



AW DRM 屏幕适配 调试指南

版本号: 1.3

发布日期: 2025.07.11

版本历史

版本号	日期	制/修订人	内容描述
1.0	2023.03.08	AWA1442	初始版本。
1.1	2024.12.08	AWA2081	增加多屏兼容、dsc 配置。
1.2	2025.03.11	AWA2271	增加 A537_A333 的描述。
1.3	2025.07.11	AWA2447	增加 T153 的描述。

目 录

1 前言	1
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
2 显示连接关系	2
2.1 硬件连接关系	2
2.1.1 A523_A527_T527 显示硬件连接关系	3
2.1.2 A733 显示硬件连接关系	4
2.1.3 A537_A333 显示硬件连接关系	5
2.1.4 T153 显示硬件连接关系	5
2.2 软件 dts 连接关系	6
3 背光	9
3.1 menuconfig 配置	9
3.2 defconfig 设置	9
3.3 board.dts 配置	10
3.4 uboot-board.dts 配置	13
4 MIPI DSI	14
4.1 menuconfig 配置	14
4.1.1 dsi 控制器配置	14
4.1.2 dsi 屏配置	14
4.1.3 defconfig 设置	15
4.2 board.dts 配置	15
4.2.1 panel	15
4.2.1.1 dsi,flags	18
4.2.1.2 Command	19
4.2.1.3 dsc	21
4.2.2 dsi	22
4.2.2.1 dsi0 接口	22
4.2.2.2 dsi1 接口	23
4.2.2.3 dual-dsi 接口	23
4.2.2.4 多屏兼容	25
4.2.3 tcon	26
4.2.4 combophy	27
4.2.5 video_out	27
4.2.6 pinctl	27
4.3 常见问题	28

5	LVDS	30
5.1	menuconfig 配置	30
5.1.1	lvds 控制器配置	30
5.1.2	lvds 屏配置	30
5.1.3	defconfig 设置	30
5.2	board.dts 配置	30
5.2.1	panel	30
5.2.2	lvds	31
5.2.3	tcon	32
5.2.4	combophy	33
5.2.5	video_out	33
5.2.6	pinctl	34
5.3	常见问题	34
6	RGB	35
6.1	menuconfig 配置	35
6.1.1	rgb 控制器配置	35
6.1.2	rgb 屏配置	35
6.1.3	defconfig 设置	35
6.2	board.dts 配置	35
6.2.1	panel	35
6.2.2	rgb	36
6.2.3	tcon	37
6.2.4	video_out	37
6.2.5	pinctl	38
7	eDP	39
7.1	menuconfig 配置	39
7.1.1	edp 控制器配置	39
7.1.2	edp 屏配置	40
7.1.3	defconfig 设置	40
7.2	board.dts 配置	40
7.2.1	panel	40
7.2.2	drm_edp	43
7.2.3	tcon	44
7.2.4	video_out	45
7.2.5	combophy	46
8	各屏 uboot-board.dts 配置	47
9	开机 logo	48
9.1	route	48
9.1.1	uboot-board.dts 配置	48
9.1.1.1	单显	48

9.1.1.2 双显	48
10 单双显	50
10.1 单双显图层配置	50
11 调试方法	51
11.1 dsi	51
11.2 edp	51
11.2.1 AUX 通信失败	51
11.3 uboot 通用调试方法	51
11.3.1 确认驱动加载情况	51
11.4 kernel 通用调试方法	52
11.4.1 确认驱动加载情况	52
12 常见问题	53
12.1 disp 与 drm 屏参转化	53
13 附录	54
13.1 屏幕问题硬件 checklist	54



插 图

图 2-1	硬件连接关系简图	2
图 2-2	LVDS 硬件框图	3
图 2-3	软件 dts 连接关系简图	6
图 3-1	pwm 背光 menuconfig	9
图 3-2	pwm 背光 dts	12
图 4-1	dsi menuconfig	14
图 4-2	data-type	20
图 7-1	edp menuconfig	39
图 7-2	edp panel_menuconfig	40
图 7-3	edp panel_dts	42
图 7-4	edp node_dts	44
图 7-5	edp_tcon_dts	45
图 7-6	edp_vo_dts	45
图 7-7	edp_phy_dts	46
图 13-1	硬件排查 checklist	54
图 13-2	硬件排查 checklist	55

表 格

表 1-1	适用产品列表	1
表 2-1	显示通路硬件模块简介	2
表 2-2	A523_A527_T527 显示硬件连接关系	3
表 2-3	A733 显示硬件连接关系	4
表 2-4	A537_A333 显示硬件连接关系	5
表 2-5	T153 显示硬件连接关系	5
表 3-1	背光属性介绍	13
表 4-1	dsi 屏驱动属性介绍	17
表 4-2	dsc 模块属性介绍	21
表 4-3	dsi 模块属性介绍	22
表 5-1	lvds 屏驱动属性介绍	31
表 5-2	lvds 模块属性介绍	32
表 6-1	rgb 屏驱动属性介绍	36
表 6-2	rgb 模块属性介绍	37
表 7-1	edp 屏驱动属性介绍	43
表 7-2	edp 模块属性介绍	44
表 7-3	edp tcon 模块属性介绍	45
表 7-4	edp vo 模块属性介绍	45
表 7-5	edp phy 模块属性介绍	46
表 9-1	bootlogo route 属性介绍	48
表 10-1	chn_cfg_mode 对应显示场景	50

1 前言

1.1 文档简介

介绍全志平台基于 DRM 显示框架的各类显示屏的适配方法及调试方法，如：LVDS、MIPI DSI、RGB、eDP 等。

1.2 目标读者

需要完成新屏幕适配的显示开发工程师。

1.3 适用范围

表 1-1: 适用产品列表

产品名称	内核版本
T527-AIOT	Linux-5.15
T527-AIOT	Linux-5.10
A733	Linux-6.6
A537	Linux-6.6
A333	Linux-6.6
T153	Linux-5.10-rt

2 显示连接关系

2.1 硬件连接关系

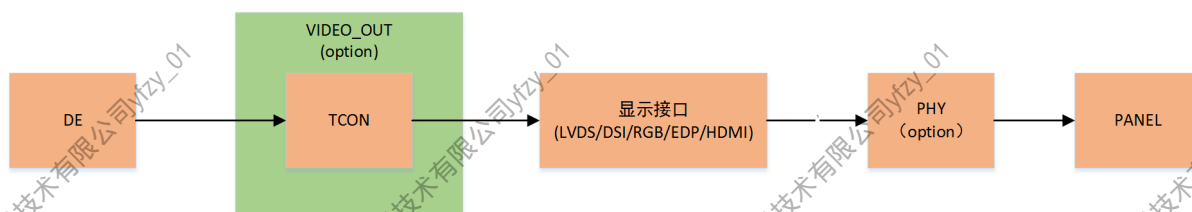


图 2-1: 硬件连接关系简图

表 2-1: 显示通路硬件模块简介

模块	介绍
DE	完成图层合成和渲染等工作的显示引擎，显示系统的核心
VIDEO_OUT	显示子系统域，并非所有平台都有这个模块，用于显示系统顶层功耗管理和通路管理
TCON	时序控制器，将 DE 合成的数据转换成对应的时序信息
显示接口	通常指显示接口的控制器，常见的显示接口有 MIPI DSI、LVDS、eDP、HDMI 等
PHY	物理层，根据时序信息生成对应的模拟信号，并非每个显示通路都有，部分显示接口集成在内部
PANEL	通常指的就是我们常规意义的物理屏幕，如 DSI 屏、LVDS 屏、eDP 屏等

• LVDS 接口硬件介绍

显示接口的硬件中，LVDS 接口的设计与实际较为特殊，在本文档中单独拎出来介绍。

在 AW 平台中，LVDS 通常分为两组（group），每组对应一个硬件模块，每个 group 中又可以支持两个 LVDS 接口，其中 group0 支持 LVDS0 和 LVDS1，group1 支持 LVDS2 和 LVDS3。当需要开启 dual link LVDS 显示时，会将同个 group 中的两个 LVDS 接口都使能，使其同时承载数据传输。

下文及日常方案开发时提到的 LVDSx 常指代 group 中具体的某个 LVDS 接口（如 LVDS1、LVDS3）。但是由于实际硬件是按 group 划分，因此驱动及 dts 配置中的 lvdsx 通常指代某个 lvds_group，这点需要额外注意，特别是在进行 LVDS 适配时需要根据下文的连接关系，打开对

应 LVDSx 对应的 group 的节点，以保证驱动正常加载。

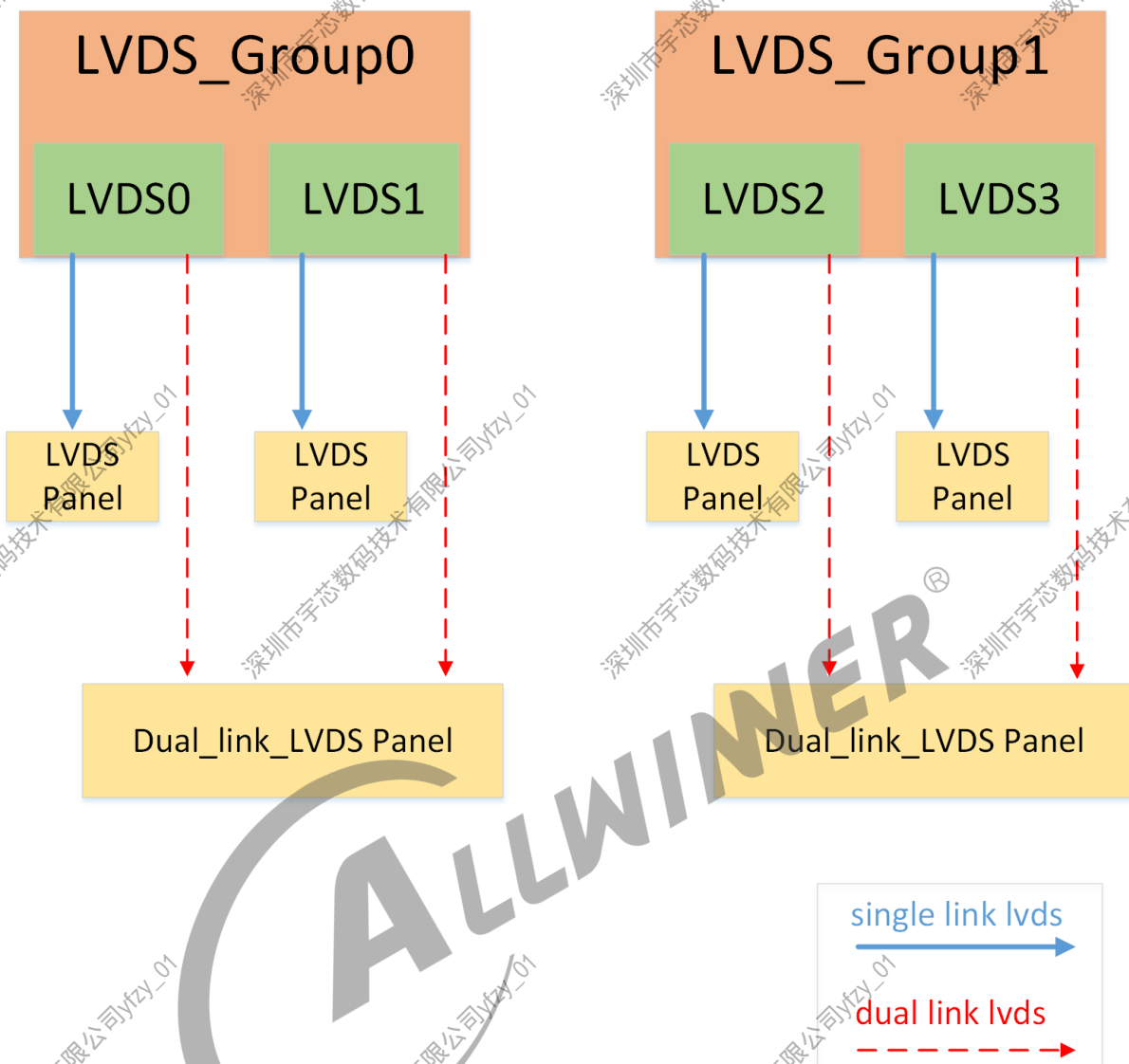


图 2-2: LVDS 硬件框图

2.1.1 A523_A527_T527 显示硬件连接关系

表 2-2: A523_A527_T527 显示硬件连接关系

显示接口	所属 Video_Out	所属 Group	Tcon 连接	外部 PHY 连接
DE	vo0	N/A	N/A	N/A
DSI0	vo0	N/A	dlcd0	dsi0combophy
DSI1	vo0	N/A	dlcd1	dsi1combophy

显示接口	所属 Video_Out	所属 Group	Tcon 连接	外部 PHY 连接
双 DSI	vo0	N/A	dlcd0 dlcd1	dsi0combophy dsi1combophy
LVDS0	vo0	lvds0	dlcd0	dsi0combophy
LVDS1	vo0	lvds0	dlcd0	dsi1combophy
Dual-LVDS0+1	vo0	lvds0	dlcd0	dsi0combophy dsi1combophy
LVDS2	vo1	lvds1	dlcd2	N/A
LVDS3	vo1	lvds1	dlcd2	N/A
Dual-LVDS2+3	vo1	lvds1	dlcd2	N/A
RGB0	vo0	N/A	dlcd0	N/A
RGB1	vo1	N/A	dlcd2	N/A
EDP	vo0	N/A	tv1	N/A
HDMI	vo0	N/A	tv0	N/A

2.1.2 A733 显示硬件连接关系

表 2-3: A733 显示硬件连接关系

显示接口	所属 Video_Out	所属 Group	Tcon 连接	外部 PHY 连接
DE	vo0	N/A	N/A	N/A
DSI0	vo0	N/A	dlcd0	dsi0combophy
DSI1	vo0	N/A	dlcd1	dsi1combophy
双 DSI	vo0	N/A	dlcd0 dlcd1	dsi0combophy dsi1combophy
LVDS0	vo0	lvds0	dlcd0	dsi0combophy
LVDS1	vo0	lvds0	dlcd0	dsi1combophy
Dual-LVDS0+1	vo0	lvds0	dlcd0	dsi0combophy dsi1combophy
LVDS2	vo0	lvds1	dlcd2	N/A
LVDS3	vo0	lvds1	dlcd2	N/A
Dual-LVDS2+3	vo0	lvds1	dlcd2	N/A

显示接口	所属 Video_Out	所属 Group	Tcon 连接	外部 PHY 连接
RGB0	vo0	N/A	dlcd0	N/A
RGB1	vo0	N/A	dlcd2	N/A
EDP	vo1	N/A	tv1	serdes
HDMI	vo1	N/A	tv0	N/A

2.1.3 A537_A333 显示硬件连接关系

表 2-4: A537_A333 显示硬件连接关系

显示接口	所属 Video_Out	所属 Group	Tcon 连接	外部 PHY 连接
DE	vo0	N/A	N/A	N/A
DSI0	vo0	N/A	dlcd0	dsi0combophy
LVDS0	vo0	lvds0	dlcd0	dsi0combophy
LVDS1	vo0	lvds0	dlcd0	dsi0combophy
Dual-LVDS0+1	vo0	lvds0	dlcd0	dsi0combophy
RGB0	vo0	N/A	dlcd0	N/A
EDP	vo1	N/A	tv1	serdes
HDMI	vo1	N/A	tv0	N/A

2.1.4 T153 显示硬件连接关系

表 2-5: T153 显示硬件连接关系

显示接口	所属 Video_Out	所属 Group	Tcon 连接	外部 PHY 连接
DE	vo0	N/A	N/A	N/A
DSI0	vo0	N/A	dlcd0	dsi0combophy
LVDS0	vo0	lvds0	dlcd0	dsi0combophy
LVDS1	vo0	lvds0	dlcd0	N/A

显示接口	所属 Video_Out	所属 Group	Tcon 连接	外部 PHY 连接
Dual-LVDS0+1	vo0	lvds0	dlcd0	dsi0combophy
RGB0	vo0	N/A	dlcd0	N/A

2.2 软件 dts 连接关系

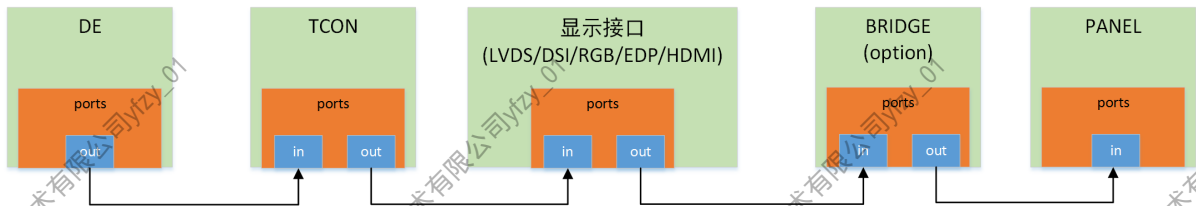


图 2-3: 软件 dts 连接关系简图

```

de: @5000000 {
    ...
    ports { /* DE端口列表 */
        #address-cells = <1>;
        #size-cells = <0>;
        disp0: port@0 { /* 第一个DE输出端口列表 */
            reg = <0>;
            #address-cells = <1>;
            #size-cells = <0>;
            disp0_out_tcon0: endpoint@0 { /* DE的第一个out端口 */
                reg = <0>;
                remote-endpoint = <&tcon0_in_disp0>; /* 设置下一级的in节点 */
            };
            disp0_out_tcon1: endpoint@1 {
                reg = <1>;
                remote-endpoint = <&tcon1_in_disp0>;
            };
            disp0_out_tcon2: endpoint@2 {
                reg = <2>;
                remote-endpoint = <&tcon2_in_disp0>;
            };
            disp0_out_tcon3: endpoint@3 {
                reg = <3>;
                remote-endpoint = <&tcon3_in_disp0>;
            };
            disp0_out_tcon4: endpoint@4 {
                reg = <4>;
                remote-endpoint = <&tcon4_in_disp0>;
            };
        };
        disp1: port@1 { /* 第二个DE输出端口 */
            reg = <1>;
            #address-cells = <1>;
            #size-cells = <0>;
            disp1_out_tcon0: endpoint@0 {
                reg = <0>;
            };
        };
    };
}
    
```



```
};  
};  
};  
  
lvds0: lvds0@0001000 { /* lvds节点 */  
...  
ports {  
#address-cells = <1>;  
#size-cells = <0>;  
lvds0_in: port@0 { /* lvds的in端口列表 */  
#address-cells = <1>;  
#size-cells = <0>;  
reg = <0>;  
lvds0_in_tcon0: endpoint@0 { /* lvds的in端口 */  
reg = <0>;  
remote-endpoint = <&tcon0_out_lvds0>; /* 设置上一级的out节点 */  
};  
};  
  
lvds0_out: port@1 { /* lvds的out端口列表 */  
reg = <1>;  
lvds_panel_out: endpoint@0 { /* lvds的out端口 */  
reg = <0>;  
remote-endpoint = <&lvds_panel_in>; /* 设置下一级的in节点 */  
};  
};  
};  
};  
  
lvds_panel: lvds_panel@0 { /* lvds_panel节点 */  
...  
port {  
lvds_panel_in: endpoint { /* lvds_panel_in端口 */  
remote-endpoint = <&lvds_panel_out>; /* 设置上一级的out节点 */  
};  
};  
};  
};
```

drm 框架的显示通路连接关系通过 remote-endpoint 表示，每个显示组件节点下定义了 ports 子节点，下面又分别有输入（in）和输出（out）子节点，其中 in 节点的 remote-endpoint 连接到上一级的 out 节点，out 节点的 remote-endpoint 连接到下一级的 in 节点。

一级设备可有多个输入输出连接点，如某个 TCON 的输入数据既可以从 DE0 来，也可以从 DE1 来，输出既可以连接到 LVDS，也可以连接到 MIPI DSI。默认设置 in 节点 reg 为 0，out 节点 reg 为 1。

3 背光

目前常见使用的背光驱动是 pwm 背光，这里只介绍 pwm 背光的相关配置，如需使用别的背光驱动，可按照对应驱动的要求完成 config 和 dts 配置。

3.1 menuconfig 配置

```
Device Drivers
Graphics support --->
Backlight & LCD device support --->
  *- Generic PWM based Backlight Driver
```

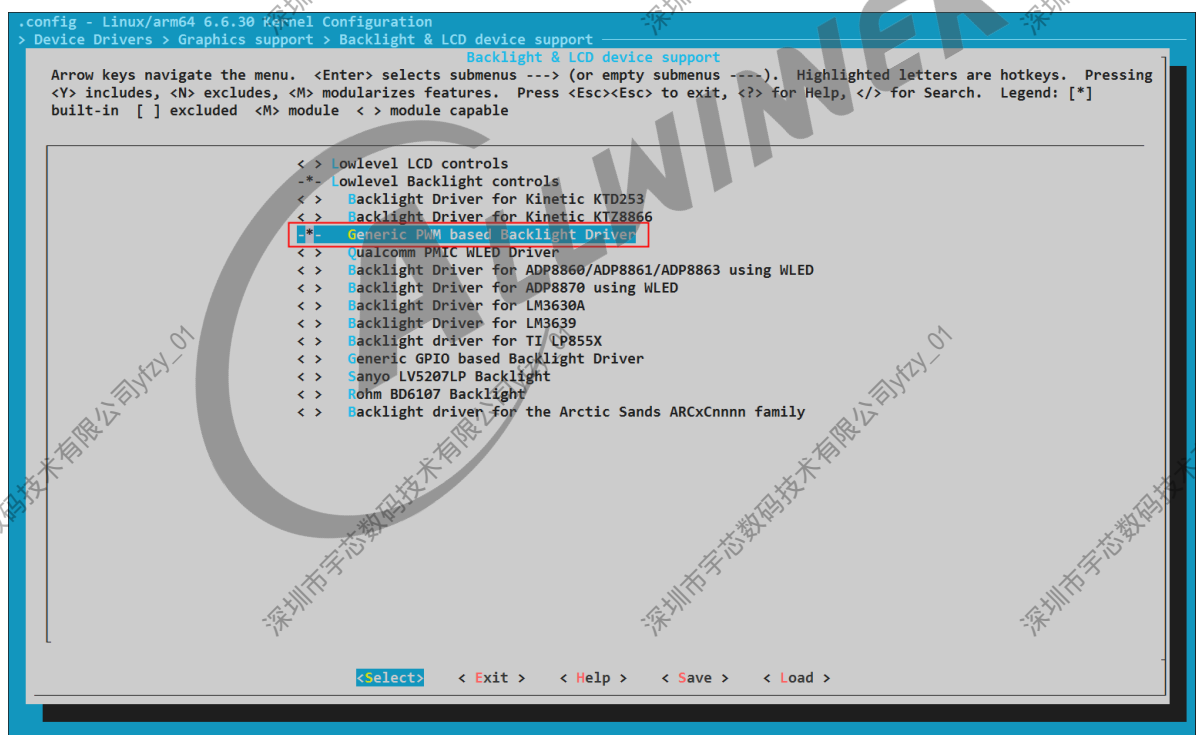


图 3-1: pwm 背光 menuconfig

3.2 defconfig 设置

```
CONFIG_BACKLIGHT_CLASS_DEVICE=y
```

3.3 board.dts 配置

```
/{
backlight0: backlight0 {
    compatible = "pwm-backlight";
    status = "okay";
    brightness-levels = <
        0 1 2 3 4 5 6 7
        8 9 10 11 12 13 14 15
        16 17 18 19 20 21 22 23
        24 25 26 27 28 29 30 31
        32 33 34 35 36 37 38 39
        40 41 42 43 44 45 46 47
        48 49 50 51 52 53 54 55
        56 57 58 59 60 61 62 63
        64 65 66 67 68 69 70 71
        72 73 74 75 76 77 78 79
        80 81 82 83 84 85 86 87
        88 89 90 91 92 93 94 95
        96 97 98 99 100 101 102 103
        104 105 106 107 108 109 110 111
        112 113 114 115 116 117 118 119
        120 121 122 123 124 125 126 127
        128 129 130 131 132 133 134 135
        136 137 138 139 140 141 142 143
        144 145 146 147 148 149 150 151
        152 153 154 155 156 157 158 159
        160 161 162 163 164 165 166 167
        168 169 170 171 172 173 174 175
        176 177 178 179 180 181 182 183
        184 185 186 187 188 189 190 191
        192 193 194 195 196 197 198 199
        200 201 202 203 204 205 206 207
        208 209 210 211 212 213 214 215
        216 217 218 219 220 221 222 223
        224 225 226 227 228 229 230 231
        232 233 234 235 236 237 238 239
        240 241 242 243 244 245 246 247
        248 249 250 251 252 253 254 255>;
    default-brightness-level = <73>;
    enable-gpios = <&pio PJ 26 GPIO_ACTIVE_HIGH>;
    pwms = <&pwm0 4 25000 0>;
};

};

&pio {
    ...

    pwm0_4_pins_active: pwm0@0 {
        pins = "PD22";
        function = "pwm0_4";
    };

    pwm0_4_pins_sleep: pwm0@1 {
        pins = "PD22";
        function = "gpio_in";
        bias-pull-down;
    };
};
```

```
...  
};  
  
&pwm0_4 {  
    pinctrl-names = "active", "sleep";  
    pinctrl-0 = <&pwm0_4_pins_active>;  
    pinctrl-1 = <&pwm0_4_pins_sleep>;  
    status = "okay";  
};
```



```
backlight0: backlight0 {
    compatible = "pwm-backlight";
    status = "okay";
    brightness-levels = <
        0   1   2   3   4   5   6   7
        8   9  10  11  12  13  14  15
        16  17  18  19  20  21  22  23
        24  25  26  27  28  29  30  31
        32  33  34  35  36  37  38  39
        40  41  42  43  44  45  46  47
        48  49  50  51  52  53  54  55
        56  57  58  59  60  61  62  63
        64  65  66  67  68  69  70  71
        72  73  74  75  76  77  78  79
        80  81  82  83  84  85  86  87
        88  89  90  91  92  93  94  95
        96  97  98  99 100 101 102 103
        104 105 106 107 108 109 110 111
        112 113 114 115 116 117 118 119
        120 121 122 123 124 125 126 127
        128 129 130 131 132 133 134 135
        136 137 138 139 140 141 142 143
        144 145 146 147 148 149 150 151
        152 153 154 155 156 157 158 159
        160 161 162 163 164 165 166 167
        168 169 170 171 172 173 174 175
        176 177 178 179 180 181 182 183
        184 185 186 187 188 189 190 191
        192 193 194 195 196 197 198 199
        200 201 202 203 204 205 206 207
        208 209 210 211 212 213 214 215
        216 217 218 219 220 221 222 223
        224 225 226 227 228 229 230 231
        232 233 234 235 236 237 238 239
        240 241 242 243 244 245 246 247
        248 249 250 251 252 253 254 255>;
    default-brightness-level = <73>;
    pwms = <&pwm0 4 25000 0>;
};
```

图 3-2: pwm 背光 dts

表 3-1: 背光属性介绍

属性	介绍
compatible	匹配 pwm 驱动，必须为 “pwm-backlight”
status	开启驱动，必须为 “okay”
brightness-levels	背光线性值表，总共 256 个值， 无特殊要求按照 0-255 填写即可，有非线性要求可按需求自行修改中间值
default-brightness-level	背光默认值，即背光开启后的默认亮度值
enable-gpios	背光使能 pin，根据实际原理图配置
pwms	pwm 通道配置，根据实际原理图配置， 四个参数依次为：使用哪组 pwm pwm 号 pwm 波周期 pwm 波极性 如无特殊需求 pwm 周期按照模板配置 25000，极性按照模板配为为 0

说明

1. 适配背光节点前，如 dts 中没有对应 pwm 节点的配置，需要手动加上 pwm 节点配置，完成 pwm 的 pinctrl 引用，具体可以从同个 dts 中别的 pwm 节点拷贝修改而来，具体可以参考上文例子。
2. 完成背光节点的增添后，还需确保背光节点被屏节点正确引用，即在屏幕节点下要添加以下属性：backlight = <xxxx>;
xxxx 为添加的背光节点名。

3.4 uboot-board.dts 配置

将 board.dts 的背光配置拷贝即可

4 MIPI DSI

4.1 menuconfig 配置

4.1.1 dsi 控制器配置

```
Allwinner BSP --->
Device Drivers --->
  DRM Drivers --->
    [*] Support DSI Output
```

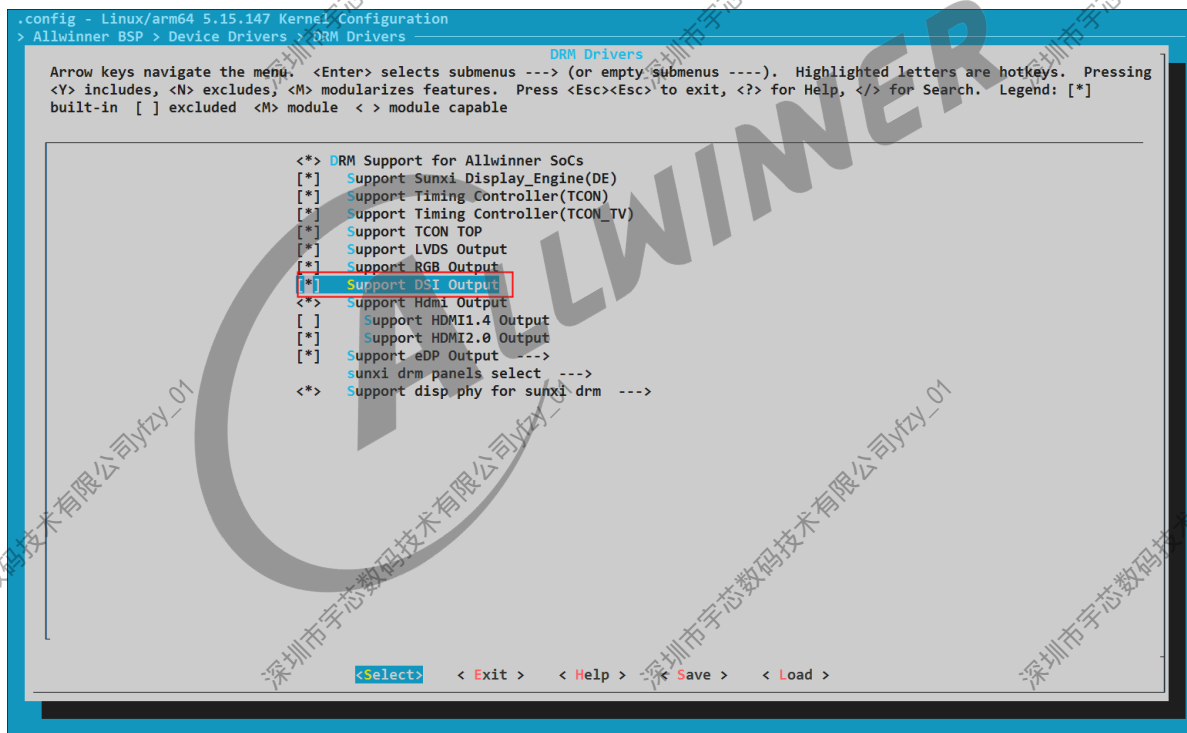


图 4-1: dsi menuconfig

4.1.2 dsi 屏配置

```
Allwinner BSP --->
Device Drivers --->
  DRM Drivers --->
    sunxi drm panels select --->
      <?> dsi general drm panel
```

4.1.3 defconfig 设置

```
CONFIG_AW_DRM=y  
CONFIG_AW_DRM_DSI=y  
CONFIG_PANEL_DSI_GENERAL=y  
CONFIG_DRM=y
```

4.2 board.dts 配置

4.2.1 panel

单 dsi panel 如下:

```
panel_0: panel_0@0 {  
    compatible = "allwinner,panel-dsi";  
    status = "okay";  
  
    power-delay-ms = <10>;  
    power-num = <2>;  
    power0-supply = <&reg_dc1sw1>;  
    power1-supply = <&reg_blldo2>;  
  
    gpio-num = <2>;  
    enable0-gpios = <&pio PH 9 GPIO_ACTIVE_HIGH>;  
    enable1-gpios = <&pio PH 10 GPIO_ACTIVE_HIGH>;  
    enable-delay-ms = <10>;  
  
    reset-gpios = <&pio PD 21 GPIO_ACTIVE_HIGH>;  
    reset-on-sequence = <1 10>, <0 20>, <1 100>;  
    reset-off-sequence = <0 100>;  
  
    backlight = <&backlight0>;  
    dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_ASYNC_INCELL)>;  
    dsi,lanes = <4>;  
    dsi,format = <0>;  
    panel-init-sequence = [  
        15 00 02 30 01  
        39 00 05 78 49 61 02 00  
        15 00 02 30 02  
        15 00 02 31 11  
        15 00 02 32 08  
        15 00 02 33 24  
        15 00 02 3d 42  
        15 00 02 3e ff  
        15 00 02 3f 30  
        15 00 02 42 81  
        15 00 02 43 30  
        15 00 02 44 21  
        15 96 02 11 00  
        15 32 02 29 00  
    ];  
    panel-exit-sequence = [  
        05 00 01 28  
    ];
```

```

    05 78 01 10
];

display-timings {
    native-mode = <&timing0>;
    timing0: timing0 {
        clock-frequency = <170422560>;
        hback-porch = <28>;
        hactive = <1200>;
        hfront-porch = <40>;
        hsync-len = <8>;
        vback-porch = <38>;
        vactive = <2000>;
        vfront-porch = <180>;
        vsync-len = <8>;
    };
};

port {
    panel0_in: endpoint@0 {
        reg = <0>;
        remote-endpoint = <&panel_output_0>;
    };
};
};

```

dual-dsi panel 如下:

```

panel_0: panel_0@0 {
    compatible = "allwinner,panel-dsi";
    status = "okay";
    reg = <0>;

    power-delay-ms = <10>;
    power-num = <2>;
    power0-supply = <&reg_dc1sw1>;
    power1-supply = <&reg_bldo2>;
    gpio-num = <1>;
    enable0-gpios = <&pio PD 23 GPIO_ACTIVE_HIGH>;
    enable-delay-ms = <10>;

    reset-gpios = <&pio PD 21 GPIO_ACTIVE_HIGH>;
    reset-on-sequence = <1 10> <0 20>, <1 100>;
    reset-off-sequence = <0 100>;

    backlight = <&backlight0>;
    dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_SLAVE_MODE)>;
    dsi,lanes = <4>;
    dsi,format = <0>;
    panel-init-sequence = [
        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 = <268627200>;
    hback-porch = <48>;
    hactive = <2560>;
    hfront-porch = <80>;
    hsync-len = <32>;
    vback-porch = <31>;
    vactive = <1600>;
    vfront-porch = <9>;
    vsync-len = <6>;

};

};
port {
    panel0_in_0: endpoint@0 {
        reg = <0>;
        remote-endpoint = <&dsi0_panel_output_0>;
    };
    panel0_in_1: endpoint@1 {
        reg = <1>;
        remote-endpoint = <&dsi1_panel_output_0>;
    };
};
};

```

表 4-1: dsi 屏驱动属性介绍

属性	介绍
compatible	匹配全志的 dsi 屏驱动，必须为 “allwinner,panel-dsi”
status	开启驱动，必须为 “okay”
powerX-supply	可选，屏幕供电，默认支持三个供电，即 X 范围为 0/1/2，根据实际原理图配置
power-num	可选，默认是 3，即控制 powerX-supply 的 X，可根据实际原理图配置数量。
power-delay-ms	可选，powerX-supply 之间的开启间隔，根据屏幕实际上电时序配置
enableX-gpios	可选，屏幕使能或供电使能，默认支持三个 GPIO，即 X 范围为 0/1/2，根据实际原理图配置
gpio-num	可选，默认是 3，即控制 enableX-gpios 的 X，可根据实际原理图配置数量。
enable-delay-ms	可选，enableX-gpios 之间的拉高间隔，根据屏幕实际上电时序配置
reset-gpios	可选，屏幕复位，根据实际原理图配置
reset-on-sequence	可选，屏幕上电时，复位引脚的电压连续变化情况，根据屏幕实际上电时序配置 <1 10>, <0 20>, <1 100>; 意思是拉高 10ms -> 拉低 20ms -> 拉高 100ms -> 发送 panel-init-sequence 指令

属性	介绍
reset-off-sequence	可选，屏幕下电时，复位引脚的电压连续变化情况，根据屏幕实际下电时序配置 <0 100>; 意思是拉低 100ms -> 关闭 powerX-supply 电
backlight	屏幕背光
panel-id-value	屏幕 id 寄存器和值，使用多屏兼容必配屏幕 id 寄存器和值，使用多屏兼容必配
dsi,flags	dsi 模式配置，具体参数配置方式参考下文说明。
dsi,lanes	Lane Number (1 ~ 4)
dsi,format	Pixel Format
panel-init-sequence	屏的上电初始化序列，具体参数配置方式参考 Command 说明。
panel-exit-sequence	屏的下电初始化序列，具体参数配置方式参考 Command 说明。
panel-timings	LCD 时序参数，按屏规格书填写。
port	必须配置，连接关系配置， remote-endpoint 需引用 dsi/panel 节点下的 out 子节点, 参考软件 dts 连接关系章节

4.2.1.1 dsi,flags

MIPI_DSI_MODE_VIDEO 必须有，其它模式可进行叠加，支持如下模式：

MIPI_DSI_MODE_VIDEO_BURST: burst 模式，必选；

MIPI_DSI_SYNC_INCELL: incell 屏时，lcd 驱动上电完成后等待 tp 初始化完成后再送显示，必选；

MIPI_DSI_ASYNC_INCELL: incell 屏时，lcd 驱动上电完成后，只通知 tp，不等待 tp 初始化完成就直接送显示，必选；

MIPI_DSI_SLAVE_MODE: **dual-dsi、使用 dsc 时**，必选；

MIPI_DSI_CLOCK_NON_CONTINUOUS: 使能 dsi 非连续时钟；

MIPI_DSI_MODE_NO_EOT_PACKET: 使能 EOTP (EoT packet) 特性；

4.2.1.2 Command

```
panel-init-sequence = [  
  ...  
  39 00 05 78 49 61 02 00  
  15 00 02 30 02  
  15 00 02 31 11  
  15 00 02 32 08  
  15 00 02 33 24  
  ...  
  15 96 02 11 00  
  15 32 02 29 00  
];  
  
panel-exit-sequence = [  
  05 00 01 28  
  05 78 01 10  
];
```

格式说明：头部 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 Payload。

例如：

```
39 00 05 78 49 61 02 00
```

解析如下：

Data Type: 0x39 (DCS Long Write)

Delay: 0x00 (0 ms)

Payload Length: 0x05 (4 Bytes)

Payload: 0x78 0x49 0x61 0x02 0x00 (0x78 是要写入的寄存器地址)

Data Type

Data Type,		Description	Packet Size	DCS	VD PKT	GN PKT
(hex)	(binary)					
01h	00 0001	Sync Event, V Sync Start	Short		0	
11h	01 0001	Sync Event, V Sync End	Short		0	
21h	10 0001	Sync Event, H Sync Start	Short		0	
31h	11 0001	Sync Event, H Sync End	Short		0	
08h	00 1000	End of Transmission Packet	Short	0	0	0
02h	00 0010	Color Mode (CM) Off Command	Short		0	
12h	01 0010	Color Mode (CM) On Command	Short		0	
22h	10 0010	Shut Down Peripheral Command	Short		0	
32h	11 0010	Turn On Peripheral Command	Short		0	
03h	00 0011	Generic Short WRITE, no parameters	Short			NOF
13h	01 0011	Generic Short WRITE, 1 parameter	Short			0
23h	10 0011	Generic Short WRITE, 2 parameters	Short			0
04h	00 0100	Generic READ, no parameters	Short			NOF
14h	01 0100	Generic READ, 1 parameter	Short			0
24h	10 0100	Generic READ, 2 parameters	Short			0
05h	00 0101	DCS WRITE, no parameters	Short	0	0	0
15h	01 0101	DCS WRITE, 1 parameter	Short	0	0	0
06h	00 0110	DCS READ, no parameters	Short	0	0	0
37h	11 0111	Set Maximum Return Packet Size	Short	0	0	0
09h	00 1001	Null Packet, no data	Long	0	0	0
19h	01 1001	Blanking Packet, no data	Long		0	
29h	10 1001	Generic Long Write	Long			0
39h	11 1001	DCS Long Write/write_LUT Command Packet	Long	0	0	0
0Eh	00_1110	Packet Pixel Stream, 16bit RGB 5-6-5 Format (Support for 1 and 2 data lanes mode)	Long		0	
1Eh	01_1110	Packet Pixel Stream, 18bit RGB 6-6-6 Format (Support for 1 and 2 data lanes mode)	Long		0	
2Eh	10_1110	Packet Pixel Stream, 18bit RGB Loosely 6-6-6 Format (Support for 1 and 2 data lanes mode)	Long		0	
3Eh	11 1110	Packed Pixel Stream, 24-bit RGB, 8-8-8 Format (Support for 1, 2 and 3 data lanes mode)	Long		0	
0h&Fh,	xx 0000 xx 1111	DO NOT USE All unspecified codes are reserved				

图 4-2: data-type

1、DCS Write

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 $n \geq 2$ ，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。

2、Generic Write

Generic 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 $n \geq 3$ ，将以 Long Packet 的形式进行对 Payload 打包，表示 n parameters，Data Type 为 0x29。

4.2.1.3 dsc

使用的平台支持 dsc 时，dsc 使用配置如下：

```
panel_1: panel_1@0 {
    compatible = "allwinner,panel-dsi";
    status = "okay";
    ...
    dsc,status = "okay";
    dsc,slice-height = <8>;
    dsc,slice-width = <600>;
    dsc,bits-per-component = <8>;
    dsc,block-pred-enable = <0>;
    dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_SLAVE_MODE)>;
    dsi,lanes = <4>;
    dsi,format = <0>;
    ...
    display-timings {
        native-mode = <&panel1_timing0>;
        panel1_timing0: panel1_timing0 {
            clock-frequency = <351000000>;
            hback-porch = <120>;
            hactive = <1200>;
            hfront-porch = <180>;
            hsync-len = <72>;
            vback-porch = <200>;
            vactive = <2000>;
            vfront-porch = <250>;
            vsync-len = <30>;
        };
    };
    ...
};
```

表 4-2: dsc 模块属性介绍

属性	介绍
dsc,status	开启 dsc 模块，必须为 “okay”
dsc,slice-height	图像在压缩过程中每个切片的高度，具体配置多少咨询屏厂
dsc,slice-width	图像在压缩过程中每个切片的宽度，具体配置多少咨询屏厂
dsc,bits-per-component	像素压缩后的位深：8/10/12

属性	介绍
----	----

dsc,block-pred-enable	块预测启用配置，是否开启咨询屏厂开启：1；关闭：0
-----------------------	---------------------------

pps 可以不用填入 dts 中，驱动会自行按照协议计算具体的 pps 发给屏端的。

4.2.2 dsi

表 4-3: dsi 模块属性介绍

属性	介绍
status	开启驱动，必须为 “okay”
ports	必须配置，连接关系配置，remote-endpoint 需引用 dsi panel 下的 in 子节点，参考软件 dts 连接关系章节

4.2.2.1 dsi0 接口

```
&dsi0 {
    status = "okay";
    pinctrl-0 = <&dsi0_4lane_pins_a>;
    pinctrl-1 = <&dsi0_4lane_pins_b>;
    pinctrl-names = "active","sleep";

    ports {
        dsi0_out: port@1 {
            dsi_out_panel: endpoint {
                remote-endpoint = <&panel_input>;
            };
        };
    };
    panel: panel@0 {
        compatible = "allwinner,virtual-panel";
        status = "okay";
        reg = <0>;
        ports {
            panel_in: port@0 {
                reg = <0>;
                panel_input: endpoint@0 {
                    reg = <0>;
                    remote-endpoint = <&dsi_out_panel>;
                };
            };
            panel_out: port@1 {
                reg = <1>;
            };
        };
    };
};
```

```

        panel_output_0: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&panel0_in>;
        };
    };
};
};

```

4.2.2.2 dsi1 接口

```

&dsi1 {
    status = "okay";
    pinctrl-0 = <&dsi0_4lane_pins_a>;
    pinctrl-1 = <&dsi0_4lane_pins_b>;
    pinctrl-names = "active", "sleep";

    ports {
        dsi1_out: port@1 {
            dsi1_out_panel: endpoint {
                remote-endpoint = <&panel_input>;
            };
        };
    };
    panel: panel@0 {
        compatible = "allwinner,virtual-panel";
        status = "okay";
        reg = <0>;
        ports {
            panel_in: port@0 {
                reg = <0>;
                panel_input: endpoint@0 {
                    reg = <0>;
                    remote-endpoint = <&dsi1_out_panel>;
                };
            };
            panel_out: port@1 {
                reg = <1>;
                panel_output_0: endpoint@0 {
                    reg = <0>;
                    remote-endpoint = <&panel0_in>;
                };
            };
        };
    };
};

```

4.2.2.3 dual-dsi 接口

```

&dsi0 {
    status = "okay";
    pinctrl-0 = <&dsi0_4lane_pins_a>;
    pinctrl-1 = <&dsi0_4lane_pins_b>;
    pinctrl-names = "active", "sleep";

```

```
dual-channel = <&dsi1>;

ports {
    dsi0_out: port@1 {
        dsi0_out_panel: endpoint {
            remote-endpoint = <&dsi0_panel_input>;
        };
    };
};

dsi0_panel: panel@0 {
    compatible = "allwinner,virtual-panel";
    status = "okay";
    reg = <0>;
    ports {
        dsi0_panel_in: port@0 {
            reg = <0>;
            dsi0_panel_input: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&dsi0_out_panel>;
            };
        };
        dsi0_panel_out: port@1 {
            reg = <1>;
            dsi0_panel_output_0: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&panel0_in_0>;
            };
        };
    };
};

&dsi1 {
    status = "okay";
    pinctrl-0 = <&dsi1_4lane_pins_a>;
    pinctrl-1 = <&dsi1_4lane_pins_b>;
    pinctrl-names = "active","sleep";

    ports {
        dsi1_out: port@1 {
            dsi1_out_panel: endpoint {
                remote-endpoint = <&dsi1_panel_input>;
            };
        };
    };
};

dsi1_panel: panel@0 {
    compatible = "allwinner,virtual-panel";
    status = "okay";
    reg = <0>;
    ports {
        dsi1_panel_in: port@0 {
            reg = <0>;
            dsi1_panel_input: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&dsi1_out_panel>;
            };
        };
        dsi1_panel_out: port@1 {
            reg = <1>;
```

```
    dsi1_panel_output_0: endpoint@0 {
        reg = <0>;
        remote-endpoint = <&panel0_in_1>;
    };
};
};
};
```

4.2.2.4 多屏兼容

用来做兼容的屏必须要有屏 ID，ID 地址及 ID 值不能与其它屏有冲突，适配步骤如下：

步骤 1：先单独点亮所有屏幕。

步骤 2：panel_X 节点中加上 ID 属性：panel-id-value=[\${id_reg} \${id_value}]; 查看 uboot 打印读到屏端的具体值是多少：dsi rx panel id 0x4: 0x99；按照 soc 实际读回来的值更新 dts 中的 \${id_value}；若对该值有疑问找屏厂咨询。

步骤 3：在 virtual-panel 驱动设备树的 panel_out 节点中按顺序 0、1、2、... 依次添加 endpoint，其中 remote-endpoint 分别引用 panel_X 下的 in 子节点；

步骤 4：若想关闭某一兼容屏，只需修改该兼容屏节点 panel_X 的 status = "disabled"; 即可，其它不用动。

参考配置如下：

```
/{
    panel_0: panel_0@0 {
        compatible = "allwinner,panel-dsi";
        status = "okay";
        panel-id-value = [04 66];

        port {
            panel0_in: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&panel_output_0>;
            };
        };
    };

    panel_1: panel_1@0 {
        compatible = "allwinner,panel-dsi";
        status = "okay";
        panel-id-value = [04 99];
        ...

        port {
            panel1_in: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&panel_output_1>;
            };
        };
    };
};
```



```
&dlcd1 {  
    status = "okay";  
};
```

4.2.4 combophy

使用 dsi0 接口：

```
&dsi0combophy {  
    status = "okay";  
};
```

使用 dual-dsi 接口：

```
&dsi0combophy {  
    status = "okay";  
};  
  
&dsi1combophy {  
    status = "okay";  
};
```

使用 dsi1 接口：

```
&dsi1combophy {  
    status = "okay";  
};
```

4.2.5 video_out

dsi0、dsi1、dual-dsi0/1 显示：

```
&vo0 {  
    status = "okay";  
};
```

4.2.6 pinctl

1、dsi0

```
pinctrl-0 = <&dsi0_4lane_pins_a>;  
pinctrl-1 = <&dsi0_4lane_pins_b>;
```

2、dsi1

```
pinctrl-0 = <&dsi1_4lane_pins_a>;  
pinctrl-1 = <&dsi1_4lane_pins_b>;
```

4.3 常见问题

1、使用 burst mode 时，屏幕无显示：

- 先用 video mode 把屏幕点亮

```
dsi,flags = <(MIPI_DSI_MODE_VIDEO)>;
```

- 在 video mode 能正常显示的基础上将 hfront-porch 加 200，然后再切换到 burst mode

```
dsi,flags = <(MIPI_DSI_MODE_VIDEO | MIPI_DSI_MODE_VIDEO_BURST)>;
```

2、重启样机屏幕概率无显示：

- 增加屏端 reset 次数或 delay 时间：

```
reset-on-sequence = <1 10>, <0 20>, <1 10>; ===》 reset-on-sequence = <1 20>, <0 30>, <1 50>;
```

- 在 0x11、0x29 指令之前及之后的指令增加 0x0f 延时，如下所示：

```
1、
panel-init-sequence = [
...
15 0F 02 33 24
15 96 02 11 00
15 32 02 29 00
];

2、
panel-init-sequence = [
...
15 0F 02 33 24
15 96 02 11 00
15 0F 02 32 08
15 32 02 29 00
];

3、
panel-init-sequence = [
...
15 0F 02 33 24
15 96 02 11 00
15 0F 02 32 08
15 32 02 29 00
15 0F 02 31 06
...
];
```

3、其它平台的屏参在 AW 无法点亮/花屏或图像上下偏移

- 无法点亮/花屏：AW 平台的 hblank 相较其它平台需要配大一点，自行将 hbp、hfp 加个 10~70 左右，还是不行找屏厂重新要套屏参；
- 图像上下偏移：vbp 配置太大导致。



5 LVDS

5.1 menuconfig 配置

5.1.1 lvds 控制器配置

```
Allwinner BSP --->
Device Drivers --->
  DRM Drivers --->
    [*] Support LVDS Output
```

5.1.2 lvds 屏配置

```
Allwinner BSP --->
Device Drivers --->
  DRM Drivers --->
    sunxi drm panels select --->
      <*> lvds general drm panel
```

5.1.3 defconfig 设置

```
CONFIG_AW_DRM=y
CONFIG_AW_DRM_LVDS=y
CONFIG_PANEL_LVDS_GENERAL=y
CONFIG_DRM=y
```

5.2 board.dts 配置

5.2.1 panel

```
lvds_panel: lvds_panel@0 {
    compatible = "sunxi-lvds";
    status = "okay";
    power-delay-ms = <10>;
    power0-supply = <&reg_dc1sw1>;
    power1-supply = <&reg_bldo2>;
    backlight = <&backlight0>;
```

```

bus-format = <MEDIA_BUS_FMT_RGB888_1X7X4_SPWG>;
enable0-gpios = <&pio PD 21 GPIO_ACTIVE_HIGH>;
display-timings {
    native-mode = <&lvds0_timing0>;
    lvds0_timing0: timing0 {
        clock-frequency = <74871600>;
        hback-porch = <70>;
        hactive = <1280>;
        hfront-porch = <83>;
        hsync-len = <18>;
        vback-porch = <13>;
        vactive = <800>;
        vfront-porch = <37>;
        vsync-len = <10>;
    };
};
port {
    lvds_panel_in: endpoint {
        remote-endpoint = <&lvds_panel_out>;
    };
};
};

```

表 5-1: lvds 屏驱动属性介绍

属性	介绍
compatible	匹配全志的 lvds 屏驱动，必须为 “sunxi-lvds”
status	开启驱动，必须为 “okay”
powerX-supply	可选，屏幕供电，最大支持三个供电，即 X 范围为 0/1/2，根据实际原理图配置
enableX-gpios	可选，屏幕使能或供电使能，最大支持三个 GPIO，即 X 范围为 0/1/2，根据实际原理图配置
backlight	屏幕背光
bus-format	LVDS 信号的数据映射方式。MEDIA_BUS_FMT_RGB888_1X7X4_SPWG: ves-a-24; MEDIA_BUS_FMT_RGB888_1X7X4_JEIDA: jeida-24
panel-timings	可选，用户自定义分辨率，驱动默认直接从 EDID 解析分辨率 mode，配上后会追加用户自定义的 mode
ports	必须配置，连接关系配置，remote-endpoint 需引用 lvds 节点下的 out 子节点，参考软件 dts 连接关系章节

5.2.2 lvds

使用 lvds0、lvds1、dual-lvds0+1 接口：

```

&lvds0 {
    status = "okay";
    dual-channel = <0>;
    pinctrl-0 = <&lvds0_pins_a>;
    pinctrl-1 = <&lvds0_pins_b>;
    pinctrl-names = "active","sleep";
    ports {
        port@1 {
            reg = <1>;
            lvds_panel_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&lvds_panel_in>;
            };
        };
    };
};

```

使用 lvds2、lvds3、dual-lvds2+3 接口：

```

&lvds1 {
    status = "okay";
    dual-channel = <0>;
    pinctrl-0 = <&lvds2_pins_a>;
    pinctrl-1 = <&lvds2_pins_b>;
    pinctrl-names = "active","sleep";
    ports {
        port@1 {
            reg = <1>;
            lvds_panel_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&lvds_panel_in>;
            };
        };
    };
};

```

表 5-2: lvds 模块属性介绍

属性	介绍
status	开启驱动，必须为“okay”
dual-channel	选择显示接口；0: lvds0、lvds2；1: dual-lvds0+1、dual-lvds2+3；2: lvds1、lvds3；
ports	必须配置，连接关系配置，remote-endpoint 需引用 lvds panel 下的 in 子节点，参考软件 dts 连接关系章节

5.2.3 tcon

使用 lvds0、lvds1、dual-lvds0+1 接口：

```
&dlcd0 {  
    status = "okay";  
};
```

使用 lvds2、lvds3、dual-lvds2+3 接口：

```
&dlcd2 {  
    status = "okay";  
};
```

5.2.4 combophy

使用 lvds0 接口：

```
&dsi0combophy {  
    status = "okay";  
};
```

使用 lvds1 接口：

```
&dsi0combophy {  
    status = "okay";  
};
```

使用 dual-lvds0+1 接口：

```
&dsi0combophy {  
    status = "okay";  
};  
  
&dsi1combophy {  
    status = "okay";  
};
```

lvds2、lvds3、dual-lvds2+3 接口不需要 combophy 驱动。

5.2.5 video_out

lvds0、lvds1、lvds0+1 显示：

```
&vo0 {  
    status = "okay";  
};
```

lvds2、lvds3、lvds2+3 显示：

```
&vo1 {  
    status = "okay";  
};
```

5.2.6 pinctl

1、lvds0

```
pinctrl-0 = <&lvds0_pins_a>;  
pinctrl-1 = <&lvds0_pins_b>;
```

2、lvds1

```
pinctrl-0 = <&lvds1_pins_a>;  
pinctrl-1 = <&lvds1_pins_b>;
```

3、dual-lvds0+1

```
pinctrl-0 = <&lvds0_pins_a>, <&lvds1_pins_a>;  
pinctrl-1 = <&lvds0_pins_b>, <&lvds1_pins_b>;
```

4、lvds2

```
pinctrl-0 = <&lvds2_pins_a>;  
pinctrl-1 = <&lvds2_pins_b>;
```

5、lvds3

```
pinctrl-0 = <&lvds3_pins_a>;  
pinctrl-1 = <&lvds3_pins_b>;
```

6、dual-lvds2+3

```
pinctrl-0 = <&lvds2_pins_a>, <&lvds3_pins_a>;  
pinctrl-1 = <&lvds2_pins_b>, <&lvds3_pins_b>;
```

5.3 常见问题

1、显示颜色异常：

LVDS 信号的数据映射方式错误：修改 bus-format 值为另外一个

2、屏幕上有异常条纹

用示波器查看 data lane 波形是否异常。

6 RGB

6.1 menuconfig 配置

6.1.1 rgb 控制器配置

```
Allwinner BSP --->
Device Drivers --->
  DRM Drivers --->
    [*] Support RGB Output
```

6.1.2 rgb 屏配置

```
Allwinner BSP --->
Device Drivers --->
  DRM Drivers --->
    sunxi drm panels select --->
      <*> rgb general drm panel
```

6.1.3 defconfig 设置

```
CONFIG_AW_DRM=y
CONFIG_AW_DRM_RGB=y
CONFIG_PANEL_RGB_GENERAL=y
CONFIG_DRM=y
```

6.2 board.dts 配置

6.2.1 panel

```
rgb_panel0: rgb_panel@1 {
    compatible = "sunxi-rgb";
    status = "okay";
    reg = <0>;
    power0-supply = <&reg_dc4>;
    power1-supply = <&reg_cldo1>;
    backlight = <&backlight0>;
};
```

```

display-timings {
    native-mode = <&rgb0_timing0>;
    rgb0_timing0: timing0 {
        clock-frequency = <48360000>;
        hback-porch = <35>;
        hactive = <1024>;
        hfront-porch = <161>;
        hsync-len = <20>;
        vback-porch = <25>;
        vactive = <600>;
        vfront-porch = <15>;
        vsync-len = <10>;
    };
};
port {
    rgb0_panel_in: endpoint {
        remote-endpoint = <&rgb0_panel_out>;
    };
};
};

```

表 6-1: rgb 屏驱动属性介绍

属性	介绍
compatible	匹配全志的 rgb 屏驱动，必须为 “sunxi-rgb”
status	开启驱动，必须为 “okay”
powerX-supply	可选，屏幕供电，最大支持三个供电，即 X 范围为 0/1/2，根据实际原理图配置
enableX-gpios	可选，屏幕使能或供电使能，最大支持三个 GPIO，即 X 范围为 0/1/2，根据实际原理图配置
backlight	屏幕背光
panel-timings	LCD 时序参数，按屏规格书填写。
ports	必须配置，连接关系配置， remote-endpoint 需引用 rgb 节点下的 out 子节点，参考软件 dts 连接关系章节

6.2.2 rgb

使用 rgb0 接口：

```

&rgb0 {
    status = "okay";
    pinctrl-0 = <&rgb0_24pins_a>;
    pinctrl-1 = <&rgb0_24pins_b>;
    pinctrl-names = "active", "sleep";
    ports {
        port@1 {
            reg = <1>;

```

```

    rgb0_panel_out: endpoint@0 {
        reg = <0>;
        remote-endpoint = <&rgb0_panel_in>;
    };
};
};

```

使用 rgb1 接口：

```

&rgb1 {
    status = "okay";
    pinctrl-0 = <&rgb1_24pins_a>;
    pinctrl-1 = <&rgb1_24pins_b>;
    pinctrl-names = "active","sleep";
    ports {
        port@1 {
            reg = <1>;
            rgb1_panel_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&rgb1_panel_in>;
            };
        };
    };
};

```

表 6-2: rgb 模块属性介绍

属性	介绍
status	开启驱动，必须为“okay”
ports	必须配置，连接关系配置，remote-endpoint 需引用 rgb panel 下的 in 子节点，参考软件 dts 连接关系章节

6.2.3 tcon

RGB 连接的 tcon 名字，可查询[硬件连接关系](#)章节

```

&dlcd0 {
    status = "okay";
};

```

6.2.4 video_out

rgb0 显示：

```
&vo0 {  
    status = "okay";  
};
```

rgb1 显示:

```
&vo1 {  
    status = "okay";  
};
```

6.2.5 pinctl

1、rgb0

```
pinctrl-0 = <&rgb0_24pins_a>;  
pinctrl-1 = <&rgb0_24pins_b>;
```

2、rgb1

```
pinctrl-0 = <&rgb1_24pins_a>;  
pinctrl-1 = <&rgb1_24pins_b>;
```

7 eDP

7.1 menuconfig 配置

7.1.1 edp 控制器配置

```
Allwinner BSP --->
Device Drivers --->
DRM Drivers --->
[*] Support eDP Output --->
   [] Support HDCP For DisplayPort
   [*] Support inno edp 1.3 controller
   [] Support trilinear edp 1.4 controller
```

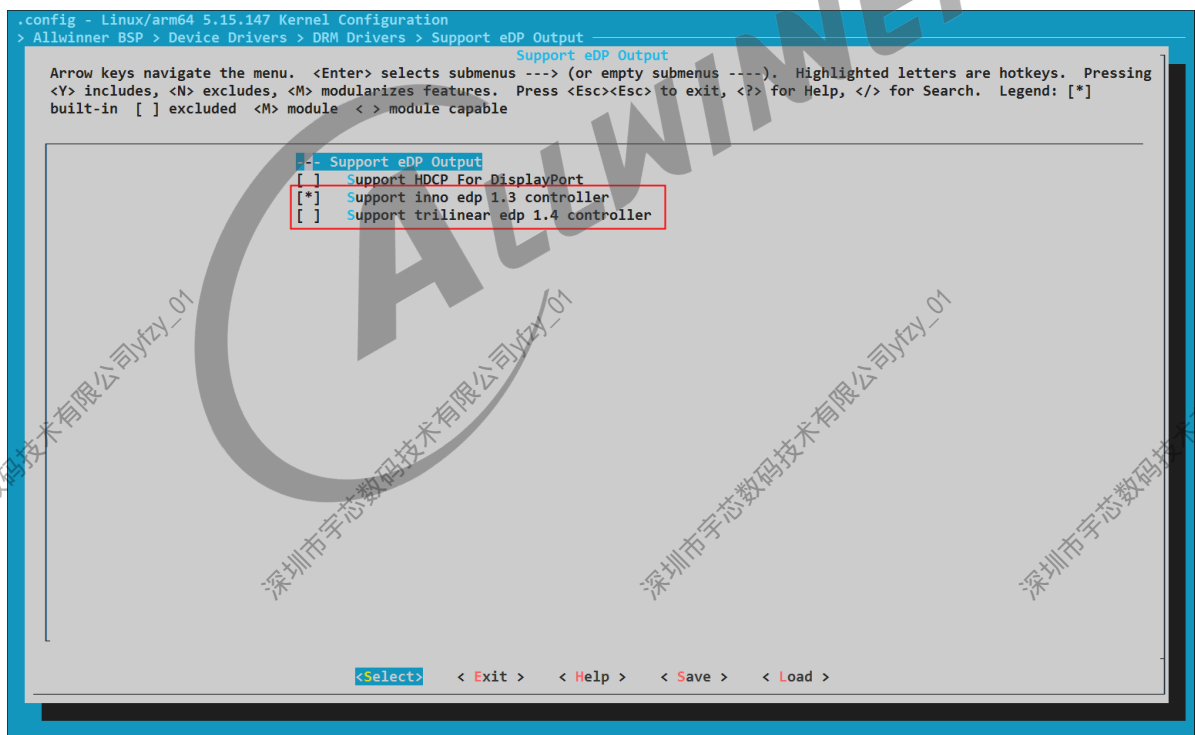


图 7-1: edp menuconfig

7.1.2 edp 屏配置

```
Allwinner BSP --->
Device Drivers ---->
DRM Drivers ---->
sunxi drm panels select ---->
<*> eDP general drm panel
```

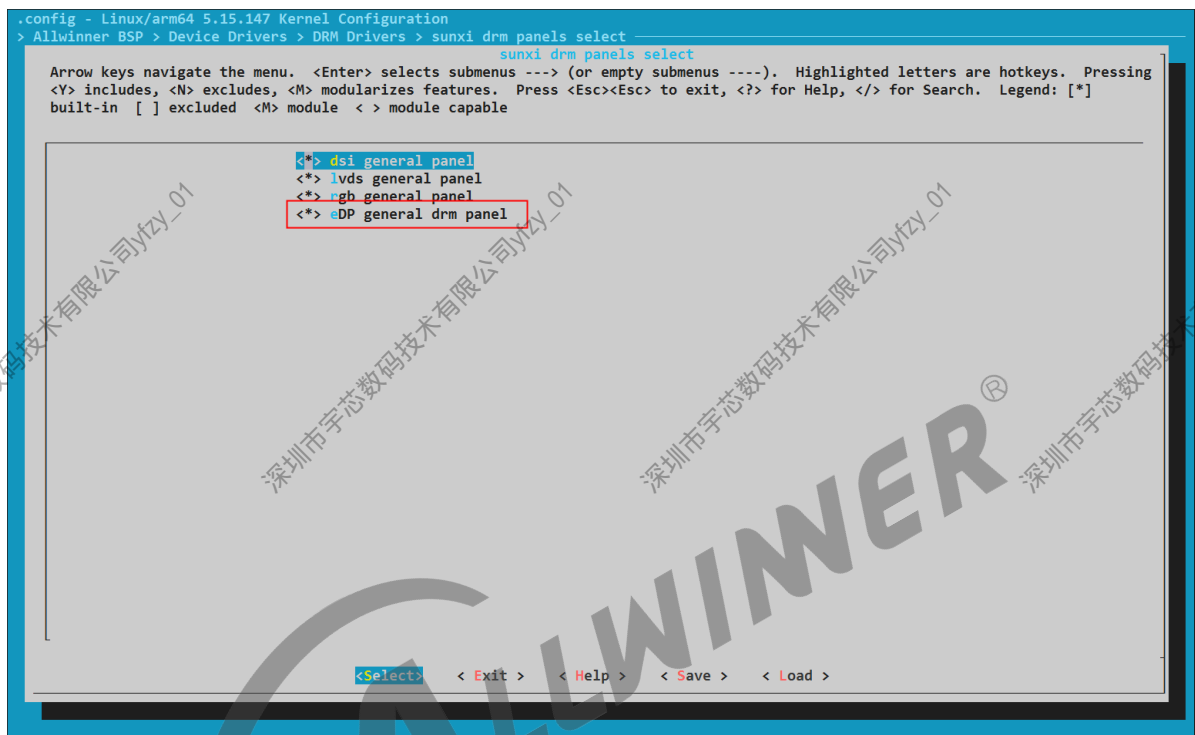


图 7-2: edp panel_menuconfig

7.1.3 defconfig 设置

```
CONFIG_AW_DRM=y
CONFIG_AW_DRM_EDP=y
CONFIG_AW_DRM_INNO_EDP13=y
CONFIG_PANEL_EDP_GENERAL=y
CONFIG_DRM=y
```

7.2 board.dts 配置

7.2.1 panel

本节只介绍全志平台的 edp 屏驱动配置方法，如需使用其他屏驱动，需按照屏驱动要求进行配置，不必参考本章节。

```
edp_panel: edp_panel {
    compatible = "edp-general-panel";
    status = "okay";
    power0-supply = <&reg_dc4>;
    enable0-gpios = <&pio PI 2 GPIO_ACTIVE_HIGH>;

    backlight = <&edp_panel_backlight>;

    panel-timing {
        clock-frequency = <348577920>; /* pixel clock */
        hactive = <2560>;
        hback-porch = <120>;
        hfront-porch = <88>;
        hsync-len = <32>;
        vactive = <1600>;
        vback-porch = <71>;
        vfront-porch = <28>;
        vsync-len = <5>;
        /* hor_sync_polarity */
        hsync-active = <1>;
        /* ver_sync_polarity */
        vsync-active = <1>;
    };
    ports {
        #address-cells = <1>;
        #size-cells = <0>;
        panel_in: port@0 {
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <0>;
            edp_panel_in: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&edp_panel_out>;
            };
        };
    };
};
```

```
edp_panel: edp_panel {
    compatible = "edp-general-panel";
    status = "okay";
    power0-supply = <&reg_dcdc4>;

    backlight = <&edp_panel_backlight>;

    panel-timing {
        clock-frequency = <348577920>; /* pixel clock */
        hactive = <2560>;
        hback-porch = <120>;
        hfront-porch = <88>;
        hsync-len = <32>;
        vactive = <1600>;
        vback-porch = <71>;
        vfront-porch = <28>;
        vsync-len = <5>;
        /* hor_sync_polarity */
        hsync-active = <1>;
        /* ver_sync_polarity */
        vsync-active = <1>;
    };
    ports {
        #address-cells = <1>;
        #size-cells = <0>;
        panel_in: port@0 {
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <0>;
            edp_panel_in: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&edp_panel_out>;
            };
        };
    };
};
```

图 7-3: edp panel_dts

表 7-1: edp 屏驱动属性介绍

属性	介绍
compatible	匹配全志的 edp 屏驱动，必须为 “edp-general-panel”
status	开启驱动，必须为 “okay”
powerX-supply	可选，屏幕供电，最大支持三个供电，即 X 范围为 0/1/2，根据实际原理图配置
enableX-gpios	可选，屏幕使能或供电使能，最大支持三个 GPIO，即 X 范围为 0/1/2，根据实际原理图配置
backlight	可选，屏幕背光，如屏幕没有单独背光控制可以不配
panel-timing	可选，用户自定义分辨率， 驱动默认直接从 EDID 解析分辨率 mode，配上后会追加用户自定义的 mode
ports	必须配置，连接关系配置， remote-endpoint 需引用 edp 节点下的 out 子节点, 参考软件 dts 连接关系章节

7.2.2 drm_edp

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

    edp_colordepth = <8>; /* 6/8/10/12/16 */
    edp_color_fmt = <0>; /* 0:RGB 1:YUV444 2:YUV422 */

    vcc-edp-supply = <&reg_bldo3>;
    vdd-edp-supply = <&reg_dcdc2>;
    ports {
        edp_out: port@1 {
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <1>;
            edp_panel_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&edp_panel_in>;
            };
        };
    };
};
```

```

&drm edp {
    status = "okay";

    edp_colordepth = <8>; /* 6/8/10/12/16 */
    edp_color_fmt = <0>; /* 0:RGB 1: YUV444 2: YUV422 */

    vcc-edp-supply = <&reg_bldo3>;
    vdd-edp-supply = <&reg_dcdc2>;
    ports {
        edp_out: port@1 {
            #address-cells = <1>;
            #size-cells = <0>;
            reg = <1>;
            edp_panel_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&edp_panel_in>;
            };
        };
    };
};

```

图 7-4: edp node_dts

表 7-2: edp 模块属性介绍

属性	介绍
status	开启驱动，必须为“okay”
edp_colordepth	必须配置，根据屏幕规格书配置屏幕的色深，常见色深有 6/8/10/12/16 bit
edp_color_fmt	必须配置，根据屏幕规格书配置屏幕的色彩空间，常见的有 RGB/YUV444/YUV422
vcc-edp-supply	可选，edp 控制器供电，根据原理图 SOC 的 edp 模块实际供电配置
vdd-edp-supply	可选，edp 控制器供电，根据原理图 SOC 的 edp 模块实际供电配置
ports	必须配置，连接关系配置， remote-endpoint 需引用 edp panel 下的 in 子节点，参考软件 dts 连接关系章节

7.2.3 tcon

edp 连接的 tcon 名字，可查询[硬件连接关系](#)章节

```
&tv1 {
    status = "okay";
};
```

```
&tv1 {
    status = "okay";
};
```

图 7-5: edp_tcon_dts

表 7-3: edp_tcon 模块属性介绍

属性	介绍
status	开启驱动，必须为“okay”

7.2.4 video_out

edp 所属的 video_out 名字，可查询[硬件连接关系](#)章节。部分平台没有 video_out 这一级，可以不用配置。

```
&vo0 {
    status = "okay";
};
```

```
&vo0 {
    status = "okay";
};
```

图 7-6: edp_vo_dts

表 7-4: edp vo 模块属性介绍

属性	介绍
status	开启驱动，必须为“okay”

7.2.5 combophy

edp 模块使用的外部 PHY 名字，可查询[硬件连接关系](#)章节。部分平台没有 PHY 这一级，可以不用配置。

```
&serdes {
    lane_invert = <1 1 1 1>;
    lane_remap = <2 3 0 1>;
    status = "okay";
};
```

```
&serdes {
    lane_invert = <0 1 1 1>;
    lane_remap = <2 3 0 1>;
    status = "okay";
};
```

图 7-7: edp_phy_dts

表 7-5: edp phy 模块属性介绍

属性	介绍
status	开启驱动，必须为 “okay”
lane_invert	选配，配置信号的极性反转，若不配置默认不反转。四个参数分别代表 lane0-lane4
lane_remap	选配，配置线序，需根据原理图配置，若不配置则使用默认 0123 的线序。 四个参数分别代表 lane0-lane4

8 各屏 uboot-board.dts 配置

将 board.dts 中对应的节点配置直接拷贝过去即可。

📖 说明

唯一需要注意的是：uboot-board.dts 中关于 GPIO 的配置，不能使用 “<&pio PI 2 GPIO_ACTIVE_HIGH>” 的配置方式，要使用 “<&pio PI 2 1 0 3 1> 的配置方式”

9 开机 logo

本章节介绍开机驱动阶段的 logo 显示配置方法

9.1 route

9.1.1 uboot-board.dts 配置

9.1.1.1 单显

```
&sunxi_drm {
    route {
        disp0_lvds0 {
            logo,uboot = "bootlogo.bmp";
            status = "okay";
        };
    };
};
```

表 9-1: bootlogo route 属性介绍

属性	介绍
dispX_YYY	必须配置，配置想要的连接关系，disp0_lvds0 表示 bootlogo 通过 de0 渲染并通过 lvds0 显示输出
status	开启驱动，必须为 “okay”
logo,uboot	选配，配置需要显示的 bootlogo 图片名，若不配置默认使用 “bootlogo.bmp”，双显时配置不同名字可实现 logo 异显

9.1.1.2 双显

增加一路 route 配置即可实现 bootlogo 双显。如以下例子配置 de0 渲染 bootlogo.bmp 图片通过 lvds0 显示，de1 渲染 bootlogo.backup.bmp 并通过 edp 显示，完成 bootlogo 双屏异显。

```
&sunxi_drm {
    route {
        disp0_lvds0 {
            logo_uboot = "bootlogo.bmp";
            status = "okay";
        };
        disp1_edp {
            logo_uboot = "bootlogo_backup.bmp";
            status = "okay";
        };
    };
};
```



10 单双显

本章节重点介绍单双显配置中的重点关注部分。

10.1 单双显图层配置

单显方案中只需要 DE0 处理图层数据，但双显方案中还需要 DE1 来处理副显的图层数据，因此在单双显中需要指定 DE 模块的硬件通道分配，以满足不同显示应用的需求。其中 DE 硬件通道分配的关键配置为 de 节点下的 **chn_cfg_mode**。不同的配置对应显示场景如下：

表 10-1: chn_cfg_mode 对应显示场景

平台	chn_cfg_mode 对应分配情况
A523/A527/T527	0: 单显 1: 常规双显 3: 双显，且主显分辨率大于 2K
A733	0: 单显 2: 常规双显

```
&de {
  chn_cfg_mode = <2>;
  status = "okay";
};
```

11 调试方法

11.1 dsi

11.2 edp

11.2.1 AUX 通信失败

串口中看到“AUX”、“EDID”、“DPCD”相关的读取失败都是由于AUX通信失败引起。AUX通信失败一般是由于屏幕没有正常工作或者排线线序不对，导致没有应答，应根据[屏幕问题硬件checklist](#)检查屏幕相关电路贴片、供电、排线等硬件情况

11.3 uboot 通用调试方法

11.3.1 确认驱动加载情况

开机后串口长按s键，可以停在uboot控制台，输入以下命令可以确认drm相关驱动加载情况

```
=> dm tree
Class  Probed Driver  Name
-----
root  [+] root_drive root_driver
simple_bus [+] generic_si |-- soc@29000000
video  [+] sunxi_disp | |-- sunxi-drm
vidconsole [+] vidconsole | | `-- sunxi-drm.vidconsole0
misc  [+] de | |-- de@5000000
misc  [+] tcon_top | |-- vo0@5500000
misc  [ ] tcon_top | |-- vo1@5510000
misc  [+] tcon | |-- tcon0@5501000
phy  [+] sunxi_dsi_ | |-- phy@5507000
display [+] sunxi_drm_ | |-- lvds0@0001000
misc  [ ] tcon | |-- tcon3@5730000
display [ ] sunxi_drm_ | `-- hdmi@5520000
backlight [+] pwm_backli |-- backlight0
panel  [+] sunxi_lvds `-- lvds_panel@0
```

11.4 kernel 通用调试方法

11.4.1 确认驱动加载情况

- 确认 drm 组件是否正确注册

```
/# ls /sys/class/drm/  
card0      card0-HDMI-A-1  card0-Writeback-1  
card0-DP-1  card0-LVDS-1   version
```

确认对应的显卡驱动，显示接口驱动是否正确注册

- 确认背光是否正确注册

```
/# ls /sys/class/backlight/  
backlight0  edp_backlight
```

12 常见问题

12.1 disp 与 drm 屏参转化

由于 disp 的 $lcd_hbp = hbp + hsync$, $lcd_vbp = vbp + vsync$; 故转换关系如下:

- $clock-frequency = lcd_ht * lcd_vt * 60$;
- $hback-porch = lcd_hbp - lcd_hspw$;
- $hactive = lcd_x$;
- $hfront-porch = lcd_ht - lcd_x - lcd_hbp$;
- $hsync-len = lcd_hspw$;
- $vback-porch = lcd_vbp - lcd_vspw$;
- $vactive = lcd_y$;
- $vfront-porch = lcd_vt - lcd_y - lcd_vbp$;
- $vsync-len = lcd_vspw$;

13 附录

13.1 屏幕问题硬件 checklist

蓝色项为选填，其余为必填		
屏幕硬件checklist		备注
方案信息		
客户公司名		
方案平台		
安卓版本		
SDK版本		
问题简述		如：使用推荐dts配置，屏幕点不亮
屏幕信息（规格书定义）		
屏幕厂商（选填）		
屏幕尺寸（选填）		
edp协议版本（选填）		
lane数量		
色深		
分辨率		如：1920x1080
有无HPD pin		
支持最大刷新率		如规格书没写，可不填
板卡硬件信息		
座上lane使用情况（贴片情况）		如：板卡端子支持4lane，但只贴了AUX_N、AUX_P、LANEO_N、LANEO_P的电容
HPD上下拉电阻贴片情况		如：上拉10K到1.8V，无下拉
VCC-LCD电压（点屏失败时）		单位：V
HPD电压（点屏失败时）		单位：V
背光电压（点屏失败时）		单位：V
VCC-LCD从机器启动到点屏过程波形（选填）		贴波形图
背光供电从机器启动到点屏过程波形（选填）		贴波形图
点屏失败时PWM波形（选填）		贴波形图

图 13-1: 硬件排查 checklist

排线线序信息		测试方法： 1. 断电 2. 通过排线连接板卡端子和屏幕 3. 使用万用表蜂鸣档，测试板卡端子的pin和对应屏端pin的导通情况
VCC-LCD导通		
HPD导通		
VCC-LED（背光供电）导通		
AUX_N导通		
AUX_P导通		
LANE0_N导通		
LANE0_P导通		
LANE1_N导通		
LANE1_P导通		
LANE2_N导通		
LANE2_P导通		
LANE3_N导通		
LANE3_P导通		

图 13-2: 硬件排查 checklist




著作权声明

版权所有 ©2025 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明

、、**全志科技**、（不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。