密级状态：绝密(　　)　　秘密(　　)　　内部(　　)　　公开(√)

Rockchip DRM Panel Porting Guide

（第二系统产品部）

文件状态： [√] 正在修改 [　] 正式发布	当前版本：	V1.0
	作　　者：	闭伟勇
	完成日期：	2017-4-15
	审　　核：	
	完成日期：	

福州瑞芯微电子股份有限公司

Fuzhou　Rockchips　Semiconductor　Co . , Ltd

版 本 历 史

版本号	作者	修改日期	修改说明	备注
V1.0	闭伟勇	2017-4-15	初始版本	
V1.1	黄家钗	2017-4-17	加入 LVDS 屏配置说明	

目 录

1 基本结构

本文档基于以下平台进行说明，不同平台相关驱动以及 DTS 配置方式可能存在细微差异。

Platform：RK3368H

Board：rk3368-sheep

1.1 文件清单

1.1.1 Kernel

驱动路径：

drivers/gpu/drm/rockchip/

drivers/gpu/drm/bridge/

drivers/gpu/drm/panel/

drivers/phy/

文档路径：

Documentation/devicetree/bindings/display/rockchip/

Documentation/devicetree/bindings/display/bridge/

Documentation/devicetree/bindings/display/panel/

Documentation/devicetree/bindings/phy/

下表只列出具体文件：

Driver	File	Doc
Core	rockchip_drm_drv.c	rockchip-drm.txt
Framebuffer	rockchip_drm_fb.c	
GEM	rockchip_drm_gem.c	
VOP	rockchip_drm_vop.c rockchip_vop_reg.c	rockchip-vop.txt
LVDS	rockchip_lvds.c	rockchip-lvds.txt

RGA	rockchip_drm_rga.c	rockchip-rga.txt
MIPI	dw-mipi-dsi.c phy-rockchip-inno-mipi-dphy.c	dw_mipi_dsi_rockchip.txt phy-rockchip-inno-mipi-dphy.txt
HDMI	dw_hdmi-rockchip.c dw-hdmi.c	dw_hdmi-rockchip.txt dw_hdmi.txt
INNO HDMI	inno_hdmi.c	inno_hdmi-rockchip.txt
eDP	analogix_dp-rockchip.c analogix_dp_core.c analogix_dp_reg.c phy-rockchip-dp.c	analogix_dp-rockchip.txt analogix_dp.txt rockchip-dp-phy.txt
DP	cdn-dp-core.c cdn-dp-reg.c	cdn-dp-rockchip.txt
Panel	panel-simple.c	simple-panel.txt

1.1.2 U-boot

驱动路径：

drivers/video/

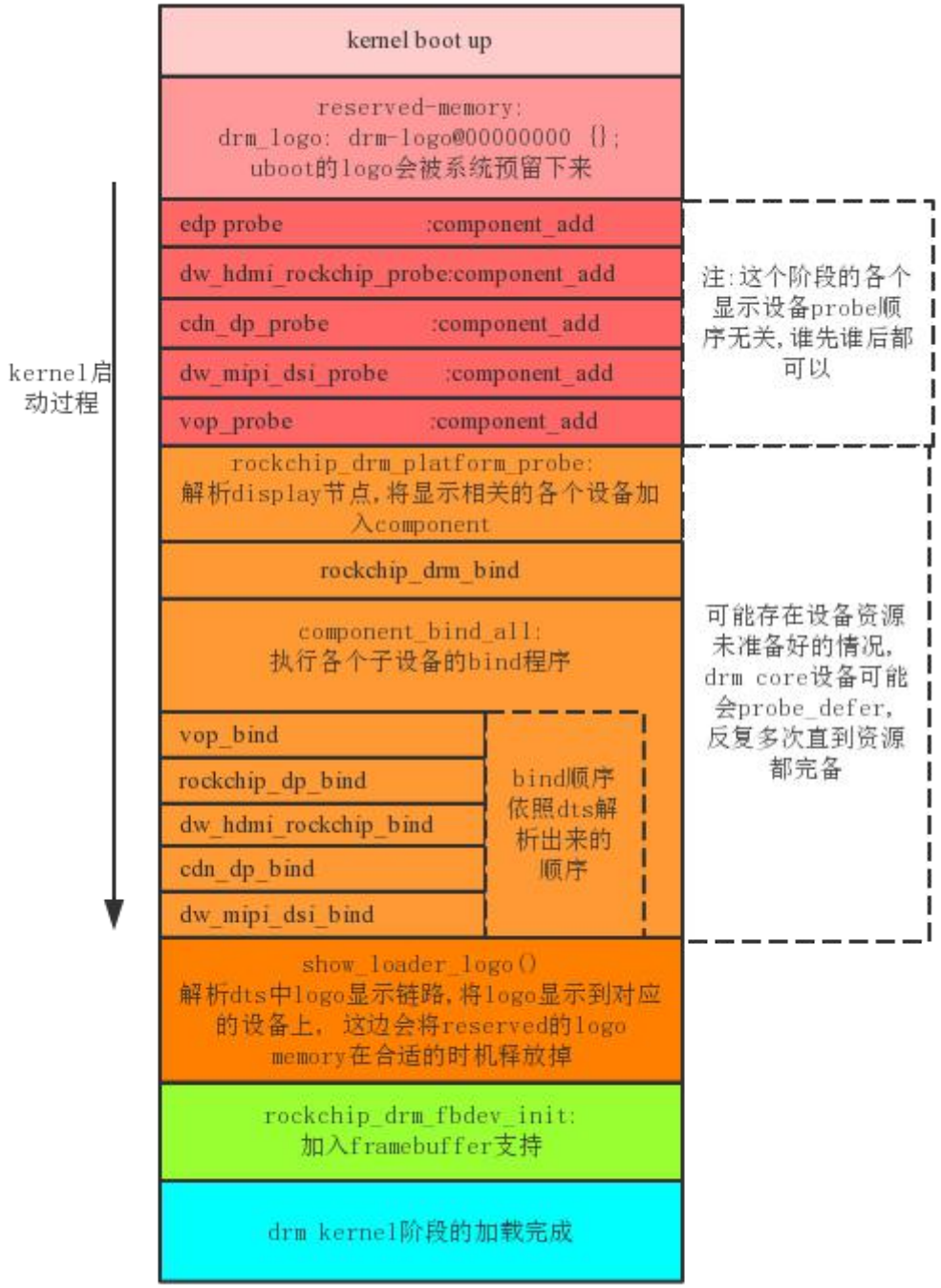
下表只列出具体文件：

Driver	**File**
Core	rockchip_display.c rockchip_crtc.c rockchip_connector.c rockchip_phy.c rockchip_panel.c
VOP	rockchip_vop.c rockchip_vop_reg.c
eDP	rockchip_analogix_dp.c rockchip_analogix_dp_reg.c

MIPI	rockchip_mipi_dsi.c rockchip-dw-mipi-dsi.c rockchip-inno-mipi-dphy.c
Panel	panel_simple.c rockchip_dsi_panel.c
LVDS	rockchip_lvds.c

1.2 整体流程

1.2.1 Kernel



2 MIPI

Platform：RK33668H

Board：Sheep

Panel：KingDisplay KD080D24-40NI-B6

2.1 Kernel

2.1.1 MIPI Host

$ vim arch/arm64/boot/dts/rockchip/rk3368-sheep.dts

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

RK3399对应&mipi_dsi

2.1.2 MIPI PHY

RK3399没有PHY这一块的配置

$ vim arch/arm64/boot/dts/rockchip/rk3368-sheep.dts

```
&mipi_dphy {
        status = "okay";
        rockchip,dsi-panel = <&dsi_panel>;
};
```

① 属性说明

Property	Value	Comment
rockchip,dsi-panel	&dsi_panel	PHY 驱动通过该 phandle，可以获取屏相关参数，根据屏的 timing 计算出所需 bit-rate，同时对 PHY 进行正确配置，并将该参数传递给 HOST，使 HOST 能正确初始化。

2.1.3 LOGO

$ vim arch/arm64/boot/dts/rockchip/rk3368-sheep.dts

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

Note：如果没有开启 logo，只有 Android 启动才会显示 Android Logo。

2.1.4 Panel

```
&mipi_dsi_host {
        dsi_panel: panel@0 {
                compatible = "simple-panel-dsi";
                reg = <0>;

                backlight = <&backlight>;
                reset-gpios = <&gpio0 21 GPIO_ACTIVE_LOW>;
                enable-gpios = <&gpio0 22 GPIO_ACTIVE_HIGH>;

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

                delay,prepare = <20>;
                reset-delay-ms = <20>;
```

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

                panel-exit-sequence = [
                        05 00 01 28
                        05 78 01 10
                ];
```

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

        timing0: timing0 {
                clock-frequency = <66000000>;
                hactive = <800>;
                hfront-porch = <18>;
                hsync-len = <18>;
                hback-porch = <18>;
                vactive = <1280>;
                vfront-porch = <6>;
                vsync-len = <4>;
                vback-porch = <6>;
                hsync-active = <0>;
                vsync-active = <0>;
                de-active = <0>;
                pixelclk-active = <0>;
        };
    };
};
```

① 属性说明

Property	**Value**	**Comment**
compatible	simple-panel-dsi	与 panel 驱动进行匹配，MIPI 屏统一使用该字符串进行匹配。
reg	0	DSI virtual channel
backlight	&backlight	引用 backlight 节点，panel 驱动会对背光进行控制。
reset-gpios	&gpio0 21 GPIO_ACTIVE_LOW	屏的 Reset 脚 GPIO 配置，参考原理图。
enable-gpios	&gpio0 22 GPIO_ACTIVE_HIGH	屏的 Enable 脚 GPIO 配置，参考原理图。
dsi,flags	(MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST)	Video Burst Mode，大部分 MIPI 屏都是使用该模式，参考屏规格书和MIPI 协议。
dsi,format	MIPI_DSI_FMT_RGB888	Pixel Format
dsi,lanes	4	Lane Number
delay,prepare	20	Enable 信号有效之后，延时 20ms，参考屏规格书和屏驱动。
reset-delay-ms	20	Reset 信号有效脉冲宽度，参考屏规格书

		和屏驱动。
panel-init-sequence	…	屏的上电初始化序列，一般由屏规格书说明或者由屏厂 FAE 提供，具体参数配置方式参考下文说明。
panel-exit-sequence	…	屏的下电初始化序列，一般由屏规格书说明或者由屏厂 FAE 提供，具体参数配置方式参考下文说明。
display-timings	…	屏的 timing，参考屏规格书。

Note：除了以上部分所列 Property，还有其他可选的 Property 可以配置，具体参考内核相关文档或者屏驱动的代码。

2.1.5 Command

2.1.5.1 命令格式

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

panel-exit-sequence = [
        05 00 01 28
        05 78 01 10
];
```

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

从第四个字节开始的数据代表长度为 Length 的实际有效 Payload。

以 panel-init-sequence 为例：

第一条命令[39 00 04 b9 ff 83 94]的解析如下：

Data Type： 0x39 (DCS Long Write)

Delay：0x00 (ms)

Payload Length：0x04 (Bytes)

Payload：0xb9 0xff 0x83 0x94

最后一条命令[05 00 01 29]的解析如下：

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

Delay：0x00 (ms)

Payload Length：0x01 (Bytes)

Payload：0x29

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
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	

说明：

以常用的 Write Packet 为例。

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。

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。

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。

Note：DCS 命令的含义由 MIPI Alliance Specification for Display Command Set.pdf 标准文档指定。厂商自定义的命令按 Generic Write 方式进行发送。有的屏不会区分 DCS Packet 和 Generic Packet，都会按 DCS 的方式发 Packet，这需要参考屏规格书或咨询屏厂 FAE。

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

Gerneic Packet 包括 n 个字节的 parameters。

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。

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。

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。

3）Payload Length

表示 Packet 的有效负载长度。

4）Payload

表示 Packet 的有效负载，长度为 Payload Length。

Note：对于不同的 SDK，Payload 之前的头部可能会有细微差异，因为 Rockchip 可能会根据需求，对 DTS 中 Command 的前三个字节格式进行相应扩展，具体参考 simple-panel.txt 文档，它会与发布的代码同步更新。

2.1.5.2 Example

Note：为了方便，以下例子以 Japan Display 的 H070ME05000 进行说明，该屏的规格书有详细的初始化序列说明。

(B) On sequence

sequence	DataType (hex)	index (hex)	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 b6 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 #	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
];
```

2.1.6 常见问题

（1）驱动 bind 不成功。

解决方法：

① 检查相关驱动是否有编进内核。

② 检查 dts 中相关驱动的 node 是否配置正确。

③ 检查 dts 中 panel 节点下的资源是否与其他设备冲突，从而导致屏驱动因获取不到资源而加载失败，例如 enable-gpios 与 backlight 的 enable-gpios 冲突。

（2）开启 logo 的情况下，uboot 阶段显示正常，从 kernel logo 开始图像有偏移，休眠唤醒后才正

常。

解决方法：先尝试将 panel 的 clock-frequency（像素时钟）往下调，直到显示正常。之后再调整 iming 将帧率调到所需帧率。

3 eDP

Platform： RK33668H

Board： Sheep

Panel： LG LP079QX1-SP0V

3.1 配置方式 1

3.1.1 Kernel

3.1.1.1 Panel

$ vim arch/arm64/boot/dts/rockchip/rk3368-sheep.dts

```
/ {
        edp_panel: edp-panel {
                compatible = "simple-panel";
                backlight = <&backlight>;
                enable-gpios = <&gpio0 22 GPIO_ACTIVE_HIGH>;

                delay,prepare = <120>;
```

```
                disp_timings: 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>;
                        };
                };
        };
};
```

Note： 具体 Property 含义以及其他可选 Property 配置，参考内核相关文档和驱动。

3.1.1.2 eDP Host

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

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

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

Note： 具体 Property 含义以及其他可选 Property 配置，参考内核相关文档和驱动。

3.1.1.3 eDP PHY

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

3.1.1.4 LOGO

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

3.2 配置方式 2

把 Timing 写在 panel-simple.c 中，直接以短字符串匹配，该方式为 upstream 推荐的使用方式。

3.2.1 Kernel

3.2.1.1 Panel

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

```
static const struct drm_display_mode lg_lp079qx1_sp0v_mode = {
        .clock = 200000,
        .hdisplay = 1536,
        .hsync_start = 1536 + 12,
        .hsync_end = 1536 + 12 + 16,
        .htotal = 1536 + 12 + 16 + 48,
        .vdisplay = 2048,
        .vsync_start = 2048 + 8,
        .vsync_end = 2048 + 8 + 4,
        .vtotal = 2048 + 8 + 4 + 8,
        .vrefresh = 60,
        .flags = DRM_MODE_FLAG_NVSYNC | DRM_MODE_FLAG_NHSYNC,
};

static const struct panel_desc lg_lp079qx1_sp0v = {
        .modes = &lg_lp079qx1_sp0v_mode,
        .num_modes = 1,
        .size = {
                .width = 129,
                .height = 171,
        },
        .bus_format = MEDIA_BUS_FMT_RGB666_1X18,
};
```

```
static const struct of_device_id platform_of_match[] = {
        {
                .compatible = "simple-panel",
                .data = NULL,
        }, {
                .compatible = "lg,lp079qx1-sp0v",
                .data = &lg_lp079qx1_sp0v,
        }, {
                /* sentinel */
        }
};
MODULE_DEVICE_TABLE(of, platform_of_match);
```

$ vim arch/arm64/boot/dts/rockchip/rk3368-sheep.dts

```
/ {
        edp_panel: edp-panel {
                compatible = "lg,lp079qx1-sp0v";
                backlight = <&backlight>;
                enable-gpios = <&gpio0 22 GPIO_ACTIVE_HIGH>;

                delay,prepare = <120>;

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

Note： 具体 Property 含义以及其他可选 Property 配置，参考内核相关文档和驱动。

3.2.1.2 eDP Host

$ vim arch/arm64/boot/dts/rockchip/rk3368-sheep.dts

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

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

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

Note： 具体 Property 含义以及其他可选 Property 配置，参考内核相关文档和驱动。

3.2.1.3 eDP PHY

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

3.2.1.4 LOGO

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

3.2.2 U-boot

$ vim drivers/video/rockchip_panel.c

```
static const struct drm_display_mode lg_lp079qx1_sp0v_mode = {
        .clock = 200000,
        .hdisplay = 1536,
        .hsync_start = 1536 + 12,
        .hsync_end = 1536 + 12 + 16,
        .htotal = 1536 + 12 + 16 + 48,
        .vdisplay = 2048,
        .vsync_start = 2048 + 8,
        .vsync_end = 2048 + 8 + 4,
        .vtotal = 2048 + 8 + 4 + 8,
        .vrefresh = 60,
        .flags = DRM_MODE_FLAG_NVSYNC | DRM_MODE_FLAG_NHSYNC,
};
```

```
static const struct rockchip_panel g_panel[] = {
        {
                .compatible = "simple-panel",
                .funcs = &panel_simple_funcs,
        }, {
                .compatible = "simple-panel-dsi",
                .funcs = &rockchip_dsi_panel_funcs,
        }, {
                .compatible = "lg,lp079qx1-sp0v",
                .funcs = &panel_simple_funcs,
                .data = &lg_lp079qx1_sp0v_mode,
        }, {
                .compatible = "auo,b125han03",
                .funcs = &panel_simple_funcs,
                .data = &auo_b125han03_mode,
        },
};
```

3.3 配置方式 3

不填写任何 Timing，直接使用 EDID 来获取 Timing。

3.3.1 Kernel

3.3.1.1 Panel

$ vim arch/arm64/boot/dts/rockchip/rk3368-sheep.dts

```
/ {
        edp_panel: edp-panel {
                compatible = "simple-panel";
                backlight = <&backlight>;
                enable-gpios = <&gpio0 22 GPIO_ACTIVE_HIGH>;

                delay,prepare = <120>;

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

3.3.1.2 eDP Host

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

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

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

3.3.1.3 eDP PHY

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

3.3.1.4 LOGO

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

```
delay prepare[120] unprepare[0] enable[0] disable[0]
read logo on state from dts [1]
EDID data does not include any extensions.
Using display timing from edid
EDID version: 1.4
Product ID code: c118
Manufacturer: APP
Serial number: 00000000
Manufactured in week: 26 year: 2012
Video input definition: digital signal, voltage level 1, composite sync, serration v
Monitor is RGB
Maximum visible display size: 12 cm x 16 cm
Power management features: no active off, no suspend, no standby
Estabilished timings:
Standard timings:
        1536x2048       59 Hz (detailed)
Monitor ID: 079L1JY41
Monitor name: Color LCD
Detailed mode clock 200000 kHz, flags[a]
    H: 1536 1548 1564 1612
    V: 2048 2056 2060 2068
bus_format: 100e
Link Training Clock Recovery success
Link Training success!
```

4 LVDS

Platform：RK33668H

Board：Sheep

Panel：SAMSUNG LSL070NL01

4.1 LVDS 节点配置

$ vim arch/arm64/boot/dts/rockchip/rk3368-sheep.dts

1）打开 uboot 显示，这边如果没有设置为 okay，到 android 起来后才可以看到显示。

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

2） LVDS 输出模式配置

```
&lvds {
        status = "okay";
        rockchip,data-mapping = "vesa";
        rockchip,data-width = <24>;
        rockchip,output = "lvds";
        rockchip,panel = <&lvds_panel>;
};
```

3）时序、电源等相关配置

```
        lvds_panel: lvds-panel {
                status = "okay";
                compatible = "simple-panel";
                backlight = <&backlight>;
                enable-gpios = <&gpio0 22 GPIO_ACTIVE_HIGH>;
                display-timings {
                        native-mode = <&timing0>;
                        timing0: timing0 {
                                clock-frequency = <54000000>;
                                hactive = <1024>;
                                vactive = <600>;
                                hback-porch = <134>;
                                hfront-porch = <134>;
                                vback-porch = <10>;
                                vfront-porch = <10>;
                                hsync-len = <134>;
                                vsync-len = <10>;
                                hsync-active = <0>;
                                vsync-active = <0>;
                                de-active = <0>;
                                pixelclk-active = <0>;
                        };
                };
        };
```

4.2 属性说明

Property	Value	Comment
rockchip,data-mapping	vesa or jeida	LVDS 信号的两种编码方式，具体对应关系参考 4.3 的 data mapping 说明。
rockchip,data-width	18 or 24 or 30	LVDS 的数据位，RGB 三个分量都是 6 的填 18，RGB 三个分量都是 8 的填 24，RGB 三个分量都是 10 的填 30，
rockchip,output	lvds or rgb	LVDS 输出的两种模式： lvds：LVDS 屏的配置； rgb：RGB 屏的配置。

compatible	simple-panel	与 panel 驱动进行匹配。
backlight	&backlight	引用 backlight 节点，panel 驱动会对背光进行控制。
enable-gpios	&gpio0 22 GPIO_ACTIVE_HIGH	屏的 Enable 脚 GPIO 配置，参考原理图。
delay,prepare	20	Enable 信号有效之后，延时 20ms。

4.3 Data mapping

1）6 bit output mode

采用 4+1 的传输模式，既 4 组数据信号加一组时钟信号，最后一组数据信号传输无效数据。

		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

2）8 bit output mode

采用 4+1 的传输模式，既 4 组数据信号加一组时钟信号。

		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

3）10 bit output mode

采用 5+1 的传输模式，既 5 组数据信号加一组时钟信号

		VESA_10BIT	JEIDA_10BIT
	TX0	R0	R4
	TX1	R1	R5
	TX2	R2	R6
Y	TX3	R3	R7
0	TX4	R4	R8
	TX6	R5	R9
	TX7	G0	G4
	TX8	G1	G5
	TX9	G2	G6
Y	TX12	G3	G7
1	TX13	G4	G8
	TX14	G5	G9
	TX15	B0	B4
	TX18	B1	B5
	TX19	B2	B6
	TX20	B3	B7
	TX21	B4	B8
Y	TX22	B5	B9
2	TX24	HSYNC	HSYNC
	TX25	VSYNC	VSYNC
	TX26	ENABLE	ENABLE
	TX27	R6	R2
	TX5	R7	R3
	TX10	G6	G2
Y	TX11	G7	G3
3	TX16	B6	B2
	TX17	B7	B3
	TX23	GND	GND
	TX27	R8	R0
	TX5	R9	R1
	TX10	G8	G0
Y	TX11	G9	G1
4	TX16	B8	B0
	TX17	B9	B1
	TX23	GND	GND