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

Rockchip_DRM_Panel_Porting_Guide

（技术部，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-228
	当前版本： **Current Version：**	V1.6
	作　　者： **Author:**	闭伟勇 Bi Weiyong
	完成日期： **Finish Date:**	2019-2-28
	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	闭伟勇 Bi Weiyong	2017-4-15	初始版本 Initial version release	
V1.1	黄家钗 Huang Jiachai	2017-4-17	加入 LVDS 屏配置说明 Add LVDS panel configuration description	
V1.2	闭伟勇 Bi Weiyong	2017-8-15	同步代码，更新 MIPI 章节。 Sync code, and update MIPI chapter	
V1.3	闭伟勇 Bi Weiyong	2017-12-9	更新文档，增加 Display Routing&FAQ 章节。 Add Display Routing&FAQ chapter	
V1.4	黄家钗 Huang Jiachai	2018-7-9	加入 MCU/CPU 屏配置说明 Add MCU/CPU panel configuration description	
V1.5	闭伟勇 Bi Weiyong	2018-8-30	更新 RGB 配置说明 Update RGB configuration description	
V1.6	闭伟勇 Bi Weiyong	2019-2-28	同步更新文档，增加 RK618 说明 Add RK618 description	

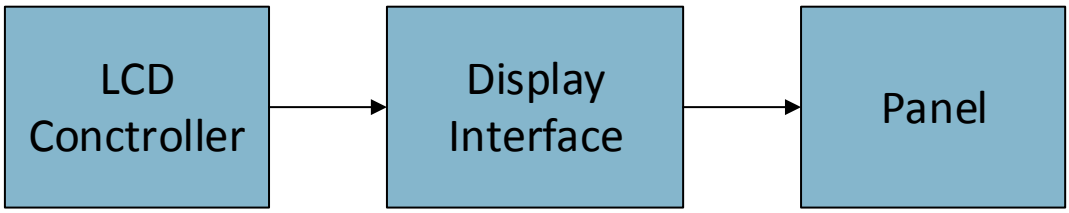
目 录 Contents

1 Introduction

本文档主要描述 Rockchip 平台基于 DRM 显示框架的底层显示配置，以及相关调试手段。

This document mainly describes Rockchip platform bottom layer display configuration and related debugging method based on DRM display framework.

1.1 Display Pipe



1） Rockchip 平台的 LCD Controller 称为 VOP（Video Output Processor），芯片中一般集成 1~2 个 VOP。只有支持两个 VOP 的芯片，才能支持双屏异显。在进行显示路由配置时，应该选择哪个 VOP 作为输入的依据主要是 VOP 支持的最大分辨率，以 RK3399 为例，RK3399 有两个 VOP，分别为 VOPB（4096x2160），VOPL（2560x1600），所以对于分辨率大于 2560x1600 的应用，只能选择 VOPB 作为输入。

Rockchip platform LCD Controller is called VOP (Video Output Processor), and generally there is 1~2 VOP integrated in the chip. Only the chip supporting two VOP can support dual display. When configuring the display route, to select which VOP as the input is mainly determined by the maximum resoltuion supported by VOP. Take RK3399 as example, RK3399 has two VOP: VOPB (4096x2160) and VOPL (2560x1600), so for the application with resolution higher than 2560x1600, you can only select VOPB as input.

2） Rockchip 平台的芯片集成丰富的显示接口，包括 HDMI/MIPI-DSI/RGB/LVDS/eDP/DP 等等，不同芯片可能包含其中的 N 个接口，不同的显示接口对应不同的驱动。

Rockchip platform chips integrate plenty of display interfaces, including HDMI/MIPI-DSI/RGB/LVDS/eDP/DP and so on. Different chips may include different interfaces and different display interfaces correspond to different drivers.

3） 对于 Embedded Connection 的 Panel，一般不需要支持 HPD，需要手动配置上电时序以及显示相关信息，所以需要单独的驱动。而对于 box-to-box 的 Monitor，一般支持 HPD，可以自动识别显示相关信息，所以不需要单独的驱动。

Panel embedded connection generally doesn’t need to support HPD, and need to manually configure power up sequence and display related information, so it requires separate driver. While the box-to-box Monitor generally supports HPD, and can automatically recognize display related information, so it doesn’t require separate driver.

2 Panel

2.1 Documentation and Source Code

Kernel (develop-4.4)

drivers/gpu/drm/panel/panel-simple.c

Documentation/devicetree/bindings/display/panel/simple-panel.txt

U-Boot (next-dev)

drivers/video/drm/rockchip_panel.c

U-Boot (rkdevelop)

drivers/video/rockchip_panel.c

drivers/video/rockchip_dsi_panel.c

2.2 DT Bindings

1) simple-panel（LVDS/RGB/eDP）

```
/ {
        panel {
                compatible = "simple-panel";
                backlight = <&backlight>;
                power-supply = <&vcc_lcd>;
                enable-gpios = <&gpio1 13 GPIO_ACTIVE_HIGH>;
                prepare-delay-ms = <20>;
                enable-delay-ms = <20>;

                display-timings {
                        native-mode = <&timing0>;

                        timing0: timing0 {
                                clock-frequency = <200000000>;
                                hactive = <1536>;
                                vactive = <2048>;
                                hfront-porch = <12>;
                                hsync-len = <16>;
                                hback-porch = <48>;
                                vfront-porch = <8>;
                                vsync-len = <4>;
                                vback-porch = <8>;
                                hsync-active = <0>;
                                vsync-active = <0>;
                                de-active = <0>;
                                pixelclk-active = <0>;
                        };
                };

                port {
                        panel_in_edp: endpoint {
                                remote-endpoint = <&edp_out_panel>;
                        };
                };
        };
};
```

2) simple-panel-dsi (MIPI-DSI)

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

        panel@0 {
                compatible = "pvo,p101nwwbp-01g", "simple-panel-dsi";
                reg = <0>;
                power-supply = <&vcc3v3_lcd>;
                reset-gpios = <&gpio0 RK_PA2 GPIO_ACTIVE_LOW>;
                backlight = <&backlight>;
                prepare-delay-ms = <20>;
                reset-delay-ms = <20>;
                init-delay-ms = <20>;
                enable-delay-ms = <20>;
                disable-delay-ms = <20>;
                unprepare-delay-ms = <20>;

                width-mm = <135>;
                height-mm = <216>;

                dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST |
                              MIPI_DSI_MODE_LPM | MIPI_DSI_MODE_EOT_PACKET)>;
                dsi,format = <MIPI_DSI_FMT_RGB888>;
                dsi,lanes = <4>;
```

```
                panel-init-sequence = [
                        15 00 02 E0 00
                        ...
                        15 00 02 E7 06
                        15 80 01 11
                        15 16 01 29
                ];

                panel-exit-sequence = [
                        05 00 01 28
                        05 00 01 10
                ];

                display-timings {
                        native-mode = <&timing0>;

                        timing0: timing0 {
                                clock-frequency = <68500000>;
                                hactive = <800>;
                                hfront-porch = <16>;
                                hsync-len = <16>;
                                hback-porch = <48>;
                                vactive = <1280>;
                                vfront-porch = <8>;
                                vsync-len = <4>;
                                vback-porch = <4>;
                                hsync-active = <0>;
                                vsync-active = <0>;
                                de-active = <0>;
                                pixelclk-active = <0>;
                        };
                };
        };
};
```

这里只列出通用配置，其他与特定显示接口相关的配置在各个显示接口章节中单独说明。

Here only list the commonly used configurations, and other configurations related with the specific display interfaces will be introduced in the chapter of each display interface.

Property	**Value**	**Comment**
compatible	simple-panel or simple-panel-dsi	
backlight		背光节点引用 Phandle of the backlight device
power-supply		可选，Regulator 配置。 Optional, configure for Regulator.
reset-gpios		可选，Reset GPIO 配置。 Optional, configure for Reset GPIO
enable-gpios		可选，Enable GPIO 配置。 Optional, configure to Enable GPIO.
prepare-delay-ms		可选，具体时序参考屏驱动。 Optional, refer to panel driver for detailed timing
reset-delay-ms		可选，具体时序参考屏驱动。 Optional, refer to panel driver for detailed timing
init-delay-ms		可选，具体时序参考屏驱动。 Optional, refer to panel driver for detailed timing
enable-delay-ms		可选，具体时序参考屏驱动。 Optional, refer to panel driver for detailed timing
unprepare-delay-ms		可选，具体时序参考屏驱动。 Optional, refer to panel driver for detailed timing
disable-delay-ms		可选，具体时序参考屏驱动。 Optional, refer to panel driver for detailed timing
display-timings		LCD 时序参数，按屏规格书填写。 LCD timing parameter, fill in according to panel spec
width-mm		LCD 物理宽度，按屏规格书填写。 LCD physical width, fill in according to panel spec
height-mm		LCD 物理高度，按屏规格书填写。

		LCD physical height, fill in according to panel spec

2.3 常见问题 Common issues

1. reset 脚一般为 LOW 有效，驱动最后会把 reset 脚拉高，如果 reset 脚为 HIGH 有效，驱动最后会把 reset 脚拉低。enable 脚一般为 HIGH 有效，驱动最后会把 enable 脚拉高，如果 enable 脚为 LOW 有效，驱动最后会把 enable 脚拉低。

   The reset pin is generally LOW effective, and the driver will pull up the reset pin at last. If the reset pin is HIGH effective, the driver will pull down the reset pin at last. The enable pin is generally HIGH effective, and the driver will pull up the enable pin at last. If the enable pin is LOW effective, the driver will pull down the enable pin at last.
2. 如果 drm 驱动一直 bind 失败，返回-517（-EPROBE_DEFER），往往是由于 panel 驱动 probe 失败引起的，这个时候就需要检查 panel 相关的配置，比如 reset/enable 是否与其他模块配置冲突。

   If drm driver always fails to bind, returning -517 (-EPROBE_DEFER), usually it is caused by panel driver probe failure, in this case, you need to check panel related configuration, such as whether reset/enable conflict with other module configuration or not.
3. Simple-panel 只是一个通用驱动，只能满足一般需求，如果代码不支持，可以考虑对现有驱动进行扩展或者单独写一个特定的驱动。

   Simple-panel is just a generic driver which can only meet normal requirement. If the code doesn't support, consider to extend the existing driver or separately develop a specific driver.

3 MIPI-DSI

1） rk3128/rk3326/px30/rk3368 (1~4lanes, 1Gbps per lane)

2） rk3288/rk3399 (1~8lanes, 1.5Gbps per lane)

3.1 Documentation and Source Code

Kernel (develop-4.4)

drivers/gpu/drm/rockchip/dw-mipi-dsi.c

drivers/phy/rockchip/phy-rockchip-inno-video-combo-phy.c

Documentation/devicetree/bindings/display/rockchip/dw_mipi_dsi_rockchip.txt

Documentation/devicetree/bindings/phy/phy-rockchip-inno-video-combo-phy.txt

U-Boot (next-dev)

drivers/video/drm/dw_mipi_dsi.c

drivers/video/drm/inno_video_combo_phy.c

U-boot (rkdevelop)

drivers/video/rockchip-dw-mipi-dsi.c

drivers/video/rockchip-inno-mipi-dphy.c

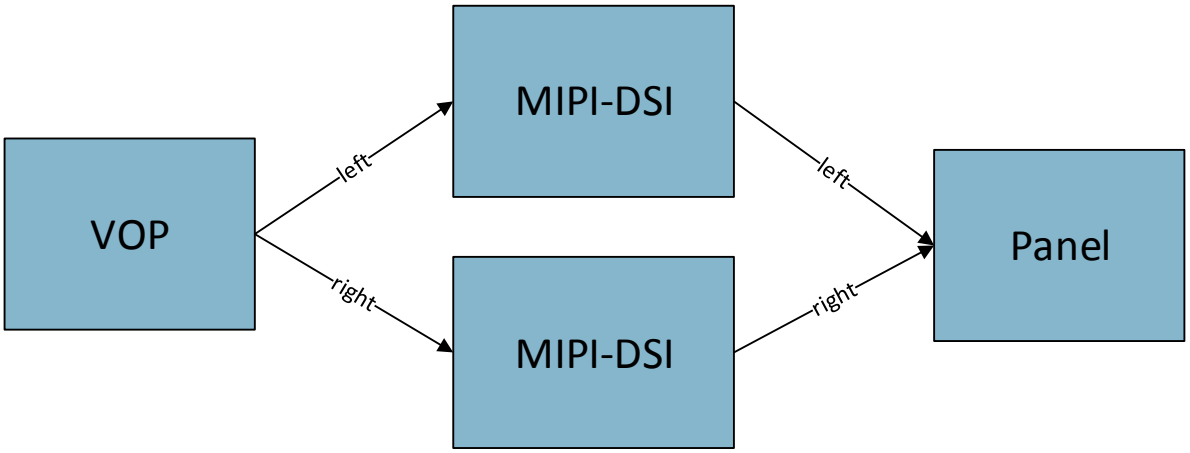
3.2 DT Bindings

3.2.1 Host

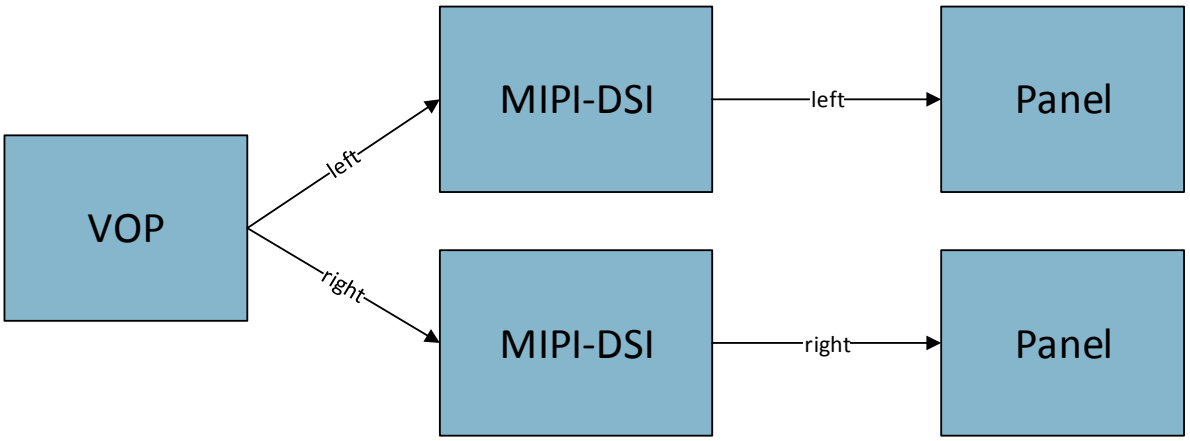
1） Single-channel


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

2） Dual-channel (RK3288/RK3399)



① 标准的 dual-channel 接口 MIPI 屏；
MIPI panel with standard dual-channel interface.



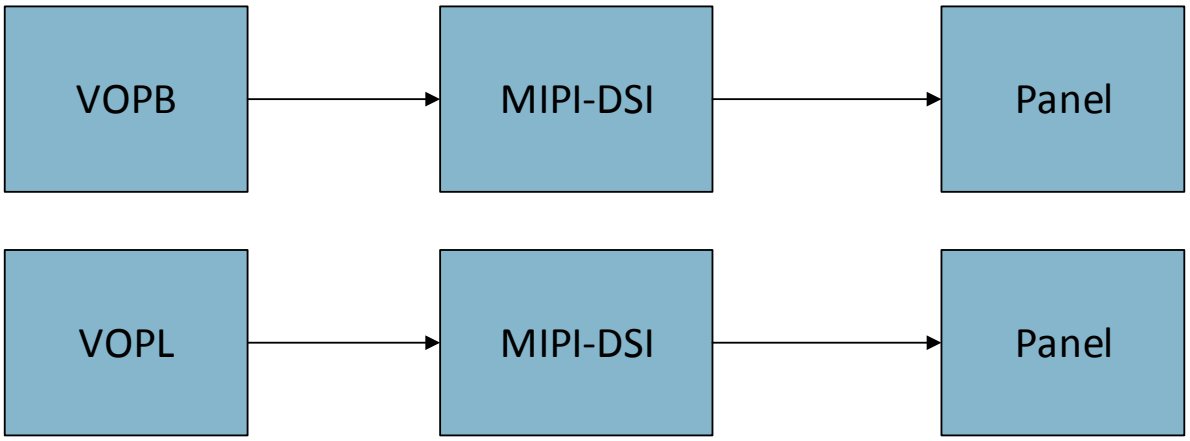
② 分别接一样的屏，组合成 dual-channel 接口 MIPI 屏，panel0 显示左半屏，panel1 显示右半屏。
Separately connect two same panels to form the MIPI panel with dual-channel interface, panel0

displays the left screen, and panel1 displays the right screen.

```
&dsi {
        rockchip,dual-channel = <&dsi1>;
        status = "okay";
};
```

```
&dsi1 {
        status = "okay";
};
```

3） Dual-link (RK3399)



因为 MIPI_PHY_TX0 和 MIPI_PHY_TXRX 共用一个 PLL，所以 Dual-link 模式下，如果 panel0 和 panel1 是一样的屏，那么在同一个 lane-rate 下都能正常显示；但如果 panel0 和 panel1 是不一样的屏，只有两个屏都能在同一个 lane-rate 下都能正常显示，才能支持这种模式。

Because MIPI_PHY_TX0 and MIPI_PHY_TXRX use one PLL, in Dual-link mode, if panel0 and panel1 are the same panel, both of them can display normally with the same lane-rate. But if panel0 and panel1 are different, this mode is supported only when the two panels can be displayed normally with the same lane-rate.

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

```
&dsi1 {
        status = "okay";
};
```

Property	Value	Comment
rockchip,lane-rate	0~1500	指定 DATA_LANE 的速率，单位为 mbps/lane，CLK_LANE 的频率为该值的一半，比如配置为 400Mbps，相应的 CLK 频率为 200MHz。如果没有配置该属性，驱动会自动计算 lane-rate。 Specify the rate of DATA_LANE, the unit is

		mbps/lane, CLK_LANE frequency is half of the value, for example, if it is configured as 400Mbps, the corresponding CLK frequency is 200MHz. The driver will automatically calculate lane-rate if this property is not configured.
rockchip,dual-channel		对于 Dual-channel mode，该属性必须配置。 For Dual-channel mode, this property must be configured.

3.2.2 PHY

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

NOTE: 对于有单独 PHY 节点的芯片(rk3128/px30/rk3326/rk3368)，需要使能该节点。

Note: For the chips with separate PHY node (rk3128/px30/rk3326/rk3368), need to enable this node.

3.2.3 VOP Routing

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

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

NOTE：对于有两个 VOP 的芯片，需要选择其一，如果有打开 LOGO，route_dsi 的 connect 属性也要指定为同一个 VOP。

Note: For the chip with two VOP, need to select one of them. If LOGO is enabled, need to specify the same VOP for connect property of route_dsi.

3.2.4 Logo

```
&route_dsi {
        connect = <&vopb_out_dsi>;
        status = "okay";
};
```

3.2.5 Panel

1）Single-channel

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

        panel@0 {
                compatible = "simple-panel-dsi";
                reg = <0>;
                power-supply = <&vcc_lcd>;
                backlight = <&backlight>;
                reset-gpios = <&gpio3 13 GPIO_ACTIVE_LOW>;
                enable-gpios = <&gpio2 27 GPIO_ACTIVE_HIGH>;

                dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST |
                              MIPI_DSI_MODE_LPM | MIPI_DSI_MODE_EOT_PACKET)>;
                dsi,format = <MIPI_DSI_FMT_RGB888>;
                dsi,lanes = <4>;

                prepare-delay-ms = <20>;
                reset-delay-ms = <20>;
                init-delay-ms = <20>;
                enable-delay-ms = <20>;
                disable-delay-ms = <20>;
                unprepare-delay-ms = <20>;
                panel-init-sequence = [
                        39 00 04 b9 ff 83 94
                        ...
                        15 00 02 df 8e
                        ...
                        05 78 01 11
                        05 14 01 29
                ];

                panel-exit-sequence = [
                        05 00 01 28
                        05 78 01 10
                ];
                display-timings {
                        native-mode = <&timing0>;

                        timing0: timing0 {
                                clock-frequency = <74000000>;
                                hactive = <800>;
                                hfront-porch = <68>;
                                hsync-len = <18>;
                                hback-porch = <68>;
                                vactive = <1280>;
                                vfront-porch = <6>;
                                vsync-len = <4>;
                                vback-porch = <6>;
                                hsync-active = <0>;
                                vsync-active = <0>;
                                de-active = <0>;
                                pixelclk-active = <0>;
                        };
                };
        };
};
```

2) Dual-channel

子模式①和 Single-channel 的主要区别是 dsi,lanes 的值大于 4。

The main difference between sub mode ① and Single-channel is the value of dsi,lanes is bigger than 4.

子模式②和 Single-channel 的主要区别是 dsi,lanes，clock-frequency，hactive，hfront-porch，hsync-len，hback-portch 在单个 panel 的基础上 x2。

The main difference between sub mode ② and Single-channel is dsi,lanes, clock-frequency, hactive, hfront-porch, hsync-len, hback-portch are 2 times that of single panel.

Property	Value	Comment
compatible	simple-panel-dsi	
reg	0	virtual channel
dsi,flags	(MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST | MIPI_DSI_MODE_EOT_PACKET | MIPI_DSI_MODE_LPM)	MIPI_DSI_MODE_VIDEO, MIPI_DSI_MODE_VIDEO_BURST, 表示 Video Burst Mode。 Means Video Burst Mode. MIPI_DSI_MODE_LPM 表示默认在 LP 模式下发送初始化序列。 MIPI_DSI_MODE_LPM means to send initialization sequence in LP mode by default. MIPI_DSI_MODE_EOT_PACKET 表示关闭 EOTP 特性。 MIPI_DSI_MODE_EOT_PACKET means to disable EOTP feature.
dsi,format	MIPI_DSI_FMT_RGB888	Pixel Format
dsi,lanes	4	Lane Number（1 ~ 8），**大于 4 表示为 Dual-channel MIPI-DSI Panel**。 Lane Number (1~8), bigger than 4 means it is Dual-channel MIPI-DSI Panel.
panel-init-sequence	…	屏的上电初始化序列，具体参数配置方式参考下文说明。 Panel power up initialization sequence. Refer to the following description for the specific parameter configurations.
panel-exit-sequence	…	屏的下电初始化序列，具体参数配置方式参考下文说明。 Panel power down initialization sequence. Refer to the following description for the specific parameter configurations.

3.3 Command

```
panel-init-sequence = [
        39 00 04 b9 ff 83 94
        ...
        15 00 02 df 8e
        ...
        05 78 01 11
        05 14 01 29
];
```

格式说明：头部 3 个字节（16 进制），分别代表 Data Type，Delay，Payload Length。

Format description: 3 bytes of the header (hexadecimal), separately represent Data Type, Delay, Payload Length.

从第四个字节开始的数据代表长度为 Length 的 Payload。

The data starting from the fourth byte represents Payload with the length of Length.

第一条命令的解析如下：

The first command is parsed as below:

```
39 00 04 b9 ff 83 94
```

Data Type：0x39 (DCS Long Write)

Delay：0x00 (0 ms)

Payload Length：0x04 (4 Bytes)

Payload：0xb9 0xff 0x83 0x94

最后一条命令的解析如下：

The last command is parsed as below:

```
05 14 01 29
```

Data Type：0x05 (DCS Short Write, no parameters)

Delay：0x14 (20 ms)

Payload Length：0x01 (1 Bytes)

Payload：0x29

3.3.1 Data Type

Table 16 Data Types for Processor-sourced Packets

Data Type, hex	Data Type, binary	Description	Packet Size
0x01	00 0001	Sync Event, V Sync Start	Short
0x11	01 0001	Sync Event, V Sync End	Short
0x21	10 0001	Sync Event, H Sync Start	Short
0x31	11 0001	Sync Event, H Sync End	Short
0x08	00 1000	End of Transmission packet (EoTp)	Short
0x02	00 0010	Color Mode (CM) Off Command	Short
0x12	01 0010	Color Mode (CM) On Command	Short
0x22	10 0010	Shut Down Peripheral Command	Short
0x32	11 0010	Turn On Peripheral Command	Short
0x03	00 0011	Generic Short WRITE, no parameters	Short
0x13	01 0011	Generic Short WRITE, 1 parameter	Short
0x23	10 0011	Generic Short WRITE, 2 parameters	Short
0x04	00 0100	Generic READ, no parameters	Short
0x14	01 0100	Generic READ, 1 parameter	Short
0x24	10 0100	Generic READ, 2 parameters	Short
0x05	00 0101	DCS Short WRITE, no parameters	Short
0x15	01 0101	DCS Short WRITE, 1 parameter	Short
0x06	00 0110	DCS READ, no parameters	Short
0x37	11 0111	Set Maximum Return Packet Size	Short
0x09	00 1001	Null Packet, no data	Long
0x19	01 1001	Blanking Packet, no data	Long
0x29	10 1001	Generic Long Write	Long
0x39	11 1001	DCS Long Write/write_LUT Command Packet	Long
0x0C	00 1100	Loosely Packed Pixel Stream, 20-bit YCbCr, 4:2:2 Format	Long
0x1C	01 1100	Packed Pixel Stream, 24-bit YCbCr, 4:2:2 Format	Long
0x2C	10 1100	Packed Pixel Stream, 16-bit YCbCr, 4:2:2 Format	Long
0x0D	00 1101	Packed Pixel Stream, 30-bit RGB, 10-10-10 Format	Long
0x1D	01 1101	Packed Pixel Stream, 36-bit RGB, 12-12-12 Format	Long

Data Type, hex	Data Type, binary	Description	Packet Size
0x3D	11 1101	Packed Pixel Stream, 12-bit YCbCr, 4:2:0 Format	Long
0x0E	00 1110	Packed Pixel Stream, 16-bit RGB, 5-6-5 Format	Long
0x1E	01 1110	Packed Pixel Stream, 18-bit RGB, 6-6-6 Format	Long
0x2E	10 1110	Loosely Packed Pixel Stream, 18-bit RGB, 6-6-6 Format	Long
0x3E	11 1110	Packed Pixel Stream, 24-bit RGB, 8-8-8 Format	Long
0xX0 and 0xXF, unspecified	XX 0000 XX 1111	DO NOT USE All unspecified codes are reserved	

① DCS Write

0x05	00 0101	DCS Short WRITE, no parameters	Short
0x15	01 0101	DCS Short WRITE, 1 parameter	Short

0x39	11 1001	DCS Long Write/write_LUT Command Packet	Long

DCS packet 包括一个字节的 dcs 命令，以及 n 个字节的 parameters。

DCS packet includes dcs command with one byte, and parameters with n bytes.

如果 n < 2，将以 Short Packet 的形式对 Payload 进行打包。n = 0，表示只发送 dcs 命令，不带参数，Data Type 为 0x05；n = 1，表示发送 dcs 命令，带一个参数，Data Type 为 0x15。

If n < 2, it will package Payload in the form of Short Packet. n=0 means only to send dcs command without parameter, Data Type is 0x05. n=1 means to send dcs command with one parameter, Data Type is 0x15.

如果 n >= 2，将以 Long Packet 的形式对 Payload 进行打包。此时发送 dcs 命令，带 n 个参数，Data Type 为 0x39。

If n >= 2, it will package Payload in the form of Long Packet. In this case, it will send dcs command along with n parameters, Data Type is 0x39.

② Generic Write

0x03	00 0011	Generic Short WRITE, no parameters	Short
0x13	01 0011	Generic Short WRITE, 1 parameter	Short
0x23	10 0011	Generic Short WRITE, 2 parameters	Short

0x29	10 1001	Generic Long Write	Long

Generic Packet 包括 n 个字节的 parameters。

Generic Packet includes parameters with n bytes.

如果 n < 3，将以 Short Packet 的形式对 Payload 进行打包。n = 0，表示 no parameters，Data Type 为 0x03；n = 1，表示 1 parameter，Data Type 为 0x13；n = 2，表示 2 parameters，Data Type 为 0x23。

If n < 3, it will package Payload in the form of Short Packet. n=0 means no parameters, Data Type is 0x03. n=1 means 1 parameter, Data Type is 0x13. n=2 means 2 parameters, Data Type is 0x23.

如果 n >= 3，将以 Long Packet 的形式进行对 Payload 打包，表示 n parameters，Data Type 为 0x29。

If n >= 3, it will package Payload in the form of Long Packet, which means n parameters and Data Type is 0x29.

3.3.2 Delay

表示当前 Packet 发送完成之后，需要延时多少 ms，再开始发送下一条命令。

It means after sending current Packet, how many ms latency are required before sending next command.

3.3.3 Payload Length

表示 Packet 的有效负载长度。

It means the effective load length of Packet.

3.3.4 Payload

表示 Packet 的有效负载，长度为 Payload Length。

It means the effective load of Packet, and the length is Payload Length.

3.3.5 Example

(B) On sequence

sequence	DataType (hex)	index (hex)	parameters #	parameters (hex)	description	comment
SLEEP MODE						
↓						
DCDC_EN L->H					DCDC_EN L->H (VSP,VSN on)	
wait 20ms						
command	05	01	-	-	soft reset	
wait 5ms						
command	23	B0	1	00	MCAP	
command	29	B3	1	04	Interface setting	
			2	08		
			3	00		
			4	22		
			5	00		
command	29	B4	1	0C	Interface ID setting	
command	29	B6	1	3A	DSI control	
			2	D3		
command	15	51	1	E6	write display brightness	
command	15	53	1	2C	write control display	
command	15	3A	1	77	set pixel format	
command	39	2A	1	00	set column address	
			2	00		
			3	04		
			4	AF		
command	39	2B	1	00	set page address	
			2	00		
			3	07		
			4	7F		
send image	39	2C/3C			write memory / write memory continue	
command	05	11	-	-	exit sleep mode	
wait 120ms						
command	05	29	-	-	set display on	
wait min 0ms						
LED_EN L->H					LED_EN L->H	
↓						
NORMAL MODE						

```
panel-init-sequence = [
        05 05 01 01
        23 00 02 b0 00
        23 00 02 d6 01
        29 00 06 b3 14 08 00 22 00
        29 00 02 b4 0c
        29 00 03 b9 3a c3
        15 00 02 51 e6
        15 00 02 53 2c
        15 00 02 3a 77
        39 00 05 2a 00 00 04 af
        39 00 05 2b 00 00 07 7f
        05 78 01 29
        05 00 01 11
];
```

(C) Off sequence

sequence	DataTyp (hex)	index (hex)	parameters #	(hex)	description	comment
NORMAL MODE						
command	05	28	-	-	set display off	
wait 20ms						
command	05	10	-	-	enter sleep mode	
wait 80ms						
DCDC_EN H->L					DCDC_EN H->L (VSP,VSN off)	
wait 20ms						
SLEEP MODE						

```
panel-exit-sequence = [
        05 14 01 28
        05 50 01 10
];
```

3.4 常见问题 Common issues

1. 如何对 MIPI-DSI 外设进行读写操作。

   How to do read/write operation on MIPI-DSI peripheral.

   drivers/gpu/drm/drm_mipi_dsi.c

   drivers/gpu/drm/drm_mipi_dsi.h

   提供了对 MIPI-DSI 外设通信的相关 API。

   Provide relative API communicating with MIPI-DSI peripheral.

```
ssize_t mipi_dsi_dcs_write_buffer(struct mipi_dsi_device *dsi,
                  const void *data, size_t len);
ssize_t mipi_dsi_dcs_write(struct mipi_dsi_device *dsi, u8 cmd,
                const void *data, size_t len);
ssize_t mipi_dsi_dcs_read(struct mipi_dsi_device *dsi, u8 cmd, void *data,
              size_t len);
```

```
ssize_t mipi_dsi_generic_write(struct mipi_dsi_device *dsi, const void *payload,
                   size_t size);
ssize_t mipi_dsi_generic_read(struct mipi_dsi_device *dsi, const void *params,
                  size_t num_params, void *data, size_t size);
```

2. 如何判断 MIPI-DSI 外设有正常工作？

   How to judge whethe MIPI-DSI peripheral works normally or not?

   可以对 MIPI-DSI 外设进行读操作，如果通信正常，说明外设有正常工作。比如支持 DCS 标准 MIPI_DCS_GET_POWER_MODE（0x0A）命令的外设，可以通过读取 power_mode 来判断。

   You can do read operation on MIPI-DSI peripheral. If the communication is normal, it means the peripheral works normally. For example, the peripheral supporting the command of DCS standard MIPI_DCS_GET_POWER_MODE(0x0A) can be judged by reading power_mode.

```
--- a/drivers/gpu/drm/panel/panel-simple.c
+++ b/drivers/gpu/drm/panel/panel-simple.c
@@ -627,6 +627,7 @@ static int panel_simple_prepare(struct drm_panel *panel)
 {
        struct panel_simple *p = to_panel_simple(panel);
        int err;
+       u8 mode;

        if (p->prepared)
                return 0;
@@ -655,6 +656,11 @@ static int panel_simple_prepare(struct drm_panel *panel)
        if (p->desc && p->desc->delay.init)
                panel_simple_sleep(p->desc->delay.init);

+       if (p->dsi) {
+               err = mipi_dsi_dcs_get_power_mode(p->dsi, &mode);
+               dev_info(p->dev, "err=%d, mode=%02x\n", err, mode);
+       }
+
        if (p->on_cmds) {
                if (p->dsi)
                        err = panel_simple_dsi_send_cmds(p, p->on_cmds);
@@ -664,6 +670,11 @@ static int panel_simple_prepare(struct drm_panel *panel)
                        dev_err(p->dev, "failed to send on cmds\n");
        }

+       if (p->dsi) {
+               err = mipi_dsi_dcs_get_power_mode(p->dsi, &mode);
+               dev_info(p->dev, "err=%d, mode=%02x\n", err, mode);
+       }
+
        p->prepared = true;

        return 0;
```

```
[   29.339267] dw-mipi-dsi ff450000.dsi: final DSI-Link bandwidth: 460 x 4 Mbps
[   29.407062] panel-simple-dsi ff450000.dsi.0: err=0, mode=08
[   29.566445] panel-simple-dsi ff450000.dsi.0: err=0, mode=9c
```

在外设硬件复位之后，读取 power_mode，err=0 说明通信正常，从而判断设备供电和复位是正常的，mode=0x08 说明设备目前是 OFF 状态。在发送初始化命令之后，读取 power_mode，mode=0x9c，说明设备目前是 ON 状态。

Read power_mode after the peripheral is reset through hardware, err=0 means the communication is normal, then you can judge the power supply and reset of the device are normal. mode=0x08 means current device is in OFF state. Read power_mode after sending the initialization command, mode=0x9c means current device is in ON state.

3. 如何支持 DCS 背光。

How to support DCS backlight

如果设备支持标准 DCS 背光调节，并且代码已经实现 DCS 背光驱动，可以使能 DCS 背光功能。

If the device supports standard DCS backlight adjustment, and the code already implements DCS backlight driver, you can enable DCS backlight function.

```
1. 删除dsi-panel节点下的backlight属性。
2.
bivvy@rk-intel-1:~/rk3288/hardware/rockchip/liblights$ git diff
diff --git a/lights.cpp b/lights.cpp
index eebcd8f..55e3900 100644
--- a/lights.cpp
+++ b/lights.cpp
@@ -34,7 +34,7 @@
 #define LOGE(fmt,args...) ALOGE(fmt,##args)

 /******************************************************************************/
-#define BACKLIGHT_PATH  "/sys/class/backlight/rk28_bl/brightness"
+#define BACKLIGHT_PATH  "/sys/class/backlight/dcs-backlight/brightness"
 #define BACKLIGHT_PATH1 "/sys/class/backlight/backlight/brightness" // for kernel 4.4
 #define BUTTON_LED_PATH "sys/class/leds/rk29_key_led/brightness"
 #define BATTERY_LED_PATH "sys/class/leds/battery_led/brightness"
3.
bivvy@rk-intel-1:~/rk3288/device/rockchip/common$ git diff
diff --git a/init.rk30board.rc b/init.rk30board.rc
index b7ae3e1..f031e73 100755
--- a/init.rk30board.rc
+++ b/init.rk30board.rc
@@ -139,7 +139,7 @@ on boot
     write /proc/sys/net/core/wmem_max 1048576

     # backlight
-    chown system system /sys/class/backlight/rk28_bl/brightness
+    chown system system /sys/class/backlight/dcs-backlight/brightness
     chown system system /sys/class/backlight/backlight/brightness
```

4. 如何使能 EOTP（EoT packet）特性。

   How to enable EOTP (EoT packet) feature

   DSI 驱动根据 MIPI_DSI_MODE_EOT_PACKET 来判断是否使能 EOTP，如果 flags 没有配置 MIPI_DSI_MODE_EOT_PACKET，使能 EOTP 特性，如果 flags 有配置 MIPI_DSI_MODE_EOT_PACKET，关闭 EOTP 特性。

   DSI driver will determine whether to enable EOTP or not according to MIPI_DSI_MODE_EOT_PACKET. If flags doesn't configure MIPI_DSI_MODE_EOT_PACKET, enable EOTP feature, and if flags configures MIPI_DSI_MODE_EOT_PACKET, disable EOTP feature.

```
--- a/arch/arm64/boot/dts/rockchip/px30-evb-ddr3-v10.dts
+++ b/arch/arm64/boot/dts/rockchip/px30-evb-ddr3-v10.dts
@@ -209,7 +209,7 @@
                height-mm = <121>;

                dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST |
-                             MIPI_DSI_MODE_LPM | MIPI_DSI_MODE_EOT_PACKET)>;
+                             MIPI_DSI_MODE_LPM)>;
                dsi,format = <MIPI_DSI_FMT_RGB888>;
                dsi,lanes = <4>;
```

5. 如何发送 data_type 为 MIPI_DSI_SHUTDOWN_PERIPHERAL（0x22）和 MIPI_DSI_TURN_ON_PERIPHERAL（0x32）的 packet。

   How to send the packet which data_type is MIPI_DSI_SHUTDOWN_PERIPHERAL (0x22) and MIPI_DSI_TURN_ON_PERIPHERAL (0x32)

```
--- a/drivers/gpu/drm/panel/panel-simple.c
+++ b/drivers/gpu/drm/panel/panel-simple.c
@@ -580,6 +580,16 @@ static int panel_simple_disable(struct drm_panel *panel)
        if (p->desc && p->desc->delay.disable)
                panel_simple_sleep(p->desc->delay.disable);

+       if (p->dsi) {
+               p->dsi->mode_flags &= ~MIPI_DSI_MODE_LPM;
+               err = mipi_dsi_shutdown_peripheral(p->dsi);
+               if (err < 0) {
+                       dev_err(p->dev, "failed to shutdown peripheral\n");
+                       return err;
+               }
+               p->dsi->mode_flags |= MIPI_DSI_MODE_LPM;
+       }
+
        if (p->cmd_type == CMD_TYPE_MCU) {
                err = panel_simple_mcu_send_cmds(p, p->off_cmds);
                if (err)
@@ -685,6 +695,16 @@ static int panel_simple_enable(struct drm_panel *panel)
        if (p->desc && p->desc->delay.enable)
                panel_simple_sleep(p->desc->delay.enable);

+       if (p->dsi) {
+               p->dsi->mode_flags &= ~MIPI_DSI_MODE_LPM;
+               err = mipi_dsi_turn_on_peripheral(p->dsi);
+               if (err < 0) {
+                       dev_err(p->dev, "failed to turn on peripheral\n");
+                       return err;
+               }
+               p->dsi->mode_flags |= MIPI_DSI_MODE_LPM;
+       }
+
        backlight_enable(p->backlight);
```

6. 如何使能非连续时钟？

How to enable the non-continuous clock?

DSI驱动根据MIPI_DSI_CLOCK_NON_CONTINUOUS来判断是否使能非连续时钟，如果flags没有配置MIPI_DSI_CLOCK_NON_CONTINUOUS，表示连续时钟，如果flags有配置MIPI_DSI_MODE_EOT_PACKET，表示非连续时钟。

DSI driver will determine whether to enable the non-continuous clock according to MIPI_DSI_CLOCK_NON_CONTINUOUS. If flags doesn't configure MIPI_DSI_CLOCK_NON_CONTINUOUS, it represents continuous clock. If flags configures MIPI_DSI_MODE_EOT_PACKET, it represents non-continuous clock.

```
--- a/arch/arm64/boot/dts/rockchip/px30-evb-ddr3-v10.dtsi
+++ b/arch/arm64/boot/dts/rockchip/px30-evb-ddr3-v10.dtsi
@@ -205,7 +205,8 @@
                height-mm = <121>;

                dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST |
-                            MIPI_DSI_MODE_LPM | MIPI_DSI_MODE_EOT_PACKET)>;
+                            MIPI_DSI_MODE_LPM | MIPI_DSI_MODE_EOT_PACKET |
+                            MIPI_DSI_CLOCK_NON_CONTINUOUS)>;
                dsi,format = <MIPI_DSI_FMT_RGB888>;
                dsi,lanes = <4>;
```

4 eDP

1）rk3288/rk3368 (1/2/4 lanes, 1.62Gbps/2.7Gbps)

2）rk3399 (1/2/4 lanes, 1.62Gbps/2.7Gbps/5.4Gbps)

4.1 Documentation and Source Code

Kernel (develop-4.4)：

drivers/gpu/drm/bridge/analogix/analogix_dp_core.c

drivers/gpu/drm/bridge/analogix/analogix_dp_reg.c

drivers/gpu/drm/rockchip/analogix_dp-rockchip.c

drivers/phy/rockchip/phy-rockchip-dp.c

Documentation/devicetree/bindings/display/bridge/analogix_dp.txt

Documentation/devicetree/bindings/display/rockchip/analogix_dp-rockchip.txt

Documentation/devicetree/bindings/phy/rockchip-dp-phy.txt

U-Boot (next-dev)：

drivers/video/drm/analogix_dp.c

drivers/video/drm/analogix_dp_reg.c

U-Boot (rkdevelop)

drivers/video/rockchip_analogix_dp.c

drivers/video/rockchip_analogix_dp_reg.c

4.2 DT Bindings

4.2.1 Host

1）Embedded Connection

```
&edp {
        force-hpd;
        status = "okay";

        ports {
                port@1 {
                        reg = <1>;

                        edp_out_panel: endpoint {
                                remote-endpoint = <&panel_in_edp>;
                        };
                };
        };
};
```

2）Box-to-box Connection


```
&edp {
        hpd-gpios = <&gpio4 23 GPIO_ACTIVE_HIGH>;
        status = "okay";
};
```

Property	Value	Comment
force-hpd		对于 Embedded Connection，一般不需要 HPD 功能，需要加上该属性。 Embedded Connection generally doesn't need HPD function, and needs to add this property.
hpd-gpios		对于 Box-to-box Connection，一般需要 HPD 功能，需要配置该属性。 Box-to-box Connection generally doesn't need HPD function, and needs to configure this property.

4.2.2 PHY

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

NOTE: 对于有单独 PHY 节点的芯片，需要使能该节点。

Note: For the chip with separate PHY node, need to enable this node.

4.2.3 VOP Routing

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

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

NOTE：对于有两个 VOP 的芯片，需要选择其一，如果有打开 LOGO，route_edp 的 connect 属性也要指定为同一个 VOP。

Note: For the chip with two VOP, need to select one of them. If LOGO is enabled, need to specify the same VOP for connect property of route_edp.

4.2.4 Logo

```
&route_edp {
        connect = <&vopb_out_edp>;
        status = "okay";
};
```

4.2.5 Panel

```
/ {
        panel {
                compatible = "simple-panel";
                backlight = <&backlight>;
                power-supply = <&vcc_lcd>;
                enable-gpios = <&gpio1 13 GPIO_ACTIVE_HIGH>;
                prepare-delay-ms = <20>;
                enable-delay-ms = <20>;
                bus-format = <MEDIA_BUS_FMT_RGB888_1X24>;
                bpc = <8>;

                display-timings {
                        native-mode = <&timing0>;

                        timing0: timing0 {
                                clock-frequency = <200000000>;
                                hactive = <1536>;
                                vactive = <2048>;
                                hfront-porch = <12>;
                                hsync-len = <16>;
                                hback-porch = <48>;
                                vfront-porch = <8>;
                                vsync-len = <4>;
                                vback-porch = <8>;
                                hsync-active = <0>;
                                vsync-active = <0>;
                                de-active = <0>;
                                pixelclk-active = <0>;
                        };
                };

                ports {
                        panel_in_edp: endpoint {
                                remote-endpoint = <&edp_out_panel>;
                        };
                };
        };
};
```

① 属性说明 Property description

Property	Value	Comment
bpc	6 or 8	Bit pixel component
bus-format	MEDIA_BUS_FMT_RGB666_1X18 MEDIA_BUS_FMT_RGB888_1X24	分别对应 6bit 和 8bit 屏 Separately corresponding to 6bit and 8bit panel

4.3 常见问题 Common issues

1) Aux Transaction fail

[33.319392] rockchip-dp ff970000.edp: Rx Max Link Rate is abnormal :c0 !

[33.319543] rockchip-dp ff970000.edp: Rx Max Lane count is abnormal :0 !

[33.322377] rockchip-dp ff970000.edp: AUX CH command reply failed!

Aux 通信失败一般是由于设备端没有正常工作或者屏线问题，导致没有应答，应当检查设备端的供电是否正常以及排线。

Generally Aux Transaction failure is caused by the device end work abnormality or the panel wire issue which will lead to no response. Need to check whether the power supply of the device end is normal and the panel wire.

5 LVDS

1） rk3128/px30/rk3326/rk3368（single-channel）

2） rk3288 (single-channel/dual-channel)

5.1 Documentation and Source Code

Kernel (develop-4.4)：

drivers/gpu/drm/rockchip/rockchip_lvds.c

drivers/phy/rockchip/phy-rockchip-inno-video-combo-phy.c

drivers/phy/rockchip/phy-rockchip-inno-video-phy.c

Documentation/devicetree/bindings/display/rockchip/rockchip-lvds.txt

Documentation/devicetree/bindings/phy/phy-rockchip-inno-video-combo-phy.txt

Documentation/devicetree/bindings/phy/phy-rockchip-inno-video-phy.txt

U-Boot (next-dev)：

drivers/video/drm/rockchip_lvds.c

drivers/video/drm/inno_video_combo_phy.c

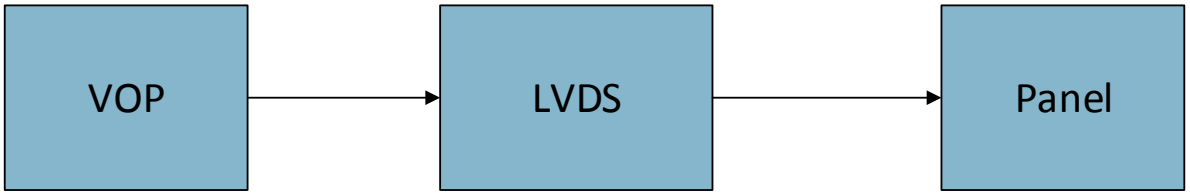
drivers/video/drm/inno_video_phy.c

U-Boot (rkdevelop)：

drivers/video/rockchip_lvds.c

5.2 DT Bindings

5.2.1 Host



1） Single-channel

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

        ports {
                port@1 {
                        reg = <1>;

                        lvds_out_panel: endpoint {
                                remote-endpoint = <&panel_in_lvds>;
                        };
                };
        };
};
```

2） Dual-channel

```
&lvds {
        dual-channel;
        status = "okay";

        ports {
                port@1 {
                        reg = <1>;

                        lvds_out_panel: endpoint {
                                remote-endpoint = <&panel_in_lvds>;
                        };
                };
        };
};
```

Property	Value	Comment
dual-channel		使能 Dual-channel 模式 Enable Dual-channel mode
rockchip,data-swap		在 Dual-channel 模式下，对两个通道的奇偶像素进行互换。 In Dual-channel mode, exchange the even and odd pixels of the two channels.

5.2.2 PHY

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

5.2.3 VOP Routing

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

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

NOTE：对于有两个 VOP 的芯片，需要选择其一，如果有打开 LOGO，route_lvds 的 connect 属性也要指定为同一个 VOP。

Note: For the chip with two VOP, need to select one of them. If LOGO is enabled, need to specify the same VOP for connect property of route_lvds.

5.2.4 Logo

```
&route_lvds {
        connect = <&vopb_out_lvds>;
        status = "okay";
};
```

5.2.5 Panel

```
/ {
        panel {
                compatible = "samsung,lsl070nl01", "simple-panel";
                backlight = <&backlight>;
                power-supply = <&vcc3v3_lcd>;
                enable-delay-ms = <20>;
                prepare-delay-ms = <20>;
                unprepare-delay-ms = <20>;
                disable-delay-ms = <20>;
                bus-format = <MEDIA_BUS_FMT_RGB888_1X7X4_SPWG>;

                width-mm = <217>;
                height-mm = <136>;

                display-timings {
                        native-mode = <&timing0>;

                        timing0: timing0 {
                                clock-frequency = <49500000>;
                                hactive = <1024>;
                                vactive = <600>;
                                hback-porch = <90>;
                                hfront-porch = <90>;
                                vback-porch = <10>;
                                vfront-porch = <10>;
                                hsync-len = <90>;
                                vsync-len = <10>;
                                hsync-active = <0>;
                                vsync-active = <0>;
                                de-active = <0>;
                                pixelclk-active = <0>;
                        };
                };

                port {
                        panel_in_lvds: endpoint {
                                remote-endpoint = <&lvds_out_panel>;
                        };
                };
        };
};
```

① 属性说明 Property description

Property	Value	Comment
bus-format	MEDIA_BUS_FMT_RGB666_1X7X3_SPWG MEDIA_BUS_FMT_RGB888_1X7X4_SP	LVDS 信号的数据映射方式，分别对应"vesa-18"，"vesa-24"，"jeida-24"，"jeida-18"。具体映射关系参考 data mapping

	WG MEDIA_BUS_FMT_RGB888_1X7X4_JEIDA MEDIA_BUS_FMT_RGB666_1X7X3_JEIDA	说明。 Data mapping method of LVDS signal, separately corresponding to "vesa-18", "vesa-24", "jeida-24", "jeida-18". Refer to data mapping description for the detailed mapping relationship.

5.3 Data Mapping

5.3.1 6 bit output mode

		VESA_6BIT	JEIDA_6BIT
	TX0	R0	R2
	TX1	R1	R3
	TX2	R2	R4
Y	TX3	R3	R5
0	TX4	R4	R6
	TX6	R5	R7
	TX7	G0	G2
	TX8	G1	G3
	TX9	G2	G4
Y	TX12	G3	G5
1	TX13	G4	G6
	TX14	G5	G7
	TX15	B0	B2
	TX18	B1	B3
	TX19	B2	B4
	TX20	B3	B5
	TX21	B4	B6
Y	TX22	B5	B7
2	TX24	HSYNC	HSYNC
	TX25	VSYNC	VSYNC
	TX26	ENABLE	ENABLE
	TX27	GND	GND
	TX5	GND	GND
	TX10	GND	GND
Y	TX11	GND	GND
3	TX16	GND	GND
	TX17	GND	GND
	TX23	RSVD	RSVD

5.3.2 8 bit output mode

		VESA_8BIT	JEIDA_8BIT
	TX0	R0	R2
	TX1	R1	R3
	TX2	R2	R4
Y	TX3	R3	R5
0	TX4	R4	R6
	TX6	R5	R7
	TX7	G0	G2
	TX8	G1	G3
	TX9	G2	G4
Y	TX12	G3	G5
1	TX13	G4	G6
	TX14	G5	G7
	TX15	B0	B2
	TX18	B1	B3
	TX19	B2	B4
	TX20	B3	B5
	TX21	B4	B6
Y	TX22	B5	B7
2	TX24	HSYNC	HSYNC
	TX25	VSYNC	VSYNC
	TX26	ENABLE	ENABLE
	TX27	R6	R0
	TX5	R7	R1
	TX10	G6	G0
Y	TX11	G7	G1
3	TX16	B6	B0
	TX17	B7	B1
	TX23	RSVD	RSVD

6 RGB

1）rk3128/rk3326/px30/rk3288/rk3368

6.1 Documentation and Source Code

Kernel (develop-4.4)：

drivers/gpu/drm/rockchip/rockchip_rgb.c

drivers/phy/rockchip/phy-rockchip-inno-video-combo-phy.c

drivers/phy/rockchip/phy-rockchip-inno-video-phy.c

Documentation/devicetree/bindings/display/rockchip/rockchip-rgb.txt

Documentation/devicetree/bindings/phy/phy-rockchip-inno-video-combo-phy.txt

Documentation/devicetree/bindings/phy/phy-rockchip-inno-video-phy.txt

U-Boot (next-dev):

drivers/video/drm/rockchip_rgb.c

drivers/video/drm/inno_video_combo_phy.c

drivers/video/drm/inno_video_phy.c

U-Boot (rkdevelop):

drivers/video/rockchip_lvds.c

6.2 DT Bingdings

6.2.1 Host

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

        ports {
                port@1 {
                        reg = <1>;

                        rgb_out_panel: endpoint {
                                remote-endpoint = <&panel_in_rgb>;
                        };
                };
        };
};
```

6.2.2 PHY

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

6.2.3 VOP Routing

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

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

NOTE：对于有两个 VOP 的芯片，需要选择其一，如果有打开 LOGO，route_rgb 的 connect 属性也要指定为同一个 VOP。

Note: For the chip with two VOP, need to select one of them. If LOGO is enabled, need to specify

the same VOP for connect property of route_rgb.

6.2.4 Logo

```
&route_rgb {
        connect = <&vopb_out_rgb>;
        status = "okay";
};
```

6.2.5 Panel

```
/ {
        panel {
                compatible ="simple-panel";
                enable-gpios = <&gpio0 RK_PB5 GPIO_ACTIVE_LOW>;
                reset-gpios = <&gpio3 RK_PB7 GPIO_ACTIVE_LOW>;
                bus-format = <MEDIA_BUS_FMT_RGB666_1X18>;

                display-timings {
                        native-mode = <&timing0>;

                        timing0: timing0 {
                                clock-frequency = <51200000>;
                                hactive = <1024>;
                                vactive = <600>;
                                hback-porch = <100>;
                                hfront-porch = <120>;
                                vback-porch = <10>;
                                vfront-porch = <15>;
                                hsync-len = <100>;
                                vsync-len = <10>;
                                hsync-active = <0>;
                                vsync-active = <0>;
                                de-active = <0>;
                                pixelclk-active = <0>;
                        };
                };

                port {
                        panel_in_rgb: endpoint {
                                remote-endpoint = <&rgb_out_panel>;
                        };
                };
        };
};
```

Property	Value	Comment

bus-format	MEDIA_BUS_FMT_RBG888_1X24 MEDIA_BUS_FMT_RGB666_1X24_CPADHI MEDIA_BUS_FMT_RGB666_1X18	RGB 信号的输出关系，分别对应"OUT_P888"，"OUT_D888_P666"，"OUT_P666"。具体参考 data mapping 说明。 The output relationship of RGB signal, separately corresponding to "OUT_P888", "OUT_D888_P666" and "OUT_P666". Refer to data mapping for more details.

6.3 Data Mapping

Display mode	*RGB Parallel 24-bit*	*RGB Parallel 18-bit*	*RGB Parallel 18-bit*	*RGB Parallel 16-bit*	*RGB Parallel 16-bit*	*ITU656 Mode0*	*ITU656 Mode1*	*ITU656 Mode2*	*MCU mode*
Screen_face	OUT_P888	OUT_D888_P666	OUT_P666	OUT_D888_P565	OUT_P565	OUT_S888/OUT_S888DUMY	OUT_S888/OUT_S888DUMY	OUT_S888/OUT_S888DUMY	OUT_P888
DCLK	DCLK	DCLK	DCLK	DCLK	DCLK	DCLK	DCLK	DCLK	RS
VSYNC	VSYNC	VSYNC	VSYNC	VSYNC	VSYNC				CS
HSYNC	HSYNC	HSYNC	HSYNC	HSYNC	HSYNC				WEN
DEN	DEN	DEN	DEN	DEN	DEN				REN
DATA	DATA[23:0]	DATA[23:18] DATA[15:10] DATA[7:2]	DATA[17:0]	DATA[23:19] DATA[15:10] DATA[7:3]	DATA[15:0]	DATA[7:0]	DATA[15:8]	DATA[14:7]	
D23	R7	R5	-	R4	-	-	-	-	D23
D22	R6	R4	-	R3	-	-	-	-	D22
D21	R5	R3	-	R2	-	-	-	-	D21
D20	R4	R2	-	R1	-	-	-	-	D20

D19	R3	R1	-	R0	-	-	-	-	D19
D18	R2	R0	-	-	-	-	-	-	D18
D17	R1	-	R5	-	-	-	-	-	D17
D16	R0	-	R4	-	-	-	-	-	D16
D15	G7	G5	R3	G5	R4	-	D7	-	D15
D14	G6	G4	R2	G4	R3	-	D6	D7	D14
D13	G5	G3	R1	G3	R2	-	D5	D6	D13
D12	G4	G2	R0	G2	R1	-	D4	D5	D12
D11	G3	G1	G5	G1	R0	-	D3	D4	D11
D10	G2	G0	G4	G0	G5	-	D2	D3	D10
D9	G1	-	G3	-	G4	-	D1	D2	D9
D8	G0	-	G2	-	G3	-	D0	D1	D8
D7	B7	B5	G1	B4	G2	D7	-	D0	D7
D6	B6	B4	G0	B3	G1	D6	-	-	D6
D5	B5	B3	B5	B2	G0	D5	-	-	D5
D4	B4	B2	B4	B1	B4	D4	-	-	D4
D3	B3	B1	B3	B0	B3	D3	-	-	D3
D2	B2	B0	B2	-	B2	D2	-	-	D2
D1	B1	-	B1	-	B1	D1	-	-	D1
D0	B0	-	B0	-	B0	D0	-	-	D0

7 DP Alt Mode

7.1 Documentation and Source Code

Kernel (develop-4.4)：

drivers/gpu/drm/rockchip/cdn-dp-core.c

drivers/gpu/drm/rockchip/cdn-dp-reg.c

drivers/gpu/drm/rockchip/cdn-dp-link-training.c

drivers/phy/rockchip/phy-rockchip-typec.c

Documentation/devicetree/bindings/display/rockchip/cdn-dp-rockchip.txt

Documentation/devicetree/bindings/phy/phy-rockchip-typec.txt

7.2 DT Bindings

7.2.1 DP_TX

```
&cdn_dp {
        extcon = <&fusb0>;
        phys = <&tcphy0_dp>;
        status = "okay";
};
```

7.2.2 USB Type-C PHY

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

7.2.3 USB PD

```
&i2c4 {
        status = "okay";

        fusb0: fusb30x@22 {
                compatible = "fairchild,fusb302";
                reg = <0x22>;
                pinctrl-names = "default";
                pinctrl-0 = <&fusb0_int>;
                int-n-gpios = <&gpio1 2 GPIO_ACTIVE_HIGH>;
                vbus-5v-gpios = <&gpio2 0 GPIO_ACTIVE_HIGH>;
                status = "okay";
        };
};
```

7.2.4 VOP Routing

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

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

NOTE：对于有两个 VOP 的芯片，需要选择其一，一般选择 VOPB，因为需要支持 4K。

Note: For the chip with two VOP, need to select one of them. Generally select VOPB, because need

to support 4K.

8 RK618

RK616/RK618 是 Rockchip 平台的配套显示转换芯片，该芯片具有如下特性：

RK616/RK618 are the display conversion chips of Rockchip platform. The chips have the following features:

① 两个 RGB 输入接口（LCD0,LCD1）。如果分别把 LCD0 和 LCD1 接到不同的源，可以实现双屏异显；如果只接 LCD0，可以实现单屏/双屏同显，此时可以不使用 LCD1 或者将 LCD1 复用为 RGB 输出接口。

Two RGB input interfaces (LCD0, LCD1). If separately connect LCD0 and LCD1 to different sources, you can implement dual display. If only connect LCD0, you can implement single display/dual panel same display, in this case, LCD1 can be unused or reused as RGB output interface.

② 一个 RGB 输出口，与 LVDS 输出口复用。

One RGB output interface, reused with LVDS output interface.

③ 一个 LVDS 输出口，与 RGB 输出口复用，RK618 支持 dual-channel，RK616 不支持 dual-channel。

One LVDS output interface, reused with RGB output interface. RK618 supports dual-channel and RK616 doesn’t support dual-channel.

④ 一个 MIPI-DSI 输出口，RK618 支持该接口，RK616 不支持该接口。

One MIPI-DSI output interface, RK618 supports this interface while RK616 doesn’t support this interface.

⑤ 一个 HDMI 输出口。

One HDMI output interface.

8.1 Documentation and Source Code

Kernel (develop-4.4)：

drivers/mfd/rk618.c

drivers/clk/rockchip/rk618/clk-rk618.c

drivers/gpu/drm/rockchip/rk618/rk618_lvds.c

drivers/gpu/drm/rockchip/rk618/rk618_rgb.c

drivers/gpu/drm/rockchip/rk618/rk618_scaler.c

drivers/gpu/drm/rockchip/rk618/rk618_vif.c

drivers/gpu/drm/rockchip/rk618/rk618_hdmi.c

drivers/gpu/drm/rockchip/rk618/rk618_dither.c

drivers/gpu/drm/rockchip/rk618/rk618_dsi.c

Documentation/devicetree/bindings/mfd/rk618.txt

Documentation/devicetree/bindings/clock/rockchip,rk618-cru.txt

Documentation/devicetree/bindings/display/rockchip/rockchip,rk618.txt

U-Boot (next-dev):

drivers/video/drm/rk618.c

drivers/video/drm/rk618_lvds.c

8.2 DT Bindings

8.2.1 RK618

```
&i2c0 {
        status = "okay";

        rk618: rk618@50 {
                compatible = "rockchip,rk618";
                reg = <0x50>;
                pinctrl-names = "default";
                pinctrl-0 = <&i2s1_2ch_mclk>;
                clocks = <&cru SCLK_I2S1_OUT>;
                clock-names = "clkin";
                assigned-clocks = <&cru SCLK_I2S1_OUT>;
                assigned-clock-rates = <12000000>;
                reset-gpios = <&gpio0 RK_PA0 GPIO_ACTIVE_LOW>;
                status = "okay";
        };
};
```

Property	Value	Comment
pinctrl-names pinctrl-0		输入时钟 CLKIN 引脚复用配置 The input clock CLKIN pin reuse configuration
clocks clock-names		输入时钟 CLKIN 引用 The input clock CLKIN phandle
assigned-clocks assigned-clock-rates		指定 CLKIN 初始频率为 12MHz Specify CLKIN initial frequency as 12MHz
reset-gpios		Reset 引脚配置，可选。 Reset pin configuration, optional.
enable-gpios		Enable 引脚配置，可选。 Enable pin configuration, optional.
power-supply		Regulator 配置，可选。 Regulator configuration, optional.

8.2.2 CRU

```
&rk618 {
        status = "okay";

        clock: cru {
                compatible = "rockchip,rk618-cru";
                clocks = <&cru SCLK_I2S1_OUT>, <&cru DCLK_VOPL>;
                clock-names = "clkin", "lcdc0_dclkp";
                assigned-clocks = <&clock SCALER_PLLIN_CLK>,
                                  <&clock VIF_PLLIN_CLK>,
                                  <&clock SCALER_CLK>,
                                  <&clock VIF0_PRE_CLK>,
                                  <&clock CODEC_CLK>,
                                  <&clock DITHER_CLK>;
                assigned-clock-parents = <&cru SCLK_I2S1_OUT>,
                                         <&clock LCDC0_CLK>,
                                         <&clock SCALER_PLL_CLK>,
                                         <&clock VIF_PLL_CLK>,
                                         <&cru SCLK_I2S1_OUT>,
                                         <&clock VIF0_CLK>;
                #clock-cells = <1>;
                status = "okay";
        };
};
```

Property	Value	Comment
clocks clock-names		输入时钟 CLKIN 引用，以及 LCDC0_DCLKP 引用 The input clock CLKIN phandle, and LCDC0_DCLKP phandle
assigned-clocks assigned-clock-rates		指定内部时钟默认父时钟 Specify the default father clock of internal clock

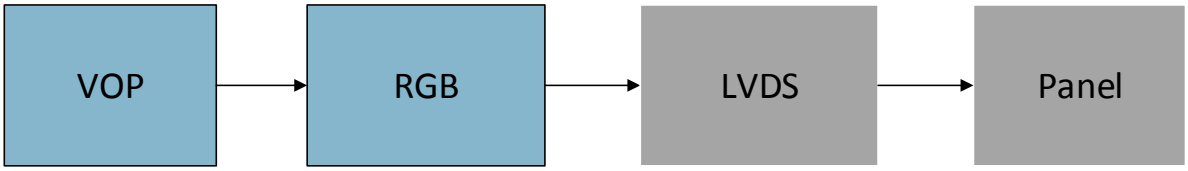
8.2.3 HDMI

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

        ports {
                port@1 {
                        reg = <1>;

                        rgb_out_hdmi: endpoint {
                                remote-endpoint = <&hdmi_in_rgb>;
                        };
                };
        };
};

&rgb_in_vopb {
        status = "disabled";
};

&rgb_in_vopl {
        status = "okay";
};

&route_rgb {
        connect = <&vopl_out_rgb>;
        status = "disabled";
};
```

```
&rk618 {
        status = "okay";

        hdmi {
                compatible = "rockchip,rk618-hdmi";
                clocks = <&clock HDMI_CLK>;
                clock-names = "hdmi";
                assigned-clocks = <&clock HDMI_CLK>;
                assigned-clock-parents = <&clock VIF0_CLK>;
                interrupt-parent = <&gpio2>;
                interrupts = <12 IRQ_TYPE_LEVEL_HIGH>;
                status = "okay";

                ports {
                        #address-cells = <1>;
                        #size-cells = <0>;

                        port@0 {
                                reg = <0>;

                                hdmi_in_rgb: endpoint {
                                        remote-endpoint = <&rgb_out_hdmi>;
                                };
                        };
                };
        };
};
```

Property	Value	Comment
interrupt-parent interrupts		INTERUPT 引脚配置 INTERUPT pin configuration

8.2.4 LVDS



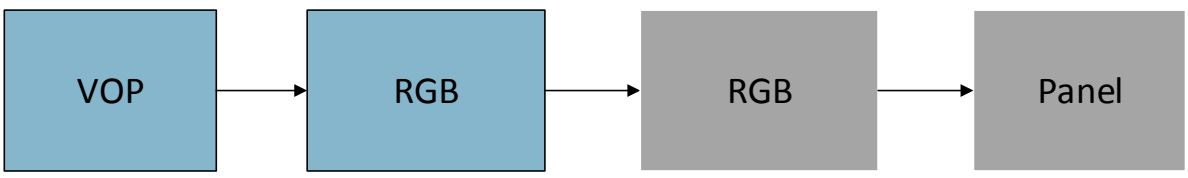
```
&rgb {
        status = "okay";

        ports {
                port@1 {
                        reg = <1>;

                        rgb_out_lvds: endpoint {
                                remote-endpoint = <&lvds_in_rgb>;
                        };
                };
        };
};

&rgb_in_vopb {
        status = "disabled";
};

&rgb_in_vopl {
        status = "okay";
};

&route_rgb {
        connect = <&vopl_out_rgb>;
        status = "disabled";
};
```

```
&rk618 {
        status = "okay";

        lvds {
                compatible = "rockchip,rk618-lvds";
                clocks = <&clock LVDS_CLK>;
                clock-names = "lvds";
                status = "okay";

                ports {
                        #address-cells = <1>;
                        #size-cells = <0>;

                        port@0 {
                                reg = <0>;

                                lvds_in_rgb: endpoint {
                                        remote-endpoint = <&rgb_out_lvds>;
                                };
                        };

                        port@1 {
                                reg = <1>;

                                lvds_out_panel: endpoint {
                                        remote-endpoint = <&panel_in_lvds>;
                                };
                        };
                };
        };
};
```

Property	Value	Comment
dual-channel		使能 dual-channel 模式，RK616 不支持 dual-channel，RK618 支持 dual-channel。 Enable dual-channel mode, RK616 doesn't support dual-channel, and RK618 supports dual-channel.

```
/ {
        panel {
                compatible = "chunghwa,claa101wh31-cw", "simple-panel";
                backlight = <&backlight>;
                power-supply = <&vcc3v3_lcd>;
                enable-gpios = <&gpio0 RK_PB5 GPIO_ACTIVE_LOW>;
                prepare-delay-ms = <120>;
                enable-delay-ms = <120>;
                disable-delay-ms = <120>;
                unprepare-delay-ms = <120>;
                bus-format = <MEDIA_BUS_FMT_RGB888_1X7X4_SPWG>;
                width-mm = <231>;
                height-mm = <154>;

                display-timings {
                        native-mode = <&timing1>;

                        timing1: timing1 {
                                clock-frequency = <72000000>;
                                hactive = <1280>;
                                vactive = <800>;
                                hback-porch = <60>;
                                hfront-porch = <60>;
                                vback-porch = <16>;
                                vfront-porch = <16>;
                                hsync-len = <40>;
                                vsync-len = <6>;
                                hsync-active = <0>;
                                vsync-active = <0>;
                                de-active = <0>;
                                pixelclk-active = <0>;
                        };
                };

                port {
                        panel_in_lvds: endpoint {
                                remote-endpoint = <&lvds_out_panel>;
                        };
                };
        };
};
```

8.2.5 RGB

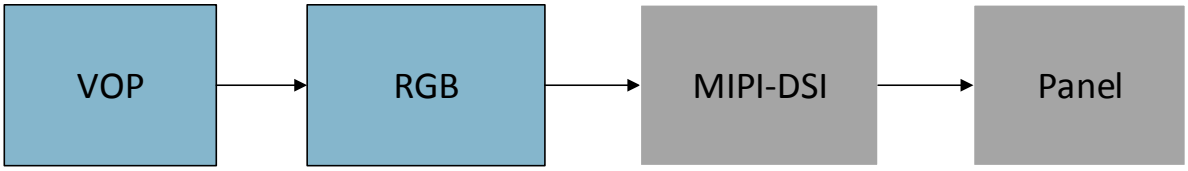


```
&rgb {
        status = "okay";

        ports {
                port@1 {
                        reg = <1>;

                        rgb_out_rgb: endpoint {
                                remote-endpoint = <&rgb_in_rgb>;
                        };
                };
        };
};

&rgb_in_vopb {
        status = "disabled";
};

&rgb_in_vopl {
        status = "okay";
};

&route_rgb {
        connect = <&vopl_out_rgb>;
        status = "disabled";
};
```

```
&rk618 {
        rgb {
                compatible = "rockchip,rk618-rgb";
                clocks = <&clock RGB_CLK>;
                clock-names = "rgb";
                status = "okay";

                ports {
                        #address-cells = <1>;
                        #size-cells = <0>;

                        port@0 {
                                reg = <0>;

                                rgb_in_rgb: endpoint {
                                        remote-endpoint = <&rgb_out_rgb>;
                                };
                        };

                        port@1 {
                                reg = <1>;

                                rgb_out_panel: endpoint {
                                        remote-endpoint = <&panel_in_rgb>;
                                };
                        };
                };
        };
};
```

```
/ {
        panel {
                compatible = "simple-panel";
                backlight = <&backlight>;
                enable-gpios = <&gpio0 RK_PB5 GPIO_ACTIVE_HIGH>;
                enable-delay-ms = <20>;
                prepare-delay-ms = <20>;
                unprepare-delay-ms = <20>;
                disable-delay-ms = <20>;
                bus-format = <MEDIA_BUS_FMT_RGB666_1X24_CPADHI>;

                width-mm = <154>;
                height-mm = <86>;

                display-timings {
                        native-mode = <&timing0>;

                        timing0: timing0 {
                                clock-frequency = <49000000>;
                                hactive = <1024>;
                                vactive = <600>;
                                hback-porch = <90>;
                                hfront-porch = <90>;
                                vback-porch = <10>;
                                vfront-porch = <10>;
                                hsync-len = <90>;
                                vsync-len = <10>;
                                hsync-active = <0>;
                                vsync-active = <0>;
                                de-active = <0>;
                                pixelclk-active = <0>;
                        };
                };

                port {
                        panel_in_rgb: endpoint {
                                remote-endpoint = <&rgb_out_panel>;
                        };
                };
        };
};
```

8.2.6 MIPI-DSI



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

        ports {
                port@1 {
                        reg = <1>;

                        rgb_out_dsi: endpoint {
                                remote-endpoint = <&dsi_in_rgb>;
                        };
                };
        };
};

&rgb_in_vopl {
        status = "okay";
};

&rgb_in_vopb {
        status = "disabled";
};

&route_rgb {
        connect = <&vopl_out_rgb>;
        status = "disabled";
};
```

```
&rk618 {
        dsi {
                compatible = "rockchip,rk618-dsi";
                clocks = <&clock MIPI_CLK>;
                clock-names = "dsi";
                #address-cells = <1>;
                #size-cells = <0>;
                status = "okay";

                ports {
                        #address-cells = <1>;
                        #size-cells = <0>;

                        port@0 {
                                reg = <0>;

                                dsi_in_rgb: endpoint {
                                        remote-endpoint = <&rgb_out_dsi>;
                                };
                        };
                };
};
```

```
panel@0 {
        compatible = "simple-panel-dsi";
        reg = <0>;
        power-supply = <&vcc3v3_lcd>;
        backlight = <&backlight>;
        reset-gpios = <&gpio2 RK_PA0 GPIO_ACTIVE_LOW>;
        prepare-delay-ms = <20>;
        reset-delay-ms = <20>;
        init-delay-ms = <20>;
        enable-delay-ms = <120>;
        disable-delay-ms = <20>;
        unprepare-delay-ms = <20>;

        width-mm = <95>;
        height-mm = <151>;

        dsi,flags = <(MIPI_DSI_MODE_VIDEO |
                      MIPI_DSI_MODE_VIDEO_BURST |
                      MIPI_DSI_MODE_LPM |
                      MIPI_DSI_MODE_EOT_PACKET)>;
        dsi,format = <MIPI_DSI_FMT_RGB888>;
        dsi,lanes = <4>;

        panel-init-sequence = [
                15 00 02 b0 00
                15 00 02 d6 01
                39 00 06 b3 14 08 00 22 00
                15 00 02 b4 0c
                15 00 02 DE 00
                39 00 03 b6 3a d3
                15 00 02 51 E0
                15 00 02 53 04
                15 00 02 3a 77
                15 00 02 35 01
                39 00 05 2A 00 00 04 AF
                39 00 05 2B 00 00 07 7F
                05 96 01 29
                05 14 01 11
        ];
```

```
                        panel-exit-sequence = [
                                05 00 01 28
                                05 00 01 10
                        ];

                        display-timings {
                                native-mode = <&timing1>;

                                timing1: timing1 {
                                        clock-frequency = <156000000>;
                                        hactive = <1200>;
                                        vactive = <1920>;
                                        hback-porch = <60>;
                                        hfront-porch = <80>;
                                        vback-porch = <4>;
                                        vfront-porch = <4>;
                                        hsync-len = <10>;
                                        vsync-len = <1>;
                                        hsync-active = <0>;
                                        vsync-active = <0>;
                                        de-active = <0>;
                                        pixelclk-active = <0>;
                                };
                        };
                };
        };
};
```

8.2.7 Clone Mode

该模式需要选用横屏的 LCD，因为输入源的分辨率是 HDMI 分辨率，SCALER 模块会对这个源根据 LCD 分辨率进行缩放，如果 LCD 是竖屏，会造成显示效果不佳。

This mode requires the horizontal LCD. Because the resolution of the input source is HDMI resolution, and SCALER module will zoom in/out this input source according to LCD resolution, if LCD is vertical, the display effect will not be good.

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

        ports {
                port@1 {
                        reg = <1>;

                        rgb_out_vif: endpoint {
                                remote-endpoint = <&vif_in_rgb>;
                        };
                };
        };
};

&rgb_in_vopb {
        status = "disabled";
};

&rgb_in_vopl {
        status = "okay";
};

&route_rgb {
        connect = <&vopl_out_rgb>;
        status = "disabled";
};
```

```
&rk618 {
        status = "okay";

        hdmi {
                compatible = "rockchip,rk618-hdmi";
                clocks = <&clock HDMI_CLK>;
                clock-names = "hdmi";
                assigned-clocks = <&clock HDMI_CLK>;
                assigned-clock-parents = <&clock VIF0_CLK>;
                interrupt-parent = <&gpio2>;
                interrupts = <12 IRQ_TYPE_LEVEL_HIGH>;
                status = "okay";

                ports {
                        #address-cells = <1>;
                        #size-cells = <0>;

                        port@0 {
                                reg = <0>;

                                hdmi_in_vif: endpoint {
                                        remote-endpoint = <&vif_out_hdmi>;
                                };
                        };

                        port@1 {
                                reg = <1>;

                                hdmi_out_scaler: endpoint {
                                        remote-endpoint = <&scaler_in_hdmi>;
                                };
                        };
                };
        };
```

```
lvds {
        compatible = "rockchip,rk618-lvds";
        clocks = <&clock LVDS_CLK>;
        clock-names = "lvds";
        status = "okay";

        ports {
                #address-cells = <1>;
                #size-cells = <0>;

                port@0 {
                        reg = <0>;

                        lvds_in_scaler: endpoint {
                                remote-endpoint = <&scaler_out_lvds>;
                        };
                };

                port@1 {
                        reg = <1>;

                        lvds_out_panel: endpoint {
                                remote-endpoint = <&panel_in_lvds>;
                        };
                };
        };
};
```

```
scaler {
        compatible = "rockchip,rk618-scaler";
        clocks = <&clock SCALER_CLK>, <&clock VIF0_CLK>,
                 <&clock DITHER_CLK>;
        clock-names = "scaler", "vif", "dither";
        status = "okay";

        ports {
                #address-cells = <1>;
                #size-cells = <0>;

                port@0 {
                        reg = <0>;

                        scaler_in_hdmi: endpoint {
                                remote-endpoint = <&hdmi_out_scaler>;
                        };
                };

                port@1 {
                        reg = <1>;

                        scaler_out_lvds: endpoint {
                                remote-endpoint = <&lvds_in_scaler>;
                        };
                };
        };
};
```

```
        vif {
                compatible = "rockchip,rk618-vif";
                clocks = <&clock VIF0_CLK>, <&clock VIF0_PRE_CLK>;
                clock-names = "vif", "vif_pre";
                status = "okay";

                ports {
                        #address-cells = <1>;
                        #size-cells = <0>;

                        port@0 {
                                reg = <0>;

                                vif_in_rgb: endpoint {
                                        remote-endpoint = <&rgb_out_vif>;
                                };
                        };

                        port@1 {
                                reg = <1>;

                                vif_out_hdmi: endpoint {
                                        remote-endpoint = <&hdmi_in_vif>;
                                };
                        };
                };
        };
};
```

```
/ {
        panel {
                compatible = "simple-panel";
                backlight = <&backlight>;
                power-supply = <&vcc3v3_lcd>;
                enable-gpios = <&gpio0 RK_PB5 GPIO_ACTIVE_LOW>;
                prepare-delay-ms = <120>;
                enable-delay-ms = <120>;
                disable-delay-ms = <120>;
                unprepare-delay-ms = <120>;
                bus-format = <MEDIA_BUS_FMT_RGB888_1X7X4_SPWG>;
                width-mm = <231>;
                height-mm = <154>;

                display-timings {
                        native-mode = <&timing1>;

                        timing1: timing1 {
                                clock-frequency = <72000000>;
                                hactive = <1280>;
                                vactive = <800>;
                                hback-porch = <60>;
                                hfront-porch = <60>;
                                vback-porch = <16>;
                                vfront-porch = <16>;
                                hsync-len = <40>;
                                vsync-len = <6>;
                                hsync-active = <0>;
                                vsync-active = <0>;
                                de-active = <0>;
                                pixelclk-active = <0>;
                        };
                };

                port {
                        panel_in_lvds: endpoint {
                                remote-endpoint = <&lvds_out_panel>;
                        };
                };
        };
};
```

8.2.8 调试步骤 Debugging steps

1. 首先根据硬件设计，按 8.2.1 配置 RK618 节点，如果软件驱动没有报错，那么会生成相关调试节点，根据 registers 节点可以通过 I2C 读到 RK618 相关寄存器值。如果读出来的值是 XXXXXXXX，说明设备没有正常工作，需要确认 RK618 供电以及复位 IO 是否配置正确，硬件设计是否有问题。如果供电和复位正常，需要确认 CLKIN 输入时钟是否为 12MHz 连续波形。

Firstly configure RK618 node as described in 8.2.1 according to the hardware design. If software driver doesn't report error, it will generate relative debugging nodes, and then RK618 relative registers values can be read through I2C according to the registers nodes. If the value read out is XXXXXXXX, it means the device doesn't work normally. Need to confirm if the power supply and reset IO of RK618 are configured correctly or not, if the hardware design has problem or not. If the power supply and reset are normal, need to check if the input clock of CLKIN is 12MHz continuous wave.

```
rk3326_evb:/ # cat /d/regmap/0-0050-core/registers
00: 00000000
04: 000100cb
08: 002c0898
0c: 084000c0
10: 00050465
14: 04610029
18: 00000000
1c: 00000000
20: 00000000
24: 00000000
28: 00000000
2c: 00000000
30: 00000014
34: 732d77fd
38: 00090734
3c: 005a050e
40: 04b400b4
44: 000a0276
48: 026c0014
4c: 04b400b4
50: 026c0014
54: 00000011
58: 00000280
5c: 00001d3e
60: 00000000
64: 00002184
68: 00003025
6c: 00000441
70: 00200000
74: 00005028
78: 00000441
7c: 00700000
80: 00000000
84: 00000020
88: 00003c00
8c: 0000ffff
90: 0000000f
94: 0000000f
98: 00000000
9c: 0006e020
```

2. 参考现有配置，并对板级配置进行适配。

Refer to current configuration and adapt with the board level configuration

目前驱动只支持 LCD0 作为输入的单显以及双屏同显应用。

Current driver only supports the application with single display and dual panel same display using LCD0 as input.

如果需求是双屏同显，建议先把 HDMI 和单屏先分别调试完成，再修改配置为双屏同显。

If the requirement is dual panel same display, recommend to debug HDMI and single panel separately first, and then modify the configuration as dual panel same display.

Single-channel LVDS: arch/arm64/boot/dts/rockchip/px30-ad-r35-mb-rk618-lvds.dts

Dual-channel LVDS: arch/arm64/boot/dts/rockchip/px30-ad-r35-mb-rk618-dual-lvds.dts

HDMI: arch/arm64/boot/dts/rockchip/px30-ad-r35-mb-rk618-hdmi.dts

DSI: arch/arm64/boot/dts/rockchip/px30-z7-a0-rk618-dsi.dts

Clone Mode (HDMI and LVDS): arch/arm64/boot/dts/rockchip/px30-ad-r35-mb-rk618-hdmi-lvds.dts

9 MCU/CPU

9.1 Interface

MCU 屏又叫 CPU 屏基于 i80 总线协议，具有 CS、RS、RD、WR 四根控制信号线和 8/16/18/24 数据线，MCU 屏的优点是控制简单，无需同步信号和时钟信号，缺点是屏内部需要集成 GRAM，成本较高，无法做到大分辨率的屏。目前 RK3188、RK3308、RK3326/PX30 支持 MCU 屏。

MCU panel is also called CPU panel which is based on i80 bus protocol, with CS, RS, RD, WR four control signal lines and 8/16/18/24 data lines. MCU panel is easy to control, and doesn’t need sync signal and clock signal. Its disadvantage is the panel inside needs to integrate GRAM, the cost is relatively high, and the panel resolution is not able to be high. Currently RK3188, RK3308, RK3326/PX30 support MCU panel.

控制信号：

The control signal:

CS: 屏的片选信号，低有效，和 VSYNC 复用。

CS: chip selection signal of panel, low effective, and reused with VSYNC.

RS: 数据和命令区分信号，1 表示发送数据，0 表示发送命令，和 DCLK 复用。

RS: distinguish data and command signal, 1 means to output data, 0 means to output command, reused with DCLK.

RD: 1 表示发数据到屏，0 表示从屏读数据(RK 平台不支持)，和 DEN 复用。

RD: 1 means to output data to panel, 0 means to read data from panel (unsupported on RK platforms), reused with DEN.

WR: 写使能信号，上升沿有效，和 HSYNC 复用。

WR: write enable signal, rising edge effective, reused with HSYNC.

9.2 Panel

```
panel: panel {
        compatible = "simple-panel";
        bus-format = <MEDIA_BUS_FMT_RGB565_1X16>;
        backlight = <&backlight>;
        enable-gpios = <&gpio0 RK_PB7 GPIO_ACTIVE_LOW>;
        enable-delay-ms = <20>;
        reset-gpios = <&gpio0 RK_PC4 GPIO_ACTIVE_LOW>;
        reset-value = <0>;
        reset-delay-ms = <10>;
        prepare-delay-ms = <20>;
        unprepare-delay-ms = <20>;
        disable-delay-ms = <20>;
        width-mm = <217>;
        height-mm = <136>;
        rgb-mode = "p565";
        status = "okay";
        rockchip,cmd-type = "mcu";

        // type:0 is cmd, 1 is data
        panel-init-sequence = [
                //type delay num val1 val2 val3
                  00   00  01  e0
        ];

        panel-exit-sequence = [
                00   78  01  10
        ];

        display-timings {
                native-mode = <&kd050fwfba002_timing>;

                kd050fwfba002_timing: timing0 {
                        clock-frequency = <32000000>;
                        hactive = <320>;
```

特殊属性说明 Special property description

Property	Value	Comment
rgb-mode	p888、 p666、 p565、 s888、 s888_dummy	屏的数据接口类型 The type of panel data interface
rockchip,cmd-type	spi or mcu	spi：通过 spi 接口发送初始化命令 spi: output initialization command throught spi interface mcu：通过 mcu 接口发送初始化命令 mcu: output initialization command through mcu interface

9.3 MCU timing

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

        mcu-timing {
                mcu-pix-total = <9>;
                mcu-cs-pst = <1>;
                mcu-cs-pend = <8>;
                mcu-rw-pst = <2>;
                mcu-rw-pend = <5>;

                mcu-hold-mode = <0>;
        };
};
```

1) mcu-pix-total: 发送一次数据/命令需要几个 DCLK 周期；
   mcu-pix-total: each data/command output takes how many DCLK periods
2) mcu-cs-pst/mcu-cs-pend: 片选开始和结束位置；
   mcu-cs-pst/mcu-cs-pend: the start and end location of chip selection.
3) mcu-rw-pst/mcu-rw-pend: 数据发送开始和结束位置；
   mcu-rw-pst/mcu-rw-pend: the start and end location of data output.
   时序图：Timing sequence:

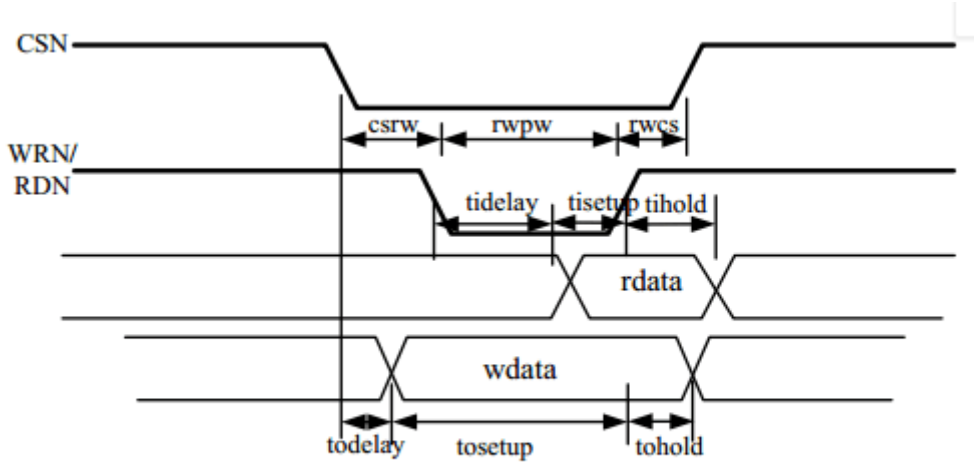


Figure 3-1 i8080 r/w timing

10 Dual-Display

1. 主副屏属性配置（HDMI-A/eDP/DP/LVDS/DPI/DSI）
   Main/Aux panel property configuration (HDMI-A/eDP/DP/LVDS/DPI/DSI)
   device/rockchip/common/ system.prop
   sys.hwc.device.primary=eDP
   sys.hwc.device.extend=HDMI-A
2. 关闭 AFBC 特性
   Disable AFBC feature

VOPL 不支持 AFBC（Arm Frame Buffer Compression），需要关闭 AFBC 特性。

VOPL doesn't support AFBC (Arm Frame Buffer Compression), so need to disable AFBC feature.

hardware/rockchip/libgralloc/Android.mk

将-DUSE_AFBC_LAYER=$(USE_AFBC_LAYER)改为-DUSE_AFBC_LAYER=0.

Modify -DUSE_AFBC_LAYER=$(USE_AFBC_LAYER) to -DUSE_AFBC_LAYER=0.

3. DCLK 父时钟配置

DCLK father clock configuration

如果是 RK3399 平台，并且 uboot 是 rkdevelop 分支，需要对 VOP 的 DCLK 父时钟进行调整，避免时钟切换影响正常显示。

If it is RK3399 platform and uboot is rkdevelop branch, need to adjust DCLK father clock of VOP, to avoid the clock switch affecting the display.

```
--- a/drivers/video/rockchip_vop.c
+++ b/drivers/video/rockchip_vop.c
@@ -221,10 +221,10 @@ static int rockchip_vop_init(struct display_state *state)

 #ifdef CONFIG_RKCHIP_RK3399
        /* Set Dclk pll parent */
-       if (conn_state->type == DRM_MODE_CONNECTOR_HDMIA)
-               rkclk_lcdc_dclk_pll_sel(crtc_state->crtc_id, 0);
-       else
-               rkclk_lcdc_dclk_pll_sel(crtc_state->crtc_id, 1);
+       if (!crtc_state->crtc_id)       /* VOPB */
+               rkclk_lcdc_dclk_pll_sel(crtc_state->crtc_id, 0); /* VPLL */
+       else                            /* VOPL */
+               rkclk_lcdc_dclk_pll_sel(crtc_state->crtc_id, 1); /* CPLL */
 #endif

        /* Set aclk hclk and dclk */
```

```
--- a/arch/arm64/boot/dts/rockchip/rk3399-sapphire-excavator-edp.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3399-sapphire-excavator-edp.dtsi
@@ -329,12 +329,12 @@

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

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

4. 关闭 DDR 变频

Disable DDR frequency scaling

如果是 RK3326/PX30 平台，如果因为带宽不足，导致 VOP 报错，需要将 auto-freq-en 属性设为 0。

If it is RK3326/PX30 platform, and VOP error occurs due to insufficient bandwidth, need to set

auto-freq-en property as 0.

```
&dmc {
        auto-freq-en = <0>;
};
```

11 DEBUG

1. 确认显示驱动已经正常加载。

   Confirm the display driver is already loaded normally.

```
px5:/ # dmesg | grep drm
[    0.000000] Reserved memory: failed to reserve memory for node 'drm-logo@00000000': base 0x0000000000000000, size 0 MiB
[    0.954629] platform vpu_service: allocator is drm
[    0.955398] platform hevc_service: allocator is drm
[    0.964883] [drm:drm_core_init] Initialized drm 1.1.0 20060810
[    0.969040] rockchip-drm display-subsystem: bound ff930000.vop (ops vop_component_ops)
[    0.969108] [drm:rockchip_lvds_bind] *ERROR* failed to find panel and bridge node
[    0.969129] rockchip-drm display-subsystem: failed to bind ff968000.lvds (ops rockchip_lvds_component_ops): -517
[    0.969400] rockchip-drm display-subsystem: master bind failed: -517
[    4.885955] rockchip-drm display-subsystem: bound ff930000.vop (ops vop_component_ops)
[    4.886031] [drm:rockchip_lvds_bind] *ERROR* failed to find panel and bridge node
[    4.886044] rockchip-drm display-subsystem: failed to bind ff968000.lvds (ops rockchip_lvds_component_ops): -517
[    4.886222] rockchip-drm display-subsystem: master bind failed: -517
[    5.401052] rockchip-drm display-subsystem: bound ff930000.vop (ops vop_component_ops)
[    5.401185] rockchip-drm display-subsystem: bound ff968000.lvds (ops rockchip_lvds_component_ops)
[    5.401194] [drm:drm_vblank_init] Supports vblank timestamp caching Rev 2 (21.10.2013).
[    5.401200] [drm:drm_vblank_init] No driver support for vblank timestamp query.
[    5.401305] rockchip-drm display-subsystem: failed to parse loader memory
[    5.404415] rockchip-drm display-subsystem: fb0:  frame buffer device
[    5.453879] [drm:drm_get_platform_dev] Initialized pvr 1.8.4610191 20110701 on minor 1
```

Drm 驱动的加载存在依赖关系，所以可能会多次因为驱动资源暂时获取不到而返回-EPROBE_DEFER（-517），但是只要配置正确，待相关组件驱动能够完整获取到资源后，最终就会 bound 成功。

There is dependency on drm driver loading, so maybe it will return -EPROBE_DEFER（-517）several times as the driver resources cannot be acquired, but as long as the configuration is correct, after relative component drivers acquiring the resources, it will finally be able to bound successfully.

2. 当前显示信息

   Current display information

```
130|rk3399_all:/ # cat /d/dri/0/summary
VOP [ff900000.vop]: ACTIVE
    Connector: eDP
        overlay_mode[0] bus_format[100a] output_mode[f] color_space[0]
    Display mode: 1536x2048p60
        clk[200000] real_clk[200000] type[8] flag[a]
        H: 1536 1548 1564 1612
        V: 2048 2056 2060 2068
    win0-0: DISABLED
    win1-0: DISABLED
    win2-0: ACTIVE
        format: XB24 little-endian (0x34324258) SDR[0] color_space[0]
        csc: y2r[0] r2r[0] r2y[0] csc mode[0]
        zpos: 0
        src: pos[512x0] rect[1536x2048]
        dst: pos[0x0] rect[1536x2048]
        buf[0]: addr: 0x00000000086e6000 pitch: 8192 offset: 0
    win2-1: DISABLED
    win2-2: DISABLED
    win2-3: DISABLED
    win3-0: ACTIVE
        format: AB24 little-endian (0x34324241) SDR[0] color_space[0]
        csc: y2r[0] r2r[0] r2y[0] csc mode[0]
        zpos: 1
        src: pos[0x0] rect[1536x2048]
        dst: pos[0x0] rect[1536x2048]
        buf[0]: addr: 0x000000000c8de000 pitch: 6144 offset: 0
    win3-1: DISABLED
    win3-2: DISABLED
    win3-3: DISABLED
    post: sdr2hdr[0] hdr2sdr[0]
    pre : sdr2hdr[0]
    post CSC: r2y[0] y2r[0] CSC mode[1]
VOP [ff8f0000.vop]: DISABLED
rk3399_all:/ # cat /d/dri/0/summary
VOP [ff900000.vop]: DISABLED
VOP [ff8f0000.vop]: DISABLED
```

3. connector 当前连接状态

   Current connection status of connector

```
rk3399_all:/ # cat /sys/class/drm/card0-eDP-1/status
connected
rk3399_all:/ # cat /sys/class/drm/card0-HDMI-A-1/status
disconnected
```

4. connector 当前使能状态

   Current enable status of connector

```
rk3399_all:/ # cat /sys/class/drm/card0-eDP-1/enabled
enabled
rk3399_all:/ # cat /sys/class/drm/card0-HDMI-A-1/enabled
disabled
```

5. connector 支持的显示模式

   The display mode supported by connector

```
rk3399_all:/ # cat /sys/class/drm/card0-eDP-1/modes
1536x2048p60
```

6. connector 当前的显示模式

   Current display mode of connector

```
rk3399_all:/ # cat /sys/class/drm/card0-eDP-1/mode
1536x2048p60
```

7. 手动灭屏/亮屏

   Manually turn off/on the screen

echo off > /sys/class/drm/card0-eDP-1/status

echo on > /sys/class/drm/card0-eDP-1/status