



Linux eDP_out_DRM 开发指南

版本号: 1.2

发布日期: 2025.5.29

版本历史

版本号	日期	制/修订人	内容描述
1.0	2024.11.27	AWA1442	初始版本。
1.1	2025.5.12	AWA1442	支持 T736 系列。
1.2	2025.5.29	AWA1442	完善 typec-dp 相关框图。

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 模块介绍	2
2.1 模块功能介绍	2
2.2 相关术语介绍	2
2.3 平台硬件特性	3
2.4 硬件连接关系	4
2.5 源码结构介绍	6
3 模块配置说明	7
3.1 menuconfig 配置	7
3.2 dts 配置介绍	11
3.2.1 TCON	11
3.2.2 VIDEO_OUT	11
3.2.3 eDP 控制器	12
3.2.4 外部 PHY 配置	13
3.2.4.1 外部 PHY 线序映射配置	13
3.2.4.2 外部 PHY 极性配置	15
3.2.4.3 配置说明	16
3.2.5 屏幕与背光配置	17
3.2.6 typec pd controller	20
3.2.6.1 husb311	20
3.2.7 typec swicher	22
3.2.8 sunxi-phy-switcher	22
3.2.9 ps8743	23
3.3 board.dts 配置	24
3.3.1 配置为 edp 模式	24
3.3.2 配置为 dp 模式	24
3.3.3 配置为 typec-dp 模式	24
3.4 uboot-board.dts 配置	25
3.4.1 配置 edp 输出模式	25
3.4.2 配置 dp 输出模式	25
3.4.3 配置为 typec-dp 模式	25
3.4.4 配置 uboot 启动显示	25
4 参考配置	26

4.1	edp 输出	26
4.1.1	board.dts	26
4.1.2	uboot-board.dts	28
4.2	dp 输出	31
4.2.1	board.dts	31
4.2.2	uboot-board.dts	31
4.3	typec-dp 输出	32
4.3.1	board.dts	32
5	常见调试方法	36
5.1	手动发起 AUX 通信（用于判断屏幕是否工作正常）	36
5.2	开启展频	36
5.3	查看当前热插拔状态	36
5.4	强制设置热插拔状态	36
5.5	确认当前使用的分辨率信息	37
5.6	手动解析 edid 的推荐分辨率	37
5.7	获取当前配置的 edp 参数	38
5.8	获取屏幕的特性支持情况	39
5.9	获取 EDID	39
5.10	获取 DPCD	40
5.11	开启更多 edp 驱动打印信息	40
5.12	使用内置 colorbar 显示	40
5.13	确认驱动注册状态	41
5.14	驱动适配完成后点屏测试	42
5.14.1	通过 drm_client 测试	42
5.14.2	modetest 测试	42
5.15	获取当前 edp 相关时钟配置	44
5.16	获取当前 phy 相关时钟配置	44
5.17	获取当前 edp 相关供电配置	44
6	FAQ	45
6.1	排查思路	45
6.1.1	硬件排查思路	45
6.1.2	软件排查思路	47
6.2	如何完整适配一个新的屏幕	47
6.3	屏相关问题	47
6.3.1	开机过程提示屏幕没有注册	47
6.3.2	背光亮，但是无法显示	48
6.3.3	显示花屏	48
6.4	启动闪屏 & 黑屏问题	49
6.4.1	uboot 启动发现 edp 报错且屏幕无法显示	49
6.4.2	uboot 启动未发现 edp 报错但无 bootlogo 显示	49
6.5	背光问题	49

6.5.1	背光不亮	49
6.5.2	显示时背光闪烁	50
6.5.3	安卓导航栏无法调节背光亮度	50
6.6	AUX 问题	50
6.6.1	AUX 通信异常或不稳定	50
6.6.2	aux wirte 时出现 NACK	51
6.7	training 问题	51
6.7.1	training 失败	51
6.7.2	training 中 lane 成功但 ALIGN 失败	51
6.8	EDID 问题	51
6.8.1	EDID 读取失败	51
6.9	兼容性问题	52
6.9.1	同个 SDK 能点亮一部分屏但是特定几款屏点不亮	52



插 图

图 2-1	edp 方案硬件连接关系	5
图 3-1	edp 驱动 menuconfig	8
图 3-2	edp 控制器 menuconfig	8
图 3-3	edp 屏幕 menuconfig	9
图 3-4	edp PHY menuconfig	9
图 3-5	edp typec switch menuconfig	10
图 3-6	edp typec pd controller menuconfig	10
图 3-7	edp typec altmode menuconfig	11
图 3-8	edp serdes combophy0 的 remap 关系	14
图 3-9	typec serdes combophy0 的 remap 关系	15
图 3-10	serdes combophy0 的极性关系	16
图 5-1	手动发起 AUX 通信	36
图 5-2	获取当前 edp 配置参数	38
图 5-3	获取屏幕支持特性	39
图 5-4	读取 EDID	40
图 5-5	读取 EDID	40
图 5-6	colorbar 样式	41
图 6-1	硬件排查 checklist	46

表 格

表 1-1	适用产品列表	1
表 2-1	术语介绍	2
表 2-2	硬件功能支持列表	3
表 2-3	edp/dp/typec-dp 硬件连接关系	4
表 2-4	edp 通路硬件介绍	5
表 3-1	edp 控制器节点属性介绍	12
表 3-2	serdes 外部 PHY 节点属性介绍	17
表 3-3	背光属性介绍	18
表 3-4	edp 屏驱动属性介绍	19
表 3-5	hub311 驱动属性介绍	21
表 3-6	sunxi-phy-switcher 驱动属性介绍	22
表 3-7	hub311 驱动属性介绍	23

ALLWINNER®

1 前言

1.1 文档简介

介绍全志平台基于 DRM 显示框架的 eDP 驱动的使用和调试方法。介绍对应 eDP 屏、Display-Port、TypeC-DP 应用场景的配置方法。

1.2 目标读者

edp 模块驱动及应用层的开发和使用人员。

1.3 适用范围

表 1-1: 适用产品列表

方案平台	内核版本
A733	Linux-6.6
T736	Linux-5.15
A537	Linux-6.6
T527_AIOT	Linux-5.15 / Linux-5.10

2 模块介绍

2.1 模块功能介绍

eDP (Embedded DisplayPort) 协议是一种数字显示技术领域的标准协议，是针对 DP (DisplayPort) 协议在嵌入式应用方向的拓展，因此 eDP 协议完全兼容 DP 协议。

eDP 接口是遵循 eDP 协议的一种标准显示接口，诞生的目的来取代比较旧的 LVDS、VGA 和 DVI 接口，以取得更好的显示效果（尤其是在全高清 FHD 或超过 FHD 解析度上的面板上）。

本文关于 edp 模块的介绍都是基于 disp2 框架下的 edp2 驱动，下文所提到的 edp 模块或 edp 驱动均指 disp2 模块下的 edp2 模块和 edp2 驱动。

由于 edp 协议兼容 dp 协议，所以全志平台的 edp 模块有时会被当作 dp 使用，通过 dp 线缆在 dp 显示器上输出内容。

2.2 相关术语介绍

表 2-1: 术语介绍

术语	说明
Sunxi	指 Allwinner 的一系列 SOC 硬件平台。
eDP	embedded displayport, VESA 定义的一种接口形式
SST	Single Stream Transport, DP/eDP 的一种数据流传输模式，大部分设备传输都是用这种模式
MST	Multi-Strame Transport, DP/eDP 的一种数据流传输模式
ASSR	Alternate Scrambler Seed Reset, 乱序传输，用于 eDP 的内容保护
HDCP	High-bandwidth Digital Content Protection, 常用于 DisplayPort 接口传输的内容保护
SSC	Spread Spectrum Clock 扩频时钟，一种用于消除频段干扰的方式
DPCD	DisplayPort Configuration Data, 本质是一片寄存器，用于设置和读取 sink 设备的特性

术语	说明
EDID	Extended Display Identification Data (VESA)，用于指示显示设备的硬件信息，I2C 地址为 0x50
TU	Transfer Unit，传输单元，main link 上传输都是按照一个个传输单元进行传输
bpc	Bits Per Component，每个像素组件/颜色占的 bit 数
bpp	Bits Per Pixel，每个像素占的 bit 数

2.3 平台硬件特性

表 2-2: 硬件功能支持列表

特性	A523/A527/T527	A733/T736	A537
驱动名	inno edp 1.3	trilinear edp 1.4	inno edp 1.3
支持协议标准	DP1.2a 和 eDP1.3	DP1.4 和 eDP1.4b	DP1.2a 和 eDP1.3
需要外部 PHY	x	CADENCE PHY	x
支持最大输出分辨率	2.5K@60Hz / 4K@30Hz	4K@60Hz	2.5K@60Hz / 4K@30Hz
支持最大 lane 数	4	4	4
支持单 lane 最大速率	2.7Gbps	5.4Gbps	2.7Gbps
理论支持最大 pixel_clk	360M	720M	360M
支持 HPD	✓	✓	✓
支持音频	I2S/SPDIF	I2S/SPDIF	I2S/SPDIF
支持 HDCP 协议	x	HDCP1.4 / HDCP2.3	x
支持 ASSR	✓	✓	✓
支持 Enhance Frame	✓	✓	✓
支持多像素模式	x	✓	✓
摆幅 (Swing) 可调	✓	✓	✓
预加重 (Pre-emphasis) 可调	✓	✓	✓

特性	A523/A527/T527	A733/T736	A537
lane 重映射 (lane remap)	x	✓	x
lane 极性取反 (lane invert)	x	✓	x
IP 支持 typec 正反插	x	✓	x
中心展频	✓	x	✓
向下展频	✓	✓	✓
PSR(屏幕自刷新)	x	x	x
fast training	x	x	x
SST 模式	✓	✓	✓
MST 模式	x	x	x
HPD pin 可否浮空	x	&check	x

2.4 硬件连接关系

表 2-3: edp/dp/typec-dp 硬件连接关系

组件 / 平台	A523/A527/T527	A733/T736	A537/A333
所属 Video_Out	vo1	vo1	vo1
Tcon 连接	tv1	tv1	tv0
外部 PHY 连接	/	serdes: combophy0	/
typec switcher	ps8743	sunxi-phy-switcher	ps8743
typec pd 控制器	husb311	husb311	husb311

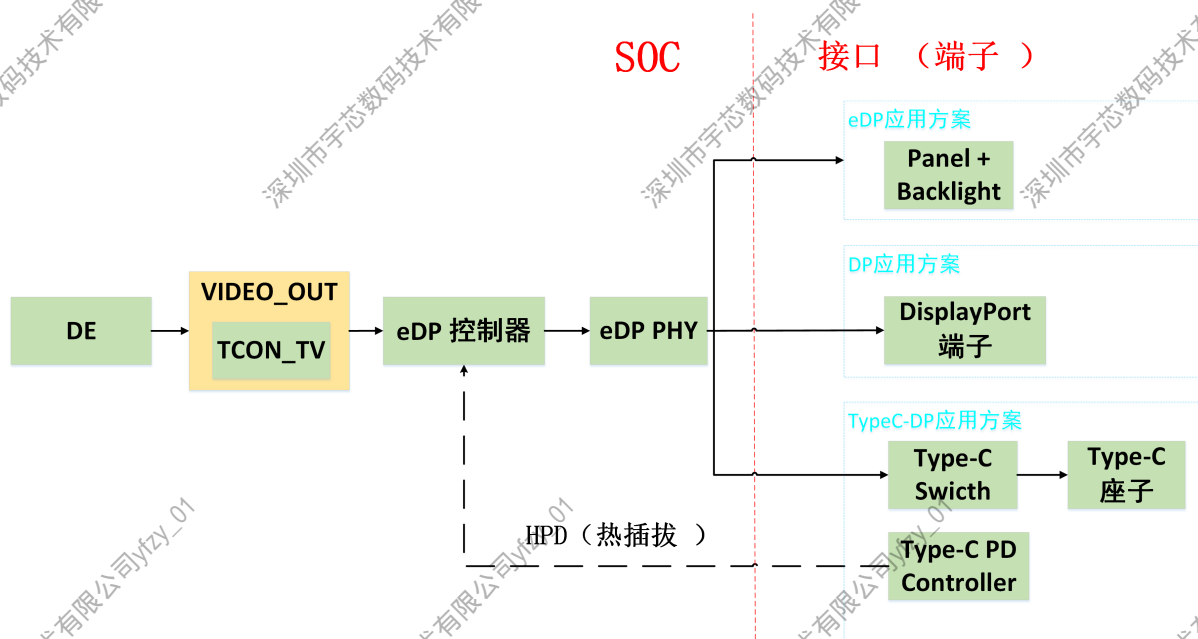


图 2-1: edp 方案硬件连接关系

表 2-4: edp 通路硬件介绍

硬件模块	介绍
VIDEO OUT	TCON 和 eDP 控制器的顶层设计，控制 TCON 和 eDP 控制器的资源，如内部供电和 gating 等。
TCON	将 DE 送来的图像信号转换为对应的时序信号
eDP 控制器	将 TCON 送来的时序信号按照 eDP 协议要求完成数据格式打包
eDP PHY	将 eDP 控制器打包完成的数据转化为对应的电平信号，送给 RX 端 部分平台的 eDP 控制器 IP 已经集成了 PHY, 就无需外部 PHY 进行配合
TypeC PD Controller	过 Typec 线上的 CC1 和 CC2 信号完成与 RX 端的 PD 通信，以此完成互相握手和 Alternate Mode 的切换
TypeC Switcher	配合 PD Controller 完成热插拔事件的上报 并将 lane 上的信号 map 到 typec 线缆中，通过 typec 线完成 DP 信号的传输

2.5 源码结构介绍

```
bsp/drivers/drm/  
├── sunxi_drm_edp.c  
├── sunxi_device  
│   ├── sunxi_edp.c  
│   ├── sunxi_edp.h  
│   └── sunxi_edp_hdcp.c  
├── hardware  
│   └── lowlevel_edp/  
│       ├── inno_edp13/  
│       │   ├── inno_edp13.c  
│       │   └── inno_edp13.h  
│       └── trilinear_dp14/  
│           ├── trilinear_dp14.c  
│           └── trilinear_dp14.h  
├── panel  
└── edp_general_panel.c  
  
bsp/drivers/phy/  
└── sunxi-cadence-combophy.c  
  
bsp/drivers/usb/typec/  
├── mux  
├── ps8743.c  
├── sunxi-phy-switcher.c  
├── tcpm  
└── tcpci_husb311.c
```

sunxi_drm_edp.c: edp 驱动代码，完成资源配置、中断处理、DRM 组件注册、设置分辨率、热插拔事件上报等基础功能。

sunxi_edp.c: edp 驱动的 core 层，主要封装常用的 api，对上对接驱动层，对下承接 lowlevel 层。

lowlevel_edp: 控制器及内部 phy 底层硬件层，主要涉及对具体功能寄存器的操作。

panels: 屏驱动相关代码，定义了通用的 edp 屏驱动，屏驱动中主要完成屏及背光的上下电、屏分辨率的上报等操作。

phy: 外部 PHY 的相关驱动代码。

typec: typec-dp 功能依赖的相关驱动代码，其中 mux 目录的为 switcher 驱动代码，tcpm 目录为 PD controller 驱动代码。

3 模块配置说明

3.1 menuconfig 配置

```
Allwinner BSP --->
/* eDP控制器及屏驱动相关 */
Device Drivers --->
  DRM Drivers --->
    <*> DRM Support for Allwinner SoCs
    [*] Support Sunxi Display_Engine(DE)
    [*] Support Timing Controller(TCON)
    [*] Support Timing Controller(TCON_TV)
    [*] Support TCON TOP
    [*] Support eDP Output --->
      *- Support HDCP For DisplayPort
      [] Support inno edp 1.3 controller
      [*] Support trilinear edp 1.4 controller
    sunxi drm panels select --->
      <*> eDP general drm panel

/* eDP 外部PHY驱动相关 */
PHY Drivers --->
  <*> Allwinner CADENCE COMBO PHY Driver

/* TypeC-DP 相关 */
USB Drivers --->
  USB Type-C Support --->
    <*> Allwinner USB Type-C support
    <*> Hynetek HUSB311 Type-C chip driver
  USB Type-C Multiplexer/DeMultiplexer Switch support --->
    <> Parade PS8743 Type-C cross switch driver
    <*> Allwinner Mux/Switcher based on external PHY

/* TypeC-DP 相关 */
Device Drivers --->
  [*] USB support --->
    <*> USB Type-C Support --->
      USB Type-C Alternate Mode drivers --->
        <*> DisplayPort Alternate Mode driver
```

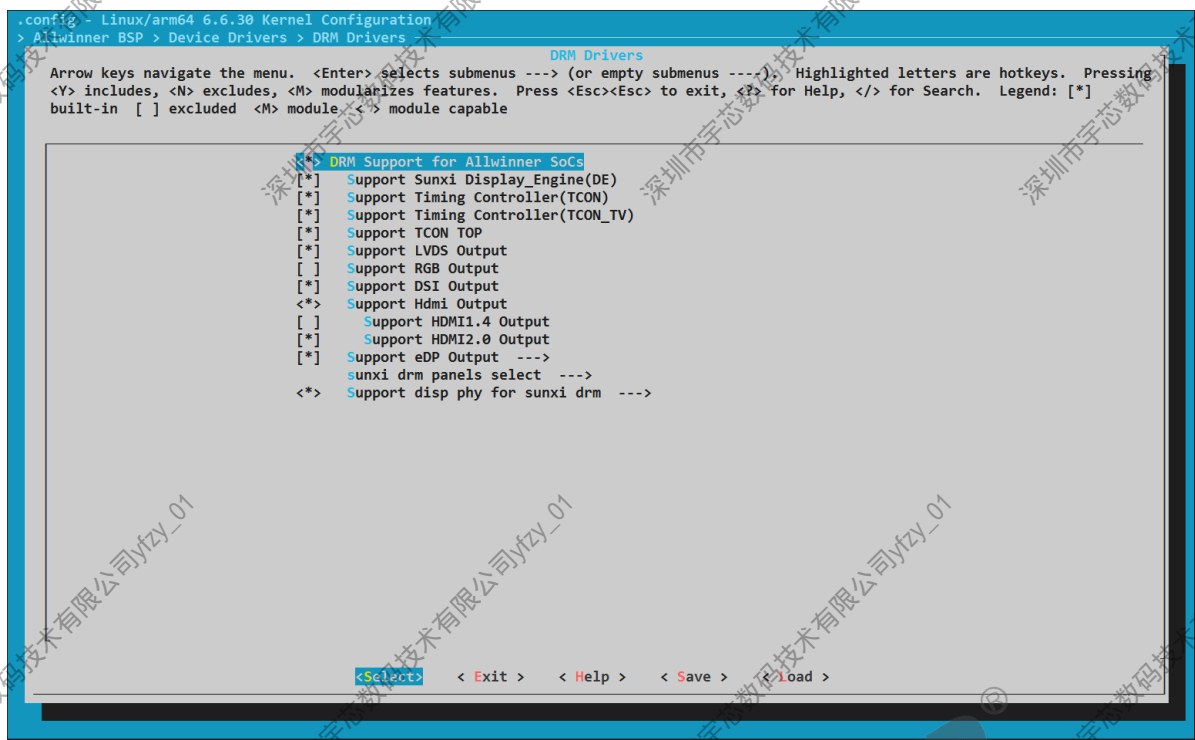


图 3-1: edp 驱动 menuconfig

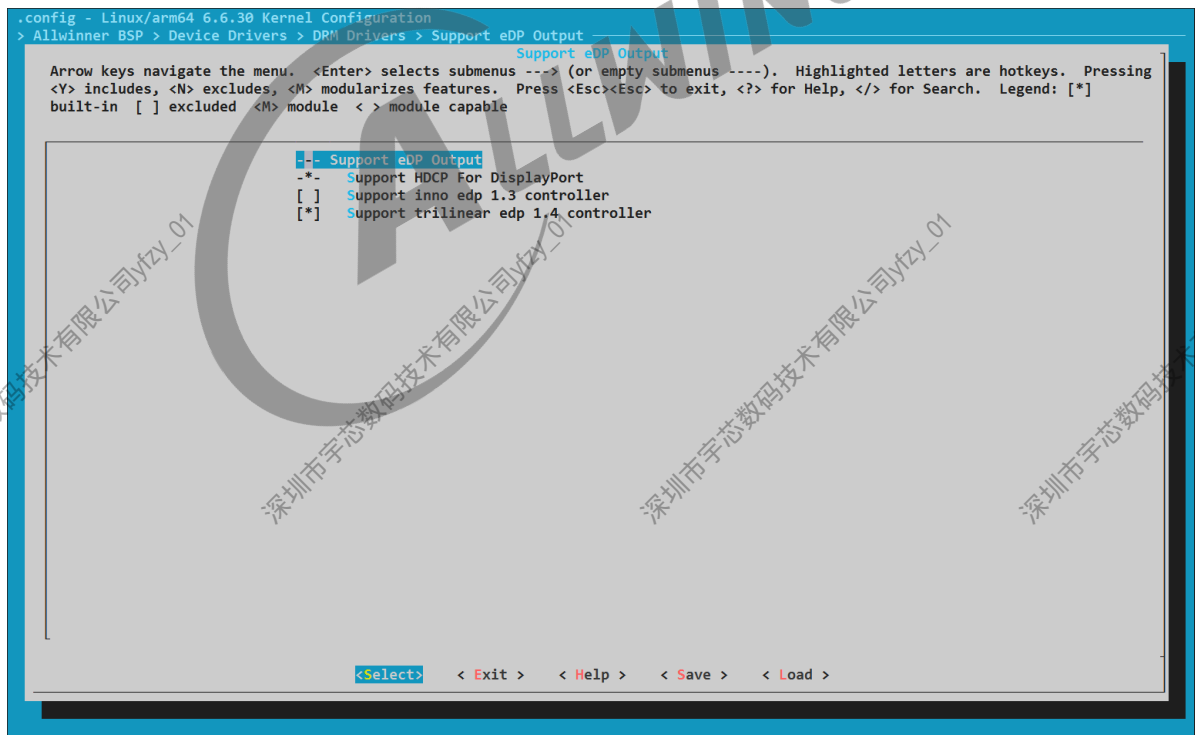


图 3-2: edp 控制器 menuconfig

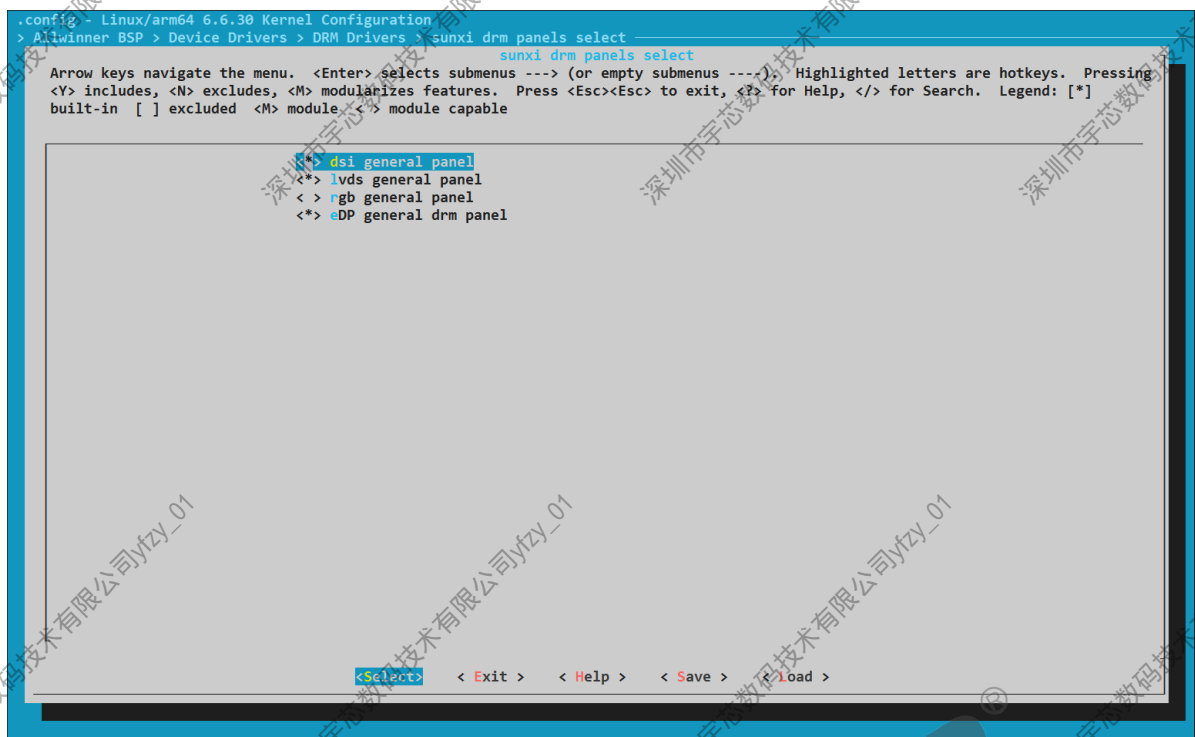


图 3-3: edp 屏幕 menuconfig

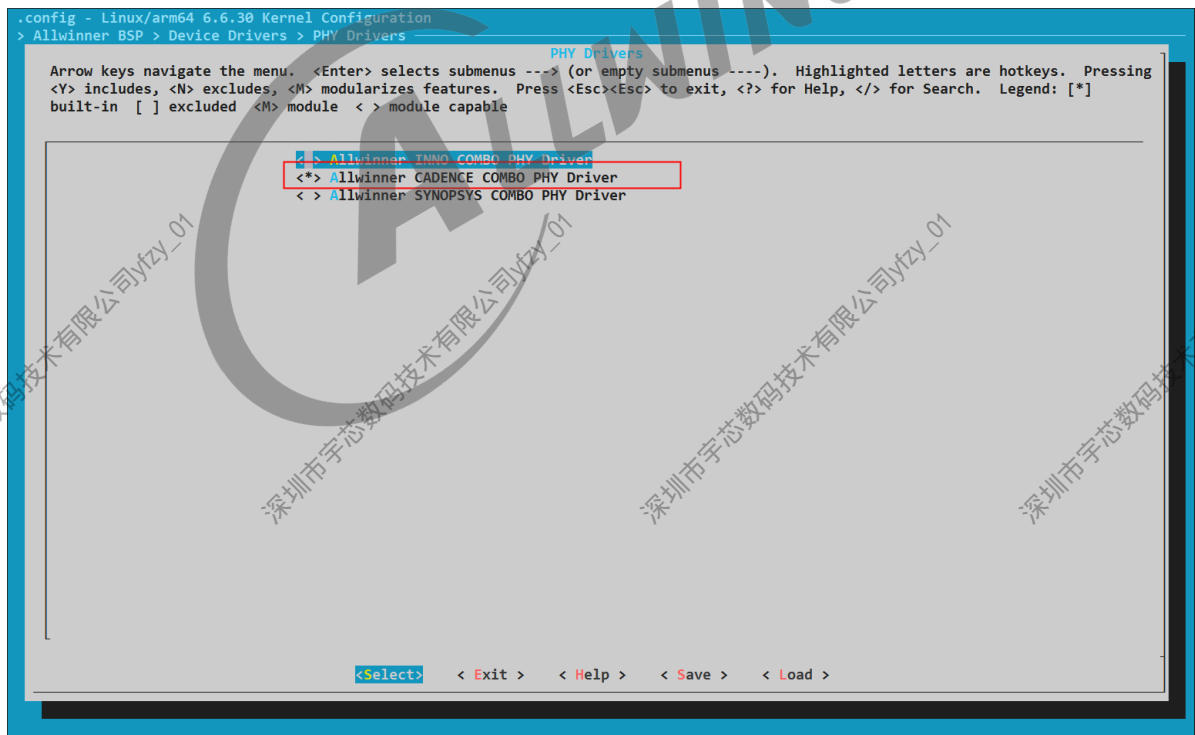


图 3-4: edp PHY menuconfig

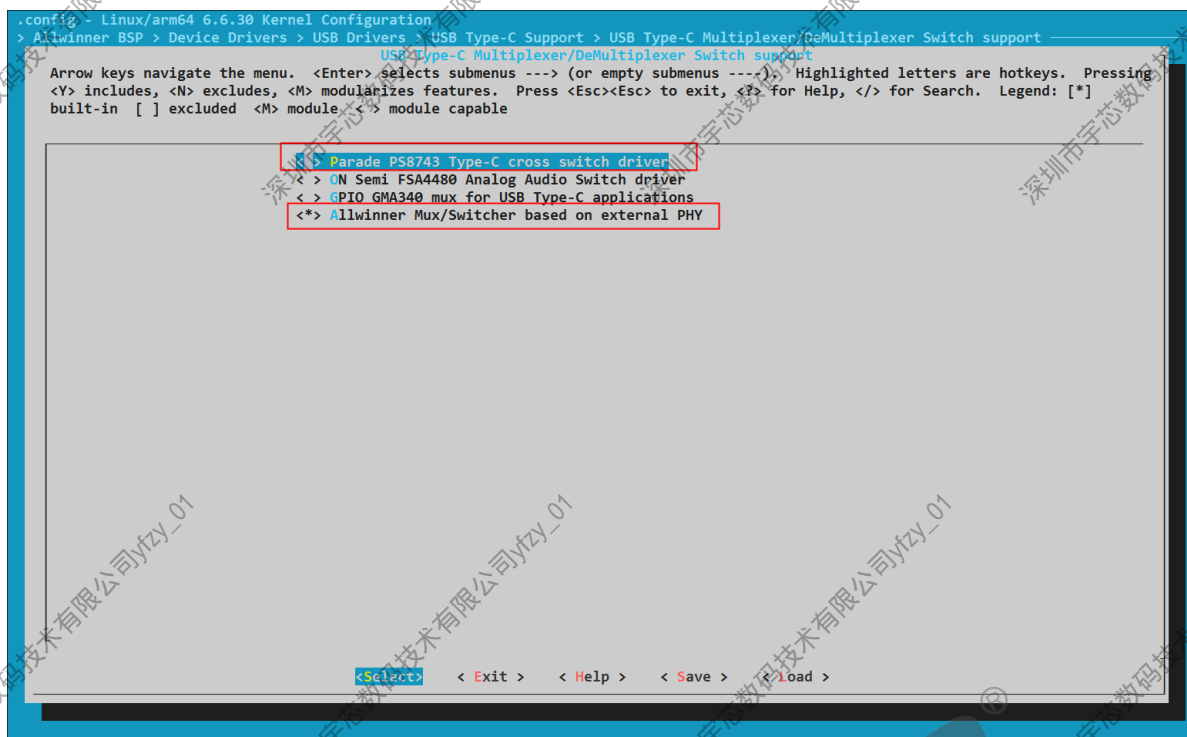


图 3-5: edp typec switch menuconfig

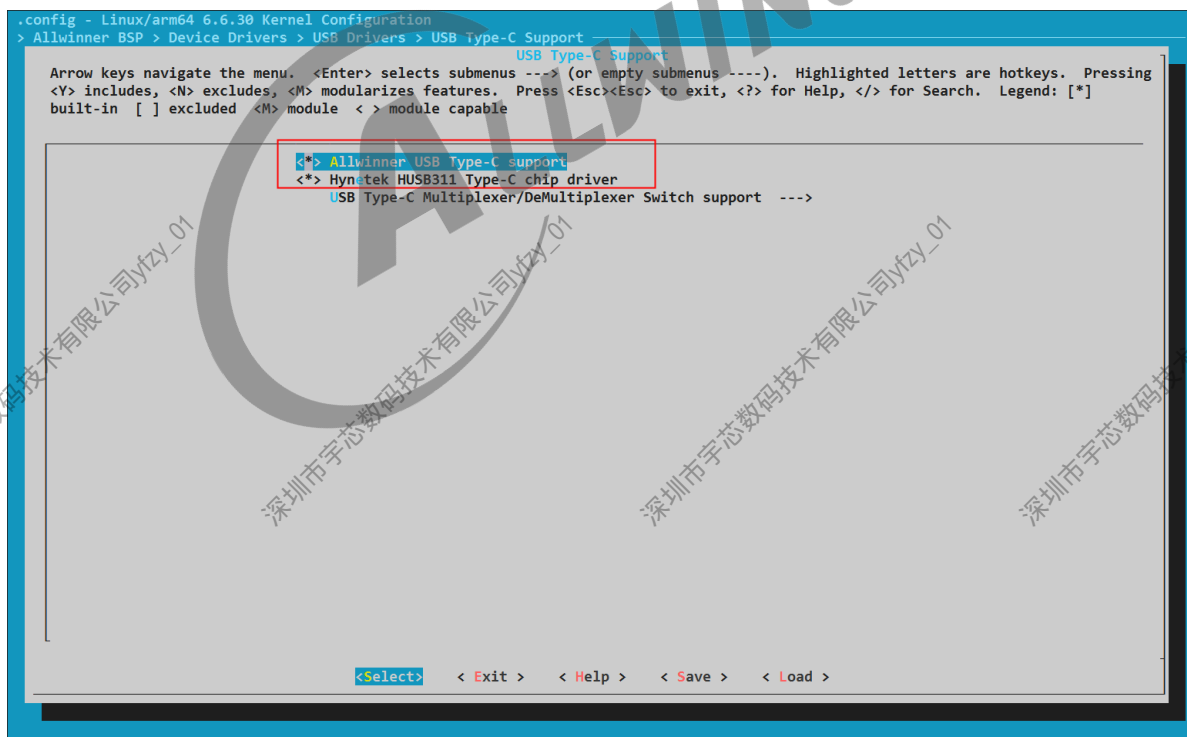


图 3-6: edp typec pd controller menuconfig

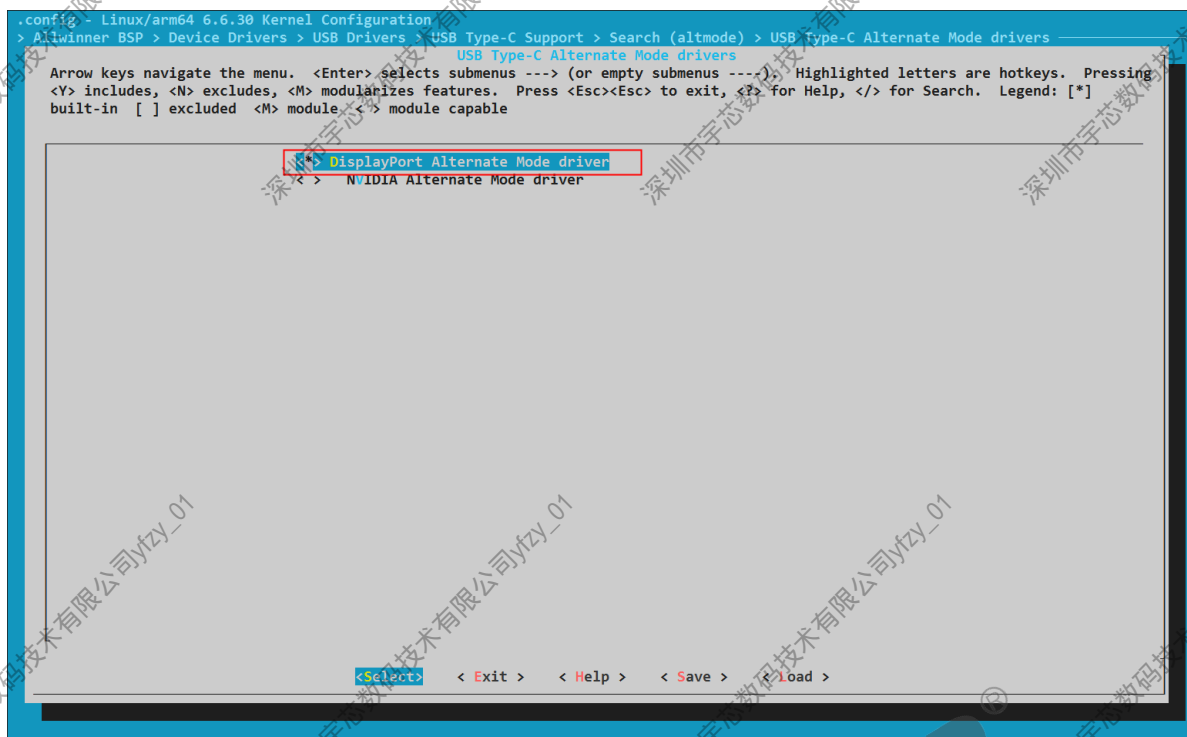


图 3-7: edp typec altmode menuconfig

3.2 dts 配置介绍

需根据实际硬件的连接关系，完成整条 edp 通路的配置，edp 通路关系可以参考[硬件连接关系](#)章节。

3.2.1 TCON

使能 TCON 节点，以确保 TCON 驱动完成加载。

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

3.2.2 VIDEO_OUT

使能 VIDEO OUT 节点，以确保 VIDEO OUT 驱动完成加载。

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

3.2.3 eDP 控制器

使能 eDP 控制器节点，并完成 edp 控制器节点的 remote-endpoint 控制，以确保 edp 驱动完成加载。

```
&drm_edp {
    compatible = "allwinner,drm-edp";
    status = "okay";
    edp_colordepth = <8>; /* 6/8/10/12/16 */
    edp_color_fmt = <0>; /* 0:RGB 1:YUV444 2:YUV422 */
    ssc_en = <0>;
    ssc_mode = <0>;

    vcc-edp-supply = <&reg_cldo5>;
    vdd-edp-supply = <&reg_dcdc2>;

    ports {
        edp_out: port@1 {
            edp_panel_out: endpoint@0 {
                reg = <0>;
                remote-endpoint = <&edp_panel_in>;
            };
        };
    };

    /* dp与typec-dp应用特有属性 */
    fps_limit_60 = <0>;
    blacklist_modes = "4096x2160@60",
        "4096x2160@59",
        "4096x2160@50",
        "4096x2160@49",
        "4096x2160@30",
        "4096x2160@29",
        "3840x2160@60",
        "3840x2160@59",
        "3840x2160@50",
        "3840x2160@49";

    /* typec-dp应用特有属性 */
    extcon = <&serdes>;
};
```

表 3-1: edp 控制器节点属性介绍

属性	介绍
compatible	edp 输出，配置为：“allwinner,drm-edp” dp 或 typec-dp 输出，配置为：“allwinner,drm-dp”
status	开启驱动，必须为 “okay”
edp_colordepth	edp 方案：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 dp 或 typec-dp 方案，配置为 8

属性	介绍
edp_color_fmt	edp 方案：根据屏幕规格书配置屏幕的色彩空间，0:RGB 1:YUV444 2:YUV422 dp 或 typec-dp 方案，配置为 0
edp_ssc_en	可选，展频功能开关，1 为使能展频，0 为关闭展频
edp_ssc_mode	可选，展频模式，1 为向下展频，0 为中心展频
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	edp 方案：必须配置，连接关系配置，
ports	remote-endpoint 需引用 edp panel 下的 in 子节点
	dp 或 typec-dp 方案：无需添加该节点
fps_limit_60	edp 方案：无需添加该节点 dp 或 typec-dp 方案：配置为 1 后屏蔽 60hz 以上分辨率， 避免分辨率超过 DE 性能出现花屏
blacklist_modes	edp 方案：无需添加该节点 dp 或 typec-dp 方案：配置后屏蔽对应分辨率， 避免部分分辨率超过 DE 性能范围出现花屏
extcon	edp 或 dp 方案：无需添加该节点 typec-dp 方案：引用外部 phy 的节点，用于 typec 方案的完成热插拔事件的通知

3.2.4 外部 PHY 配置

如芯片平台的 eDP 需要使用外部 PHY，需使能 eDP 外部 PHY 节点，完成 PHY 关于 lane mapping 和 lane invert 的配置，以确保 PHY 驱动完成加载并配置正确的信号连接关系。如果芯片平台的 eDP 模块不需要外部 PHY，可以忽略此步骤。

3.2.4.1 外部 PHY 线序映射配置

edp 数据通路最终需要根据实际原理图连接关系来配置线序映射关系，最终能实现控制器到端子的线序能对应上，即**控制器的 LANE0 最终能连接到端子的 LANE0，控制器 LANE1 连接到端子的**

LANE1, 以此类推。

- edp 或 dp 端子线序匹配

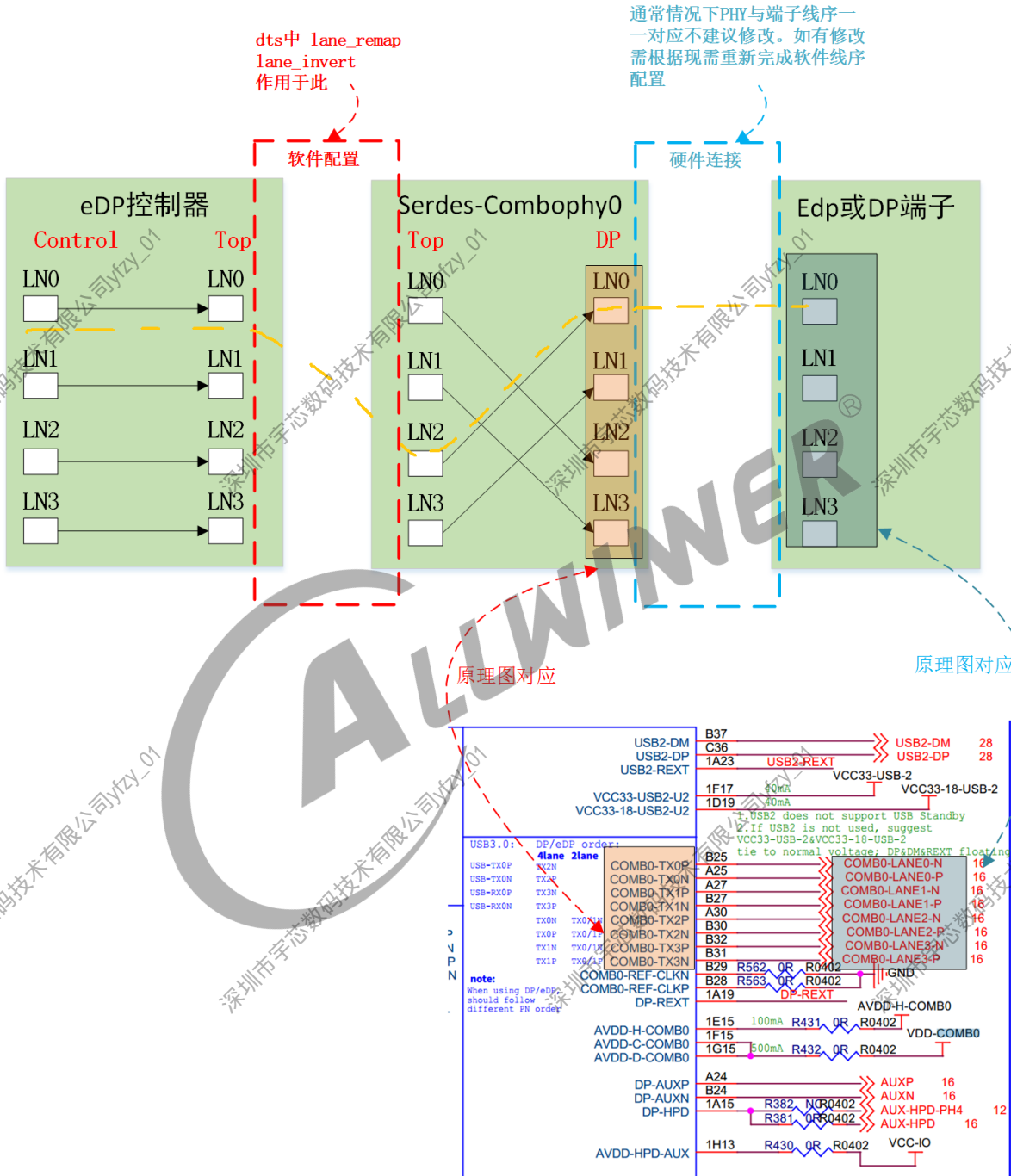


图 3-8: edp serdes combophy0 的 remap 关系

如按照上述连接关系，如果硬件连接中 COMBOPHY 的 LANE-0123（即 Serdes-Combophy0 的右端四个 LN）与端子的 LANE-0123（即示意图最右侧端子的四个 LN）是一一对应连接的，此时**如果**要让最左侧的 eDP 控制器的 LN-0123 连接到端子的 LN-0123，需要软件配置 eDP 控制器 Top

层的 LN0 连接到 Serdes Top 层的 LN2，eDP 控制器 Top 层的 LN1 连接到 Serdes Top 层的 LN3，eDP 控制器 Top 层的 LN2 连接到 Serdes Top 层的 LN0，eDP 控制器 Top 层的 LN3 连接到 Serdes Top 层的 LN1，即 dts 的 lane_remap 节点配置 <2 3 0 1>，数据通路如黄色虚线所示。

- typec-dp 端子线序匹配

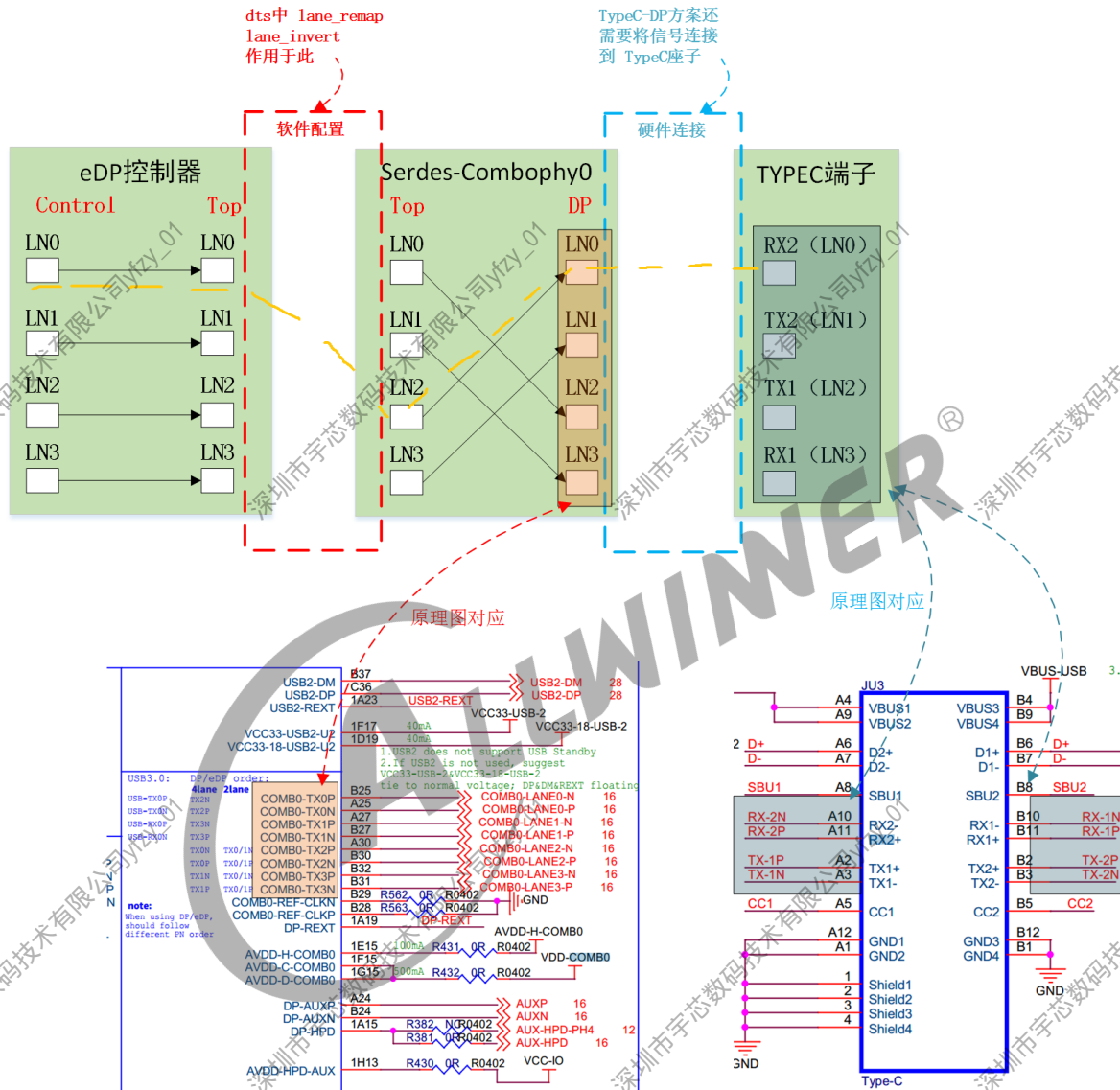


图 3-9: typec serdes combophy0 的 remap 关系

软件的线序配置与 edp 端子类似，需要确保控制器的 LN-0123 最终能一一对应到 TYPEC 端子上的 LN-0123。

3.2.4.2 外部 PHY 极性配置

edp 数据通路最终需要根据实际原理图连接关系来配置信号极性关系，最终能保证 SOC 上 edp 模块出来的差分信号的 P 和 N 最终能与屏幕端 P 和 N 能一一对应。如果实际原理图的 lane 极性与屏

端的 lane 极性相反，需手动在节点中配置取反，即配置 lane_invert 为 1。

确认两边的P和N是否一一对应，如两边相反，需要手动配置dts中对应lane_invert为 1。如本案例中4个lane的P和N对应关系都是反的，需要配置lane_invert = <1 1 1 1>;

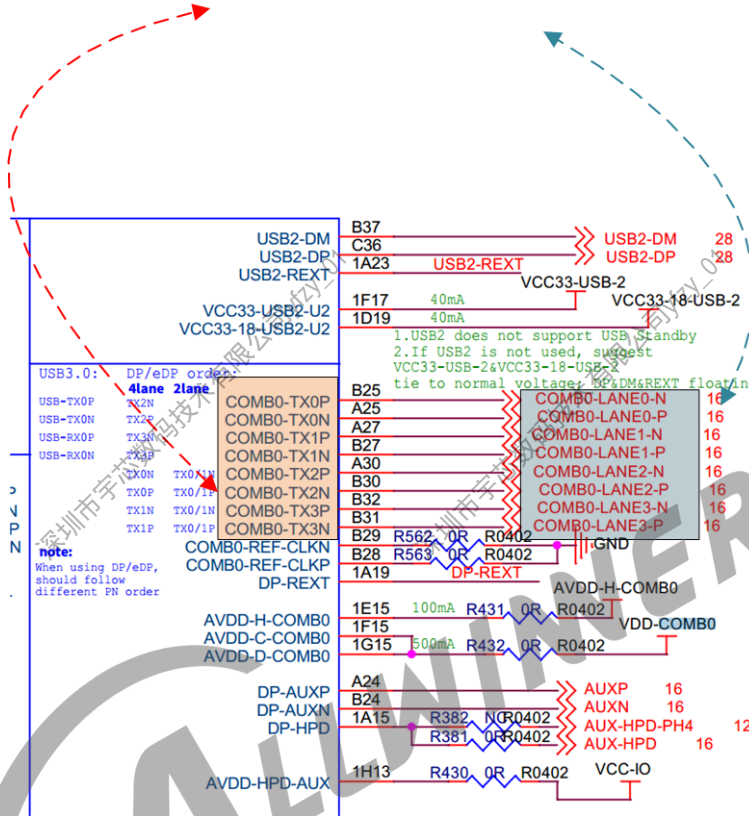


图 3-10: serdes combophy0 的极性关系

3.2.4.3 配置说明

```
&serdes {
    status = "okay";
    combophy0: combo-phy0@6c01000{
        lane_invert = <0 0 1 1>;
        lane_remap = <0 1 2 3>;
    };
};
```

配置 lane_remap 时需确保软件配置后 edp 控制器的 lane0-lane3 与实际芯片输出信号的 lane0-lane3 是一一对应的。

表 3-2: serdes 外部 PHY 节点属性介绍

属性	介绍
status	开启驱动，必须为“okay”
lane_invert	lane 的极性反转配置。 共四个参数分别代表 lane0-lane3 的极性配置，0 为不反转，1 为极性反转
lane_remap	lane 的重映射配置，共四个参数分别代表 lane0-lane3 的重映射配置。 配置的数字表示 edp phy 的 lane 映射到 edp 控制器的哪个 lane
typec_remap	typec-dp 方案时才需配置，默认使用 <2 3 0 1>，不允许擅自修改

针对 A733/T736 平台，edp/dp/typec-dp 方案的 serdes combophy0 的信号连接要完全按照全志平台提供的硬件方案标案原理图设计，不应随意更改。与方案标案原理图对应的 lane 配置可以完全按照以下完成配置：

```
/* edp/dp方案: */
lane_remap = <2 3 0 1>;
lane_invert = <1 1 1 1>;

/* typec-dp方案:*/
lane_remap = <0 1 2 3>;
lane_invert = <1 1 1 1>;
typec_remap = <2 3 0 1>;
```

3.2.5 屏幕与背光配置

根据实际原理图情况配置屏节点和背光节点，并完成屏节点的 remote-endpoint 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 = <200>;
enable-gpios = <&pio PJ 26 GPIO_ACTIVE_HIGH>;
/* power-supply = <&reg_backlight_12v>; */
pwms = <&pwm1 9 5000000 0>;
};

```

表 3-3: 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，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 0

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，具体可以参考上文的例子。

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。

```

edp_panel: edp_panel {
    compatible = "edp-general-panel";
    status = "disabled";
    power0-supply = <&reg_dc1sw1>;
    enable0-gpio = <&pio PJ 23 GPIO_ACTIVE_HIGH>;
    power-delay-ms = <50>;

    backlight = <&backlight0>;

    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>;
            };
        };
    };
};

```

表 3-4: 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，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

属性	介绍
ports	必须配置，连接关系配置， remote-endpoint 需引用 drm_edp 节点下的 out 子节点

3.2.6 typec pd controller

不同芯片平台可能搭配的 pd controller 会有差异，应根据按照[硬件连接关系](#)章节选择配置对应的节点。

3.2.6.1 husb311

```

husb311: husb311@4e {
    compatible = "hynetek,husb311";
    reg = <0x4e>;
    interrupt-parent = <&r_pio>;
    husb311,intr_gpio = <&r_pio PM 5 GPIO_ACTIVE_LOW>;
    // interrupts = <PM 5 GPIO_ACTIVE_LOW>;
    // interrupt-names = "wakeup";
    vbus-supply = <&reg_usb0_vbus>;
    // det_usb_supply = <&usb_power_supply>;
    // aw,port-reset-quirk,
    wakeup-source;
    status = "okay";

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

        port@0 {
            reg = <0>;
            usbc0_role_sw: endpoint@0 {
                remote-endpoint = <&usbc0_role_switch>;
            };
        };
    };

    usb_con: connector {
        compatible = "usb-c-connector";
        label = "USB-C";
        data-role = "dual";
        power-role = "dual";
        try-power-role = "sink";
        op-sink-microwatt = <1000000>;
        typec-power-opmode = "default";
        //pd-disable;
        slow-charger-loop;
        sink-pdos = < PDO_FIXED(5000, 500, (0x1<<29)|(0x1<<25))
                    PDO_VAR(5000, 5000, 2000)
                    /* PDO_VAR(9000, 9000, 2000)

```


属性	介绍
其余属性	switcher 节点下的对应 mux 和 sw 必须配置，原封不动 copy 即可

3.2.7 typec swicher

3.2.8 sunxi-phy-switcher

```

phy_switcher: phy_switcher@10 {
    compatible = "allwinner,sunxi-phy-switcher";
    reg = <0x0 0x10 0x0 0x0>;
    phys = <&combo0_dp>, <&combo0_usb>;
    phy-names = "dp-phy", "usb-phy";
    extcon = <&usb_power_supply>;
    usb_psy = <&usb_power_supply>;
    dp_configure_delay = <30>;

    status = "okay";

    mode-switch;
    orientation-switch;
    svid = <0xff01>;
    port {
        #address-cells = <1>;
        #size-cells = <0>;
        usbdp_orientation_switch: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&usbc0_orien_sw>;
        };
        usbdp_dp_altmode_mux: endpoint@1 {
            reg = <1>;
            remote-endpoint = <&dp_altmode_mux>;
        };
    };
};

```

表 3-6: sunxi-phy-switcher 驱动属性介绍

属性	介绍
usb_psy	必须配置，用于 vbus 的限流控制， 与 pd controller 的 vbus-supply 配置保持一致
extcon	必须配置，无实际作用，仅为保证驱动加载顺序，与 usb_psy 引用同一节点
dp_configure_delay	可选，当出现兼容性问题时，可以调整这个延时，以提高 CC 握手的兼容性

属性	介绍
ports	必须配置，mux 和 sw 的 remote-endpoint 连接到 pd controller 节点下的对应 mux 和 sw
其余属性	必须配置，原封不动 copy 即可

3.2.9 ps8743

```
ps8743: ps8743@11 {
    compatible = "parade,ps8743";
    reg = <0x11>;
    hotplug = <&pio PH 19 GPIO_ACTIVE_HIGH>;
    status = "disabled";

    mode-switch;
    orientation-switch;
    svid = <0xff01>;
    port {
        #address-cells = <1>;
        #size-cells = <0>;
        usbdp_orientation_switch: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&usbc2_orien_sw>;
        };
        usbdp_dp_altmode_mux: endpoint@1 {
            reg = <1>;
            remote-endpoint = <&dp_altmode_mux>;
        };
    };
};
```

表 3-7: husb311 驱动属性介绍

属性	介绍
hotplug	热插拔 hpd gpio，根据实际原理图上 edp 控制器 hpd pin 连接的 gpio 配置
ports	必须配置，mux 和 sw 的 remote-endpoint 连接到 pd controller 节点下的对应 mux 和 sw
其余属性	必须配置，原封不动 copy 即可

3.3 board.dts 配置

3.3.1 配置为 edp 模式

按照[硬件连接关系](#)章节和[dts 配置介绍](#)章节完成所需模块的相关节点使能和配置：

- 完成 TCON 配置
- 完成 VIDEO_OUT 配置
- 完成 eDP 控制器配置
- 如芯片平台需要使用外部 PHY，完成外部 PHY 配置
- 完成背光配置
- 完成 eDP 屏配置

详细配置可以参考《AW_DRM 屏幕适配_调试指南》，或参考本开发指南的[参考配置](#)章节。

3.3.2 配置为 dp 模式

按照[硬件连接关系](#)章节和[dts 配置介绍](#)章节完成所需模块的相关节点使能和配置：

- 完成 TCON 配置
- 完成 VIDEO_OUT 配置
- 完成 eDP 控制器配置
- 如芯片平台需要使用外部 PHY，完成外部 PHY 配置

详细配置可以参考《AW_DRM 屏幕适配_调试指南》，或参考本开发指南的[参考配置](#)章节。

3.3.3 配置为 typec-dp 模式

按照[硬件连接关系](#)章节和[dts 配置介绍](#)章节完成所需模块的相关节点使能和配置：

- 完成 TCON 配置
- 完成 VIDEO_OUT 配置
- 完成 eDP 控制器配置
- 如芯片平台需要使用外部 PHY，完成外部 PHY 配置
- 配置 typec switcher 驱动节点，确保 typec 相关驱动能正确注册。
- 配置 typec pd controller 驱动节点，确保 typec 相关驱动能正确注册。

typec-dp 方案相比 dp 方案要多增加 typec 相关组件的配置（主要是 typec pd controller 和 typec switcher 的配置），其余配置与 dp 方案保持一致。详细配置可以参考本开发指南的[参考配置](#)章节，或《Linux_TypeC 开发指南》。

3.4 uboot-board.dts 配置

3.4.1 配置 edp 输出模式

与 board.dts 配置一致，直接将 board.dts 配置拷贝过来即可。

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”

3.4.2 配置 dp 输出模式

与 board.dts 配置一致，直接将 board.dts 配置拷贝过来即可。

3.4.3 配置为 typec-dp 模式

Typec-DP 应用目前仅在 linux 起来后使用，uboot 阶段暂时不支持 Typec-DP 输出。

3.4.4 配置 uboot 启动显示

方案如需在启动阶段通过 edp 屏或 dp 显示器显示 bootlogo 等，还需配置 uboot 阶段的显示绑定关系。如下：

```
&route_edp{
    status = "okay";
    logo,uboot = "bootlogo.bmp";
};
```

4 参考配置

4.1 edp 输出

4.1.1 board.dts

```
/{
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>;
};

edp_panel: edp_panel {
```

```
compatible = "edp-general-panel";
status = "okay";
power0-supply = <&reg_dcdc4>;
enable0-gpios = <&pio PI 2 GPIO_ACTIVE_HIGH>;

backlight = <&backlight0>;

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>;
        };
    };
};
...
};

&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";
};

&drm_edp {
    status = "okay";

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

    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>;
    };
};
};

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

&vo0 {
    status = "okay";
};

&tv1 {
    status = "okay";
};

&serdes {
    status = "okay";
    combophy0: combo-phy0@6c01000{
        lane_invert = <1 1 1 1>;
        lane_remap = <2 3 0 1>;
        typec_remap = <2 3 0 1>;
    };
};

```

4.1.2 uboot-board.dts

```

/*
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 1 0 3 1>;
pwms = <&pwm0 4 25000 0>;
};

edp_panel: edp_panel {
    compatible = "edp-general-panel";
    status = "okay";
    power0-supply = <&reg_dcdc4>;
    enable0-gpios = <&pio PI 2 1 0 3 1>;

    backlight = <&backlight0>;

    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>;
            };
        };
    };
};

...
};

&pwm4_pin_active {
    allwinner,pins = "PD22";
    allwinner,function = "pwm0_4";
    allwinner,muxsel = <0x2>;
    allwinner,drive = <3>;
    allwinner,pull = <2>;
};

&pwm4_pin_sleep {
    allwinner,pins = "PD22";
    allwinner,function = "io_disabled";
    allwinner,muxsel = <0xf>;
};
```

```
allwinner,drive = <3>;
allwinner,pull = <2>;
};

&pwm4 {
    pinctrl-names = "active","sleep";
    pinctrl-0 = <&pwm4_pin_active>;
    pinctrl-1 = <&pwm4_pin_sleep>;
    status = "okay";
};

&drm_edp {
    status = "okay";

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

    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>;
            };
        };
    };
};

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

&vo0 {
    status = "okay";
};

&tv1 {
    status = "okay";
};

&serdes {
    status = "okay";
    combophy0: combo-phy0@6c01000{
        lane_invert = <1 1 1 1>;
        lane_remap = <2 3 0 1>;
        typec_remap = <2 3 0 1>;
    };
};

&route_edp {
    status = "okay";
    logo,uboot = "bootlogo.bmp";
};
```

};

4.2 dp 输出

4.2.1 board.dts

```
&drm_edp {
    compatible = "allwinner,drm-dp";
    status = "okay";

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

    vcc-edp-supply = <&reg_bldo3>;
    vdd-edp-supply = <&reg_dcdc2>;
};

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

&vo0 {
    status = "okay";
};

&tv1 {
    status = "okay";
};

&serdes {
    status = "okay";
    combophy0: combo-phy0@6c01000 {
        lane_invert = <1 1 1 1>;
        lane_remap = <2 3 0 1>;
        typec_remap = <2 3 0 1>;
    };
};
```

4.2.2 uboot-board.dts

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

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

```
vcc-edp-supply = <&reg_bldo3>;
vdd-edp-supply = <&reg_dc2c2>;
};

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

&vo0 {
    status = "okay";
};

&tv1 {
    status = "okay";
};

&serdes {
    status = "okay";
    combophy0: combo-phy0@6c01000{
        lane_invert = <1 1 1 1>;
        lane_remap = <2 3 0 1>;
        typec_remap = <2 3 0 1>;
    };
};

&route_edp {
    status = "okay";
    logo,uboot = "bootlogo.bmp";
};
```

4.3 typec-dp 输出

4.3.1 board.dts

```
{
    husb311: husb311@4e {
        compatible = "hynetek,husb311";
        reg = <0x4e>;
        interrupt-parent = <&r_pio>;
        husb311,intr_gpio = <&r_pio PM 5 GPIO_ACTIVE_LOW>;
        // interrupts = <PM 5 GPIO_ACTIVE_LOW>;
        // interrupt-names = "wakeup";
        vbus-supply = <&reg_usb0_vbus>;
        // det_usb_supply = <&usb_power_supply>;
        // aw,port-reset-quirk;
        wakeup-source;
        status = "okay";

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

            port@0 {
                reg = <0>;
```

```
usb0_role_sw: endpoint@0 {
    remote-endpoint = <&usb0_role_switch>;
};
};
};

usb_con: connector {
    compatible = "usb-c-connector";
    label = "USB-C";
    data-role = "dual";
    power-role = "dual";
    try-power-role = "sink";
    op-sink-microwatt = <1000000>;
    typec-power-opmode = "default";
    //pd-disable;
    slow-charger-loop;
    sink-pdqs = < PDO_FIXED(5000, 500, (0x1<<29)|(0x1<<25))
        PDO_VAR(5000, 5000, 2000)
        /* PDO_VAR(9000, 9000, 2000)
        PDO_VAR(12000, 12000, 3000)
        PDO_VAR(15000, 15000, 3000)
        PDO_VAR(20000, 20000, 2250) */
        >;
    source-pdos = < PDO_FIXED(5000, 500, (0x1<<29)|(0x1<<25))
        PDO_VAR(5000, 5000, 1200) >;
    /* fixme */
    sink-vdos = <0xd1002e99 0x00 0x03110000>;
    sink-vdos-v1 = <0xd1002e99 0x00 0x03110000>;

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

        altmode@0 {
            reg = <0>;
            svid = <0xff01>;
            vdo = <0xffffffff>;
        };
    };

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

        port@0 {
            reg = <0>;
            usb0_orien_sw: endpoint@0 {
                remote-endpoint = <&usbdp_orientation_switch>;
            };
        };
        port@1 {
            reg = <1>;
            dp_altmode_mux: endpoint@1 {
                remote-endpoint = <&usbdp_dp_altmode_mux>;
            };
        };
    };
};
};
```

```
phy_switcher: phy_switcher@10 {
    compatible = "allwinner,sunxi-phy-switcher";
    reg = <0x0 0x10 0x0 0x0>;
    phys = <&combo0_dp>, <&combo0_usb>;
    phy-names = "dp-phy", "usb-phy";
    extcon = <&usb_power_supply>;
    usb_psy = <&usb_power_supply>;
    dp_configure_delay = <30>;

    status = "okay";

    mode-switch;
    orientation-switch;
    svid = <0xff01>;
    port {
        #address-cells = <1>;
        #size-cells = <0>;
        usbdp_orientation_switch: endpoint@0 {
            reg = <0>;
            remote-endpoint = <&usb0_orien_sw>;
        };
        usbdp_dp_altmode_mux: endpoint@1 {
            reg = <1>;
            remote-endpoint = <&dp_altmode_mux>;
        };
    };
};

&drm_edp {
    compatible = "allwinner,drm-dp";
    status = "okay";

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

    extcon = <&serdes>;
    vcc-edp-supply = <&reg_bldo3>;
    vdd-edp-supply = <&reg_dc2c2>;
};

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

&vo0 {
    status = "okay";
};

&tv1 {
    status = "okay";
};

&serdes {
    status = "okay";
    combophy0: combo-phy0@6c01000{
```

```
lane_invert = <1 1 1 1>;  
lane_remap = <2 3 0 1>;  
typec_remap = <2 3 0 1>;  
};  
};
```



5 常见调试方法

5.1 手动发起 AUX 通信（用于判断屏幕是否工作正常）

```
echo 0x00,0x10 > /sys/class/edp/edp/attr/aux_read  
cat /sys/class/edp/edp/attr/aux_read
```

读出来非全 0 即屏幕工作正常，edp AUX 功能正常。其余情况均为屏幕工作异常

```
t527-demo:/ # echo 0x00,0x10 > /sys/class/edp0/edp0/attr/aux_read  
t527-demo:/ # cat /sys/class/edp0/edp0/attr/aux_read  
[AUX_READ] Addr:0x0000 Lenth:17  
0x0000: 0x11 0x0a 0x84 0x41 0x00 0x00 0x01 0xc0  
0x0008: 0x02 0x03 0x00 0x00 0x00 0x0b 0x00 0x00  
0x0010: 0x00
```

图 5-1: 手动发起 AUX 通信

5.2 开启展频

在 board.dts 和 uboot-board.dts 的 drm_edp 节点中，添加 **edp_ssc_en = <1>**。

5.3 查看当前热插拔状态

```
cat /sys/class/edp/edp/attr/hotplug
```

5.4 强制设置热插拔状态

```
echo 0x11 > /sys/class/edp/edp/attr/hpd_mask
```

- 0x0：无强制状态。

- 0x10：模拟强制拔出，会执行驱动的热插拔处理函数，且会上报 extcon 的拔出事件，常用于作为 DP 输出时的热插拔压测。
- 0x11：模拟强制插入，会执行驱动的热插拔处理函数，且会上报 extcon 的插入事件，常用于作为 DP 输出时的热插拔压测。
- 0x110：模拟强制拔出，会执行驱动的热插拔处理函数，但是不上报 extcon 的拔出事件。
- 0x111：模拟强制插入，会执行驱动的热插拔处理函数，但是不上报 extcon 的插入事件。
- 0x1010：模拟强制关闭热插拔功能，不管插入还是拔出事件都不会被处理，不会上报 extcon 拔出事件。

5.5 确认当前使用的分辨率信息

```
cat /sys/class/edp/edp/attr/timings_now
```

5.6 手动解析 edid 的推荐分辨率

通过 `cat /sys/class/edp/edp/attr/edid` 拿到 edid 的信息。

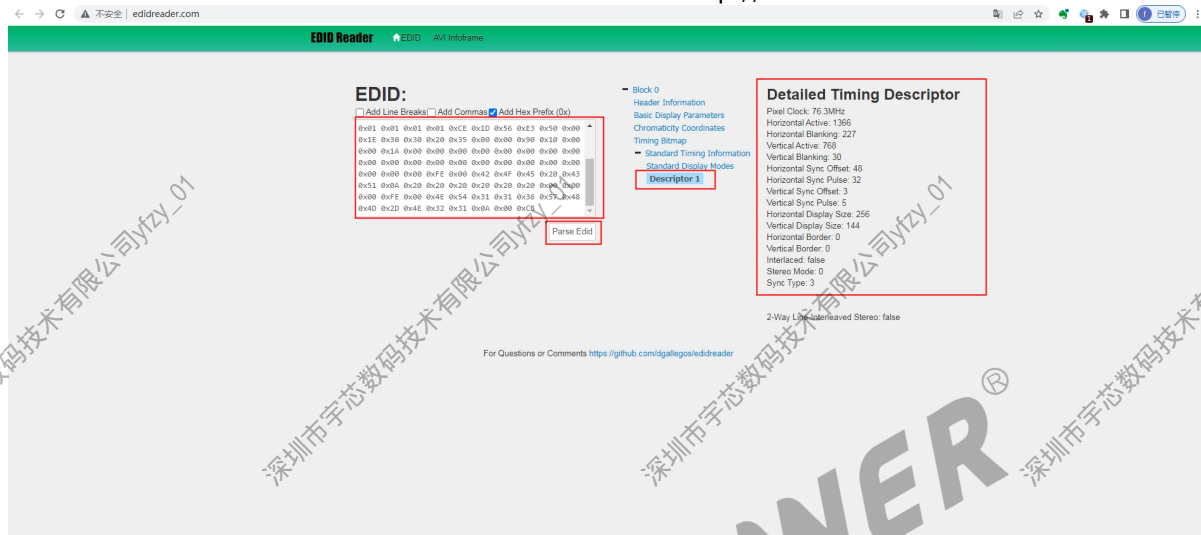
```
0x00: 0x00 0xff 0xff 0xff 0xff 0xff 0x00
0x08: 0x09 0xe5 0x31 0x07 0x00 0x00 0x00 0x00
0x10: 0x08 0x1b 0x01 0x04 0x95 0x1a 0x0e 0x78
0x18: 0x02 0xe9 0x90 0x95 0x5c 0x58 0x94 0x29
0x20: 0x24 0x50 0x54 0x00 0x00 0x00 0x01 0x01
0x28: 0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x30: 0x01 0x01 0x01 0x01 0x01 0x01 0xce 0x1d
0x38: 0x56 0xe3 0x50 0x00 0x1e 0x30 0x30 0x20
0x40: 0x35 0x00 0x00 0x90 0x10 0x00 0x00 0x1a
0x48: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x50: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x58: 0x00 0x00 0x00 0x00 0x00 0xfe 0x00 0x42
0x60: 0x4f 0x45 0x20 0x43 0x51 0x0a 0x20 0x20
0x68: 0x20 0x20 0x20 0x20 0x00 0x00 0x00 0xfe
0x70: 0x00 0x4e 0x54 0x31 0x31 0x36 0x57 0x48
0x78: 0x4d 0x2d 0x4e 0x32 0x31 0x0a 0x00 0xcb
```

稍作处理，将前面的地址打印去掉，得到 edid 的 raw 数据。

```
0x00 0xff 0xff 0xff 0xff 0xff 0x00
0x09 0xe5 0x31 0x07 0x00 0x00 0x00 0x00
0x08 0x1b 0x01 0x04 0x95 0x1a 0x0e 0x78
0x02 0xe9 0x90 0x95 0x5c 0x58 0x94 0x29
0x24 0x50 0x54 0x00 0x00 0x00 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0x01 0x01
0x01 0x01 0x01 0x01 0x01 0x01 0xce 0x1d
0x56 0xe3 0x50 0x00 0x1e 0x30 0x30 0x20
```

```
0x35 0x00 0x00 0x90 0x10 0x00 0x00 0x1a
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0xfe 0x00 0x42
0x4f 0x45 0x20 0x43 0x51 0x0a 0x20 0x20
0x20 0x20 0x20 0x20 0x00 0x00 0x00 0xfe
0x00 0x4e 0x54 0x31 0x31 0x36 0x57 0x48
0x4d 0x2d 0x4e 0x32 0x31 0x0a 0x00 0xcb
```

访问在线解析网站将数据进行粘贴和解析，得到结果：<http://www.edidreader.com>。



5.7 获取当前配置的 edp 参数

```
cat /sys/class/edp/edp/attr/lane_config_now
```

```
t527-demo: / # cat /sys/class/edp0/edp0/attr/lane_config_now
lane_para_use: default
bit_rate: 2700000000
lane_cnt: 4
color depth: 8
color fmt: 0
lane0_sw: 0
lane1_sw: 0
lane2_sw: 0
lane3_sw: 0
lane0_pre: 0
lane1_pre: 0
lane2_pre: 0
lane3_pre: 0
```

图 5-2: 获取当前 edp 配置参数

5.8 获取屏幕的特性支持情况

```
cat /sys/class/edp/edp/attr/sink_info
```

```
t527-demo:/ # cat /sys/class/edp0/edp0/attr/sink_info
[Capacity Info]
dpcd_rev: 11
is_edp_device: Yes
max_bit_rate: 2700000000
max_lane_cnt: 4
tps3_support: No
fast_link_train_support: Yes
downstream_port_support: No
downstream_port_type: NULL
downstream_port_cnt: NULL
local_edid_support: Yes
assr_support: Yes
enhance_frame_support: Yes

[Edid Info]
mfg_year: 2013
mfg_week: 255
edid_ver: 1
edid_rev: 4
width_cm: 22
height_cm: 13
input_type: Digital
input_bit_depth: 8
sink_video_interface: DP/eDP
Ycc444_support: No
Ycc422_support: No
Ycc420_support: No
audio_support: No
```

图 5-3: 获取屏幕支持特性

5.9 获取 EDID

```
cat /sys/class/edp/edp/attr/edid
```

```
t527-demo:/ # cat /sys/class/edp0/edp0/attr/edid
00: 00 ff ff ff ff ff ff 00
08: 34 a9 a2 96 00 00 00 00
10: ff 17 01 04 a5 16 0d 78
18: 02 51 73 a4 58 4d a1 26
20: 15 4f 56 00 00 00 01 01
28: 01 01 01 01 01 01 01 01
30: 01 01 01 01 01 01 d1 6f
38: 00 f0 a0 40 68 60 58 20
40: c5 04 df 7d 00 00 00 1e
48: 00 00 00 00 00 00 00 00
50: 00 00 00 00 00 00 00 00
58: 00 00 00 00 00 fd 00 3b
60: 3d 64 67 1d 01 0a 20 20
68: 20 20 20 20 00 00 00 fc
70: 00 56 56 58 31 30 54 30
78: 32 35 4a 30 30 0a 00 3b
```

图 5-4: 读取 EDID

5.10 获取 DPCD

```
cat /sys/class/edp/edp/attr/dpcd
```

5.11 开启更多 edp 驱动打印信息

```
echo 4 > /sys/class/edp/edp/attr/loglevel_debug
```

```
t527-demo:/ # cat /sys/class/edp0/edp0/attr/loglevel_debug
0:NONE 1:EDP_DRV 2:EDP_CORE 4:EDP_LOW 8:EDP_EDID
t527-demo:/ # echo x > /sys/class/edp0/edp0/attr/loglevel_debug
```

图 5-5: 读取 EDID

5.12 使用内置 colorbar 显示

```
echo 1 > /sys/class/edp/edp/attr/colorbar
```

输入上述命令，屏幕正常点亮后会输出多个颜色交错色块。

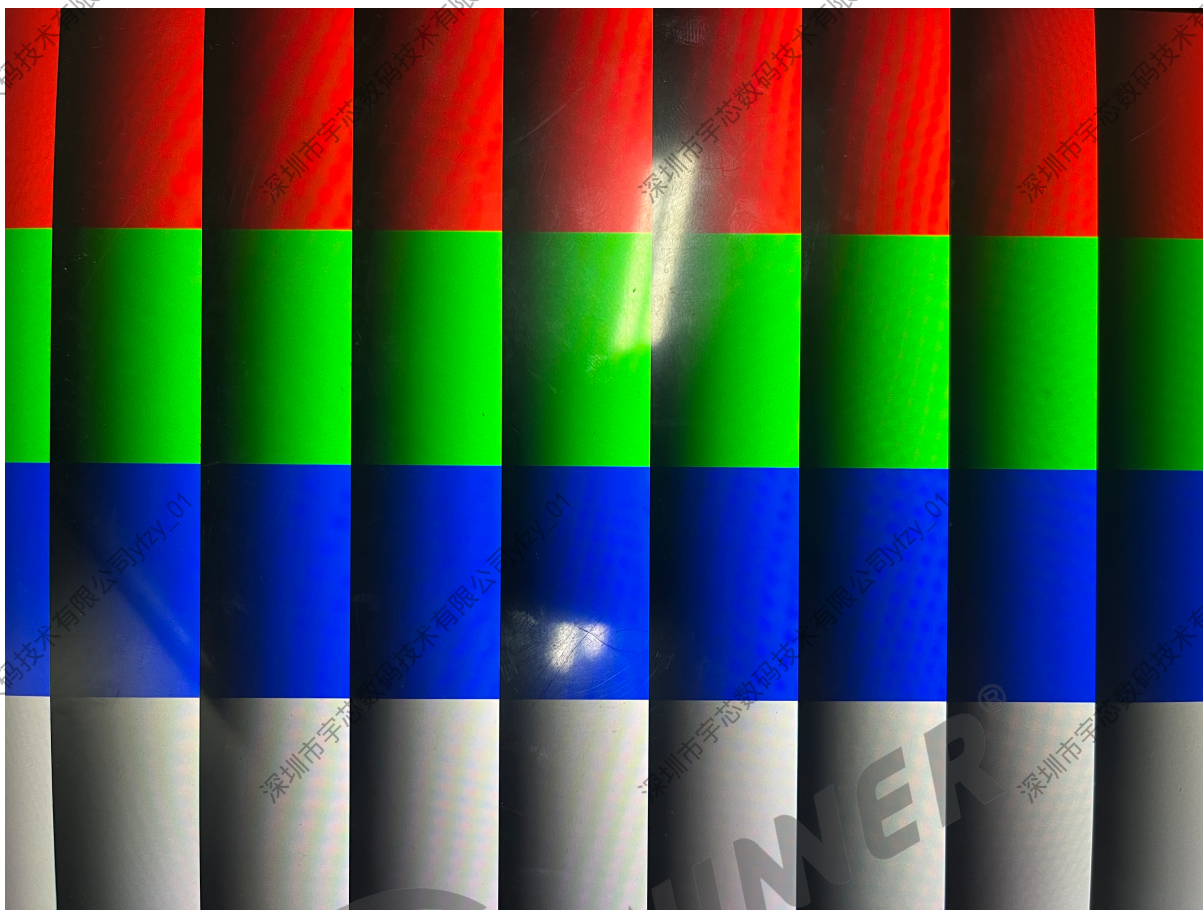


图 5-6: colorbar 样式

5.13 确认驱动注册状态

- 确认背光驱动注册状态

```
ls -l /sys/class/backlight
```

查看是否有 dts 中定义的背光的名字

- 确认 edp 驱动注册状态

```
ls -l /sys/class/drm
```

查看是否有 **card-DP-1** 或 **card-eDP-1** 的名字

5.14 驱动适配完成后点屏测试

5.14.1 通过 drm_client 测试

drm_client 点屏的原理为：sunxi-drm 驱动注册时会同时利用 drm_fbdev 相关接口注册一个 fb 设备，fb 设备注册时会注册一个 client，该 client 会在系统启动时将最识别到的第一个 connected 的设备作为主显示设备并使能其显示通路。可以利用这个特性将 edp 显示通路交由 drm_client 打开，以此验证 edp 屏的适配情况。

1. 确保方案只开启了 eDP/DP 一个显示接口（connector），避免别的显示设备抢先作为主显示设备，可以通过启动后输入：

```
ls -l /sys/class/drm
```

确认是否有其他显示接口信息，如有其他显示接口也注册了，可在 board.dts 中关闭对应节点配置（如：lvds、dsi、hdmi）status 为 disabled 暂时关闭

2. 确保 uboot 显示关闭

暂时将 uboot-board.dts 的 drm 驱动关闭。需打开 uboot-board.dts，添加以下节点：

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

3. 通过 colorbar 显示，输入以下命令：

```
echo 1 > /sys/class/edp/edp/attr/colorbar
```

4. 确认屏幕显示情况，如能正常 colorbar 色块，说明适配成功。colorbar 色块样式见[使用内置 colorbar 显示](#)章节。

5.14.2 modetest 测试

modetest 为 libdrm 开源库中自带的一个用于 KMS 功能测试的测试工具，可以指定显示设备的绑定关系和分辨率，我们可以利用此工具完成基础的点屏调试。Buildroot 方案已经直接集成该工具，安卓及其他 linux 方案需找 AW 的 FAE 工程师获取对应版本工具。

1. 通过 **adb push modetest ./** 将 modetest 推入机器
2. **chmod 777 modetest** 将 modetest 权限放开

3. 使用 `./modetest -M sunxi-drm -c` 获取 edp/dp 的相关信息，主要关注 id 号和支持的分辨率信息

```

/ # ./etc/modetest -M sunxi-drm -c
Connectors:
id      encoder status      name      size (mm)  modes  encoders
146     145     connected  DP-1      600x340  38     145
modes:
name refresh (Hz) hdisp hss hse htot vdisp vss vse vtot)
3840x2160 60 3840 3888 3952 4000 2160 2163 2168 2222 533250 flags: phsync, nvsync; type: preferred, driver
3840x2160 30 3840 3888 3920 4000 2160 2163 2168 2191 262750 flags: phsync, nvsync; type: driver
1920x2160 60 1920 1968 2000 2080 2160 2163 2173 2222 277250 flags: phsync, nvsync; type: driver
2560x1440 60 2560 2608 2640 2720 1440 1443 1448 1481 241500 flags: phsync, pvsync; type: driver
1920x1080 60 1920 2008 2052 2200 1080 1084 1089 1125 148500 flags: phsync, pvsync; type: driver
1920x1080 60 1920 2008 2052 2200 1080 1084 1089 1125 148500 flags: phsync, pvsync; type: driver
1920x1080 60 1920 2008 2052 2200 1080 1084 1089 1125 148352 flags: phsync, pvsync; type: driver
1920x1080 50 1920 2448 2492 2640 1080 1084 1089 1125 148500 flags: phsync, pvsync; type: driver
1680x1050 60 1680 1784 1960 2240 1050 1053 1059 1089 146250 flags: nhsync, pvsync; type: driver
1280x1024 75 1280 1296 1440 1688 1024 1025 1028 1066 135000 flags: phsync, pvsync; type: driver
1280x1024 60 1280 1328 1440 1688 1024 1025 1028 1066 108000 flags: phsync, pvsync; type: driver
1440x900 60 1440 1520 1672 1904 900 903 909 934 106500 flags: nhsync, pvsync; type: driver
1280x960 60 1280 1376 1488 1800 960 961 964 1000 108000 flags: phsync, pvsync; type: driver
1280x720 60 1280 1390 1430 1650 720 725 730 750 74250 flags: phsync, pvsync; type: driver
1280x720 60 1280 1390 1430 1650 720 725 730 750 74250 flags: phsync, pvsync; type: driver
1280x720 60 1280 1390 1430 1650 720 725 730 750 74176 flags: phsync, pvsync; type: driver
1280x720 50 1280 1720 1760 1980 720 725 730 750 74250 flags: phsync, pvsync; type: driver
1024x768 75 1024 1040 1136 1312 768 769 772 800 78750 flags: phsync, pvsync; type: driver
1024x768 70 1024 1048 1184 1328 768 771 777 806 75000 flags: nhsync, nvsync; type: driver
1024x768 60 1024 1048 1184 1344 768 771 777 806 65000 flags: nhsync, nvsync; type: driver
832x624 75 832 864 928 1152 624 625 628 667 57284 flags: nhsync, nvsync; type: driver
800x600 75 800 816 896 1056 600 601 604 625 49500 flags: phsync, pvsync; type: driver
800x600 72 800 856 976 1040 600 637 643 666 50000 flags: phsync, pvsync; type: driver
800x600 60 800 840 968 1056 600 601 605 628 40000 flags: phsync, pvsync; type: driver
800x600 56 800 824 896 1024 600 601 603 625 36000 flags: phsync, pvsync; type: driver
720x576 50 720 732 796 864 576 581 586 625 27000 flags: nhsync, nvsync; type: driver
720x576 50 720 732 796 864 576 581 586 625 27000 flags: nhsync, nvsync; type: driver
720x480 60 720 736 798 858 480 489 495 525 27027 flags: nhsync, nvsync; type: driver
720x480 60 720 736 798 858 480 489 495 525 27027 flags: nhsync, nvsync; type: driver
720x480 60 720 736 798 858 480 489 495 525 27000 flags: nhsync, nvsync; type: driver
720x480 60 720 736 798 858 480 489 495 525 27000 flags: nhsync, nvsync; type: driver
640x480 75 640 656 720 840 480 481 484 500 31500 flags: nhsync, nvsync; type: driver
640x480 73 640 664 704 832 480 489 492 520 31500 flags: nhsync, nvsync; type: driver

```

4. 使用 `cat /proc/sunxi-drm/status | grep DE` 获取 DE 信息，主要关注 DE id 号

```

/ #
/ # cat /proc/sunxi-drm/status | grep DE
crtc=DE-0
crtc[99]: DE-0
crtc[125]: DE-1
/ #

```

5. 指定绑定关系和分辨率，进行送显。输入以下命令：

```
./modetest -M sunxi-drm -s xxx@yyy:widthxheight-fps
```

其中，xxx为edp/dp的id号，yyy为de的id号，weight为edp/dp的分辨率宽，height为edp/dp的分辨率高，fps为刷新率，fps可以不配

6. 通过 colorbar 显示，输入以下命令：

```
echo 1 > /sys/class/edp/edp/attr/colorbar
```

7. 确认屏幕显示情况，如能正常 colorbar 色块，说明适配成功。colorbar 色块样式见[使用内置 colorbar 显示章节](#)

5.15 获取当前 edp 相关时钟配置

```
mount -t debugfs none /sys/kernel/debug/  
cat sys/kernel/debug/clk/clk_summary | grep edp
```

5.16 获取当前 phy 相关时钟配置

```
mount -t debugfs none /sys/kernel/debug/  
cat sys/kernel/debug/clk/clk_summary | grep serdes
```

5.17 获取当前 edp 相关供电配置

```
mount -t debugfs none /sys/kernel/debug/  
cat sys/kernel/debug/regulator/regulator_summary | grep edp
```

6 FAQ

6.1 排查思路

6.1.1 硬件排查思路

根据《屏幕硬件排查 checklist_v0.1》排查屏幕外围硬件情况。后续如需建立屏幕相关问题的问题单时，也需要先将实际排查的结果填写到《屏幕硬件排查 checklist_v0.1》并反馈给 AW 的 FAE 工程师。需要检查的硬件外围包括但不限于：

- 屏幕端子 lane 和 AUX 信号上的隔直电容贴片情况是否满足屏幕要求的 lane 数
- HPD pin 上下拉电阻贴片情况及 HPD pin 电平情况是否正常，HPD 波形在系统启动过程是否存在跌落情况。（在 A523/A527/T527 平台中，SOC eDP host 模块的 HPD pin 需要大于 1.8V 电压才能正常工作，否则会导致点屏时失败。因此，如果部分屏幕自身不带 HPD pin，外围电路也需要手动将 SOC eDP host 模块的 HPD pin 进行手动上拉，确保模块工作正常工作）。
- 背光及屏幕供电（VCC-LCD）的供电电压是否正常，电压波形在启动过程中是否存在跌落情况
- 背光用到的 pwm 输出波形是否正常输出方波或高电平
- 排线线序是否正确，能否满足屏幕端子与屏幕的信号导通
- Soc eDP host 模块的外部校准电阻贴片情况和阻值是否满足 AW edp 模块的参考设计要求

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		
座子上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、LANE0_N、LANE0_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
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 6-1: 硬件排查 checklist

6.1.2 软件排查思路

⚠ 注意

在进行软件排查前，请确保已经完成硬件排查无异常，避免排查方向错误导致排查周期变长。

- 检查 edp 驱动是否被正确加载，检查方法：`ls -l /sys/class/drm/`，如没有 eDP 相关的文件说明 edp 驱动没有加载成功
- 根据参考配置检查配置是否有错误、遗漏的情况。包括但不限于：
 1. lane 参数信息：`edp_lane_cnt`、`edp_lane_rate`
 2. 输出格式信息：`edp_color_fmt` 和 `edp_colordepth`
 3. 供电信息：`board.dts` 中背光供电及屏幕供电是否添加 regulator 节点，regulator 的控制 gpio 是否配置正确，添加的 regulator 句柄是否被 edp 节点引用
 4. 背光使能 gpio 和 pwm 信息：`enable-gpios`、`pwms`
- 若上述检查都无异常，通过手动发起 AUX 通信确认屏幕是否正常工作。
 1. 若手动通信结果正常但屏幕依然不亮，怀疑方向有：背光异常、分辨率错误、training 失败。可根据对应 FAQ 完成对应排查。
 2. 若手动通信失败，怀疑方向为：硬件问题，屏幕协议兼容性问题。需要再次根据硬件排查思路完成硬件检查，确认硬件没问题后，根据兼容性问题进行排查。

6.2 如何完整适配一个新的屏幕

- 如果一个方案之前已经成功通过 EDID 自动解析分辨率的方式点亮过屏幕，则 dts 无需改动即可直接适配新屏幕
- 如果一个方案之前已经通过 dts 配置分辨率的方式点亮过屏幕，则只需要根据屏幕规格书修改 `board.dts` 及 `uboot-board.dts` 中的 timings 即可点亮屏幕
- 如果需要从全新的方案开始适配屏幕，建议按照以下步骤进行：
 1. 根据参考配置配置 `uboot-board.dts`
 2. 根据参考配置配置 `board.dts`
 3. 如出现屏幕点不亮的情况，先按照排查思路排查软硬件配置
 4. 如按照排查思路依然无法解决问题，根据问题现象通过 FAQ 查找解决方法

6.3 屏相关问题

6.3.1 开机过程提示屏幕没有注册

如果开机过程提示 “edp’s panel driver maybe not registered yet!”，且通过命令 `ls /sys/class/drm` 确认没有 eDP 相关文件生成，说明屏驱动没有注册上导致 eDP 驱动缺少依赖也无法正

确加载，应检查以下方面：

1. dts 中屏相关节点是否已经填好，compatible 是否正确
2. 屏驱动的 CONFIG 是否已经打开
3. 如果是安卓固件，屏驱动是否已经添加在模块的 insmod 列表，可以打开 **android/device/sunxi/xxx/common/system/vendor_ramdisk.modules** 确认里面是否包含 **edp_general_panel.ko**。

6.3.2 背光亮，但是无法显示

确认串口中是否有以下类型的报错：

```
[EDP_ERR]: Error:edid header invalid
[EDP_ERR]: edid block0 is invalid
[EDP_ERR]: fail to read edid
[EDP_ERR]: fail to read dpcd
[EDP_WRN]: edp_aux_read wait AUX_REPLY timeout
[EDP_ERR]: retry 5 times but still fail, xxxx
```

如有上述类型报错，大概率是 edp 屏幕未正常工作引起，请先按照[硬件排查思路](#)排查屏幕外围电路及相关供电情况。

6.3.3 显示花屏

正确送显后，edp 显示花屏可能由以下几个原因造成，可根据实际情况排查确认：

- 多次输入 **cat /proc/sunxi-drm/status | grep err** 确认是否存在 err 计数增长，如果发现 err 计数持续增长说明屏幕分辨率超过当前 DE 的处理能力，如部分平台 4K@60 的应用方案在某些场景下会超过 DE 硬件的处理能力，导致花屏，应联系 AW 原厂工程师解决。

```
/ # cat /proc/sunxi-drm/status | grep err
on: 1920x1080@60&600Mhz->tcon5 irqcnt=1634102 err=1
```

- 确认点屏是否用的是 dts 中自定义的分辨率，如果是自定义的分辨率出现的花屏，可能是用户使用的屏参异常，建议删除 dts 中的 **panel-timing** 节点，使用屏幕 EDID 内部携带的分辨率信息进行点屏。

6.4 启动闪屏 & 黑屏问题

6.4.1 uboot 启动发现 edp 报错且屏幕无法显示

uboot 启动过程中发现 edp 的报错，如：

```
[EDP_ERR]: Error:edid header invalid
[EDP_ERR]: edid block0 is invalid
[EDP_ERR]: fail to read edid
[EDP_ERR]: fail to read dpcc
[EDP_WRN]: edp_aux_read wait AUX_REPLY timeout
[EDP_ERR]: retry 5 times but still fail, xxxx
```

如有上述类型报错，大概率是 edp 屏幕未正常工作引起，请先按照**硬件排查思路**排查屏幕外围电路及相关供电情况。

6.4.2 uboot 启动未发现 edp 报错但无 bootlogo 显示

目前已知几种情况会导致此类问题的出现：

- uboot-board.dts 未正确配置使用 edp 显示
uboot 未发现 edp 的报错，有可能是 uboot 未指定 edp 作为显示屏输出，不会跑到 edp 的驱动代码，自然不会出现 edp 的报错。需要排查 uboot-board.dts 的 **route_edp** 是否配置为 okay

6.5 背光问题

6.5.1 背光不亮

edp 屏的背光亮起主要依赖几个系统资源：背光供电、pwm 和 gpio。当确认 edp 开启显示的流程正确走完，但出现背光不亮的情况时，可以按照以下步骤进行排查：

1. 通过万用表量取背光的供电，确认背光供电是否开启。如果背光供电没开启，排查是否是 edp 节点中关于背光供电的引用句柄与实际原理图没对应。
2. 如果背光供电已经开启但是背光依然不亮，可能是由于背光的使能没有开启导致，通过万用表量取 backlight_en 的 gpio 电平，确认电平状态是否与屏规格书中规定的使能状态一致。排查是否 dts 中背光节点引用的 gpio 与实际原理图没对应。
3. 如果背光供电及背光使能都已经正确配置，需要确认背光的 pwm 是否正常输出，这个步骤可

通过万用表量取背光 pwm pin 电压大致确认，必要时可以接示波器确认 pwm 波形频率和占空比。

6.5.2 显示时背光闪烁

背光闪烁通常是由于背光供电不稳定造成，常见是由于背光供电电路带载能力不足造成，出现背光闪烁时可以反馈给方案硬件同事进行排查。

6.5.3 安卓导航栏无法调节背光亮度

安卓导航栏控制背光亮度需要依赖两个条件：- 背光驱动已注册，可以通过 `ls -l /sys/class/backlight/` 确认 dts 中定义的 edp 背光是否已注册商。- dts 中的背光名字是否命名为 `backlight0`，如非该名字，安卓的背光中间件无法正确识别名字，造成无法调节背光的情况。

6.6 AUX 问题

6.6.1 AUX 通信异常或不稳定

怀疑 AUX 通信有异常时，可以先手动通过 `aux_read` 节点对 sink 端的 `dpcd` 寄存器进行访问：

```
echo 0x00,0x10 > /sys/class/edp/edp/attr/aux_read; cat /sys/class/edp/edp/attr/aux_read  
echo 0x50,0x80 > /sys/class/edp/edp/attr/aux_i2c_read; cat /sys/class/edp/edp/attr/aux_i2c_read
```

通过 AUX 读取数据的结果，如果连续多次读出的数据并非全 0 且多次读的数据都一致，基本排除 AUX 异常。否则的话，结合下述现象进行进一步分析。

出现以下几种打印，都说明 AUX 通信有问题：

```
edp_aux_read fail, addr:  
  
AUX_REPLY timeout  
  
fail to read dpcd  
  
edid header invalid  
edid block0 is invalid  
fail to read edid  
  
retry 5 times but still fail, training1(clock recovery training) fail
```

根据经验，导致 AUX 通信异常的基本由以下问题造成：

- 屏幕外围硬件问题，可按照[硬件排查思路](#)进行排查。
- 屏幕协议兼容性问题，可按照[兼容性问题](#)进行排查

6.6.2 aux write 时出现 NACK

部分 dpcd 寄存器不可写，不能往这部分寄存器写值，否则就会出现 NACK。

6.7 training 问题

6.7.1 training 失败

training 失败通常伴随以下打印：

```
CR training result: lane0:FAIL lane1:FAIL lane2:FAIL lane3:FAIL
retry 5 times but still fail, training1(clock recovery training) fail

CR training result: lane0:FAIL lane1:FAIL lane2:FAIL lane3:FAIL
retry 5 times but still fail, training2(equalization training) fail
```

根据经验，training 失败通常由以下原因造成：

- 屏幕外围硬件问题，可按照**硬件排查思路**进行排查。
- 软件 lane 参数配置问题，通常伴随串口 “Try to enlarge lane count or lane rate” 打印，可按照**配置 lane 参数**进行排查。
- 信号幅度问题，可以尝试修改 “edp_training_param_type” 参数，将其配置为 1 再测试 training 是否成功。

6.7.2 training 中 lane 成功但 ALIGN 失败

这种情况通常伴随以下打印：

```
EQ training result: lane0:PASS lane1:PASS lane2:PASS lane3:PASS ALIGN:FAIL
retry 5 times but still fail, training2(equalization training) fail
```

出现此类问题说明 eDP 的 PHY 参数可能不适用当前屏幕，需要联系 AW 原厂进行分析。

6.8 EDID 问题

6.8.1 EDID 读取失败

根据经验，EDID 读取失败大概率是屏幕没有正常工作导致，可按照**硬件排查思路**进行排查。

6.9 兼容性问题

6.9.1 同个 SDK 能点亮一部分屏但是特定几款屏点不亮

⚠ 注意

怀疑兼容性问题前，请先确保已按照**硬件排查思路**和**软件排查思路**确认软硬件配置均无异常

兼容性问题有以下几种原因：

1. 市面上部分 edp 屏幕使用的 edp 协议版本为 eDP 1.1 甚至更老，这些旧版本的协议对数据包和同步头的要求与新版本的略有出入，可能会导致 AUX 通信失败。此时需要人为切换 soc edp 控制器模式，实现数据格式要求的兼容，配置方法如下：

```
&drm_edp {  
...  
force_mode = "dp";  
...  
};
```

2. 部分屏幕、排线与测试样机上的走线过长、走线阻抗不匹配，会导致 edp lane 的信号出现干扰或衰减，导致某些屏幕无法点亮，此时可以尝试提升信号幅度，配置方法如下：

```
&drm_edp {  
...  
edp_training_param_type = <1>;  
...  
};
```




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。