值，如果 4 对应的是 edid_cea_modes 中的：

With the above commit existing, modify the value of def_modes, which is corresponding to vic value. For example, 4 corresponds to the value in edid_cea_modes as below:

```
/* 4 - 1280x720@60Hz */

{ DRM_MODE("1280x720", DRM_MODE_TYPE_DRIVER, 74250, 1280, 1390,

                1430, 1650, 0, 720, 725, 730, 750, 0,

                DRM_MODE_FLAG_PHSYNC | DRM_MODE_FLAG_PVSYNC),

  .vrefresh = 60, .picture_aspect_ratio = HDMI_PICTURE_ASPECT_16_9, },
```

6.2 如何设置 HDMI 旋转 How to set HDMI rotation

参考《Rockchip_双屏显示旋转方向调试文档_xxx.pdf》。

Refer to 《Rockchip_双屏显示旋转方向调试文档_xxx.pdf》.

6.3 如何设置 HDMI 缩放 How to set HDMI overscan

如果 HDMI 是主屏：setprop persist.PROPERTY_TYPE.overscan.main "overscan 100,100,100,100"

If HDMI is main display: setprop persist.PROPERTY_TYPE.overscan.main "overscan

100,100,100,100"

如果 HDMI 是副屏：setprop persist.PROPERTY_TYPE.overscan.aux "overscan 100,100,100,100"

If HDMI is secondary display: setprop persist.PROPERTY_TYPE.overscan.aux "overscan 100,100,100,100"

overscan 的 4 个参数分别指：left_margin，top_margin，right_margin，bottom_margin。

The four parameters of overscan are: left_margin, top_margin, right_margin, bottom_margin.

Node: PROPERTY_TYPE define in <2.变量定义 Variable definition>

6.4 如何输出特殊分辨率 How to output special resolution

```
--- a/drivers/gpu/drm/bridge/synopsys/dw-hdmi.c
+++ b/drivers/gpu/drm/bridge/synopsys/dw-hdmi.c
@@ -2497,7 +2497,7 @@ static int dw_hdmi_connector_get_modes(struct drm_connector *connector)
                                                connector);
        struct edid *edid;
        struct drm_display_mode *mode;
-       const u8 def_modes[6] = {4, 16, 31, 19, 17, 2};
+       const u8 def_modes[6] = {108, 16, 31, 19, 17, 2};
        struct hdr_static_metadata *metedata =
                        &connector->display_info.hdmi.hdr_panel_metadata;
        int i, ret = 0;
@@ -2506,6 +2506,7 @@ static int dw_hdmi_connector_get_modes(struct drm_connector *connector)
                return 0;

        edid = drm_get_edid(connector, hdmi->ddc);
+       edid = NULL;
        if (edid) {
                dev_dbg(hdmi->dev, "got edid: width[%d] x height[%d]\n",
                        edid->width_cm, edid->height_cm);
diff --git a/drivers/gpu/drm/drm_edid.c b/drivers/gpu/drm/drm_edid.c
index bfe6710..5d13766 100644
--- a/drivers/gpu/drm/drm_edid.c
+++ b/drivers/gpu/drm/drm_edid.c
@@ -1220,6 +1220,11 @@ static const struct drm_display_mode edid_cea_modes[] = {
                4104, 4400, 0, 2160, 2168, 2178, 2250, 0,
                DRM_MODE_FLAG_PHSYNC | DRM_MODE_FLAG_PVSYNC),
          .vrefresh = 60, .picture_aspect_ratio = HDMI_PICTURE_ASPECT_64_27, },
+       /* 108 - 800x1280p@60Hz */
+       { DRM_MODE("800x1280", DRM_MODE_TYPE_DRIVER, 76000, 800, 848,
+               880, 960, 0, 1280, 1300, 1304, 1314, 0,
+               DRM_MODE_FLAG_PHSYNC | DRM_MODE_FLAG_PVSYNC),
+         .vrefresh = 60, .picture_aspect_ratio = HDMI_PICTURE_ASPECT_64_27, },
 };

 /*
```

如上修改：

Modify as above:

1、 在 edid_cea_modes 数据的最后定义特殊的分辨率。

Define the special resolution at the end of edid_cea_modes data.

2、 把 def_mode 数组的第一个值改成特殊分辨率对应的 vic，我们上面定义的是 108.

Replace the first value of def_mode array with the corresponding vic of the special resolution, we define 108 as above.

3、 edid = NULL;强制把 edid 赋为 NULL，不管有没有读到 edid 都强制按 def_modes 来显示。

edid = NULL; forcedly set edid as NULL. Forcedly display according to def_modes no matter whether there is edid or not.