



# **AW Display 配置使用指南**

**版本号: 1.0  
发布日期: 2024.4.2**

## 版本历史

版本号	日期	制/修订人	内容描述
1.0	2024.4.2	AWA02075	创建初始版本



# 目 录

<b>1 前言</b>	<b>1</b>
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
1.4 相关术语介绍	1
<b>2 显示通路路由</b>	<b>3</b>
<b>3 模块配置介绍</b>	<b>4</b>
3.1 模块功能/配置开关	4
3.1.1 uboot defconfig	4
3.1.2 kernel menuconfig	5
3.1.3 Android	5
3.2 DTS 节点	6
3.2.1 uboot dts	6
3.2.1.1 disp	6
3.2.1.2 hdmi	7
3.2.1.3 edp	7
3.2.1.4 lcd	8
3.2.1.5 tve	8
3.2.2 kernel dts	8
3.2.2.1 disp	8
3.2.2.2 hdmi	10
3.2.2.3 edp	11
3.2.2.4 lcd	11
3.2.2.5 tve	11
3.3 Json 配置文件	11
<b>4 配置流程</b>	<b>14</b>
4.1 uboot 阶段	14
4.1.1 配置通路路由	14
4.1.2 配置设备参数	15
4.1.3 uboot 调试	15
4.2 kernel 阶段	16
4.3 Android 阶段	17
<b>5 双显配置</b>	<b>18</b>
5.1 时钟配置	18
5.2 电源域 (power-domains)	19

5.3 双屏异显配置 . . . . .	19
<b>6 U 盘 ini 文件动态更新配置</b>	<b>20</b>
6.1 功能配置 . . . . .	20
6.1.1 uboot 配置 . . . . .	20
6.1.2 Android 配置 . . . . .	20
6.2 参数配置说明 . . . . .	20
6.2.1 [lcd0]/[lcd1]/[lcd2] . . . . .	20
6.2.2 hdmi . . . . .	21
6.2.3 edp . . . . .	21
6.2.4 [disp0]/[disp1] . . . . .	22
6.2.5 [primary_display] [extend0_display] . . . . .	23
<b>7 调试</b>	<b>24</b>
7.1 uboot . . . . .	24
7.1.1 查看显示驱动状态 . . . . .	24
7.1.2 colorbar 测试 . . . . .	24
7.2 kernel . . . . .	25
7.2.1 查看显示驱动状态 . . . . .	25
7.3 Android . . . . .	25
7.3.1 查看启动设备类型 . . . . .	25

## 插 图

图 2-1	display routing . . . . .	3
图 3-1	uboot disp 配置节点 . . . . .	7
图 3-2	kernel disp 配置节点 . . . . .	10
图 3-3	json 配置节点 . . . . .	13
图 7-1	uboot sys . . . . .	24
图 7-2	kernel_sys . . . . .	25



## 表 格

表 1-1	适用产品 . . . . .	1
表 1-2	相关术语 . . . . .	1
表 3-1	显示框架配置 . . . . .	4
表 3-2	HDMI 配置 . . . . .	4
表 3-3	EDP 配置 . . . . .	4
表 3-4	LCD 配置 . . . . .	5
表 3-5	menuconfig 配置参数 . . . . .	5
表 5-1	Tcon 时钟配置 . . . . .	18



# 1 前言

## 1.1 文档简介

介绍 Sunxi 平台 Disp2 驱动和 Android HWC 相关的一些显示配置，帮助用户快速适配不同显示接口类型。

## 1.2 目标读者

- Display 模块维护/接口适配人员
- Display 模块使用者

## 1.3 适用范围

表 1-1: 适用产品

芯片名称	内核版本	安卓版本
A523 / A527 / T527	Linux5-15	Android13
A523 / A527 / T527	Linux5-15	Android14

## 1.4 相关术语介绍

表 1-2: 相关术语

术语	解释
Disp2	Sunxi 平台显示驱动
HWC	Hardware Composer 硬件抽象层
DE	display engine, 显示引擎, 负责将输入的多图层进行叠加、混合、缩放等处理的硬件模块
Tcon	timing controller, 时序控制器, 负责接收 de 数据信号, 并将其转换为对应控制信号时序
Routing	路由, 显示与设备之间的连接关系
json	轻量配置文件类型, Sunxi hwc 用来存储显示相关配置属性

术语	解释
dtb	Device tree source, 备树的源文件
ini	常用的配置文件格式，由节、键和值组成，用于指定和控制程序的行为，以.ini 结尾





## 2 显示通路路由

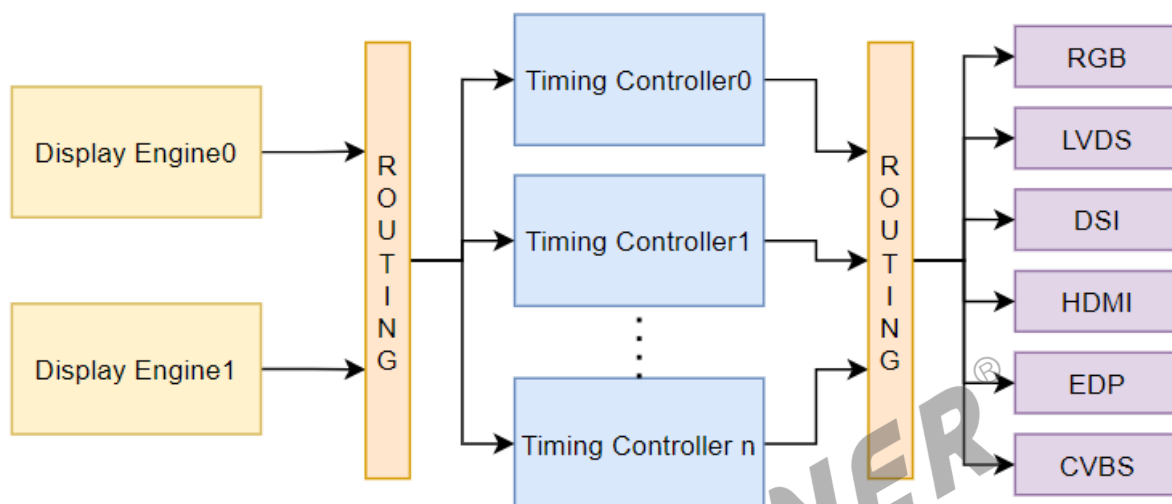


图 2-1: display routing

1. Display Engine 显示引擎简称 DE，芯片中一般集成 1 ~ 2 个 DE，支持两个 DE 的芯片才能实现双显。DE0 和 DE1，一般会有性能差异，以 T527 为例，T527 有两个 DE，DE0 支持 4096 x 2160 的 YUV 输入，DE1 支持 2560 x 1600 输入，两个 DE 均支持 4096 x 2160 输出，所以对于有 4K 视频需求的应用/方案，最好选择 DE0 输出。
2. Routing 显示路由，显示通路中存在两级路由，DE 连接到 Tcon 一级，Tcon 连接到 Interface 一级，在 DTS 配置的各种参数，最终决定底层 DE 连接到哪个 Tcon，Tcon 又连接到哪个 Interface。
3. Interface 接口类型，Sunxi 平台支持多种接口类型，如 RGB/LVDS/MIPI-DSI/HDMI/EDP/CVBS 等，不同的芯片根据定位，支持不同类型的接口，不同显示接口对应不同的驱动，Tcon 区分两种类型，Tcon\_lcd 连接到 lcd 相关接口（RGB，LVDS，DSI），Tcon\_tv 连接到 HDMI，EDP，CVBS 接口。

## 3 模块配置介绍

### 3.1 模块功能/配置开关

#### 3.1.1 uboot defconfig

通过修改对应宏定义为 is not set（关）/ y（开）来配置功能开关。根据方案需求选择对应功能。

配置文件路径：/longan/brandy/brandy-2.0/u-boot-2018/configs/{CHIP}\_{BOARD}\_defconfig

表 3-1: 显示框架配置

名称	值	说明
CONFIG_BOOT_GUI	is not set / y	uboot boot gui 框架
CONFIG_DISP2_SUNXI	is not set / y	uboot 裁剪版显示驱动
CONFIG_CMD_SUNXI_BMP	is not set / y	解析 bmp 格式 bootlogo 功能

接口相关：

#### 1) HDMI

表 3-2: HDMI 配置

名称	值	说明
CONFIG_HDMI2_DISP2_SUNXI	is not set / y	HDMI 对显示框架的对接
CONFIG_INNO_PHY	is not set / y	HDMI 的 phy 选择
CONFIG_HDMI20_UBOOT_DEBUG	is not set / y	HDMI debug 信息

#### 2) EDP

表 3-3: EDP 配置

名称	值	说明
CONFIG_EDP2_DISP2_SUNXI	is not set / y	EDP 对显示框架的对接
CONFIG_AW_EDP2_PHY_USED	is not set / y	EDP 的 phy 选择
CONFIG_DISP_PHY_INNO_EDP_1_3	is not set / y	EDP 的 phy 选择

名称	值	说明
CONFIG_EDP2_SUPPORT_{panel}	is not set / y	EDP 的 panel 选择

### 3) LCD

表 3-4: LCD 配置

名称	值	说明
CONFIG_LCD_SUPPORT_{panel}	is not set / y	LCD 的 panel 选择

## 3.1.2 kernel menuconfig

在命令行中切换到 longan/tina 根目录，执行./build.sh menuconfig 进入配置主界面，修改完后执行./build.sh saveconfig 保存。

显示配置路径：Allwinner BSP > Device Drivers > Video Drivers

表 3-5: menuconfig 配置参数

名称	值	说明
DISP Driver Support(sunxi-disp2)	M / * /	display 驱动开关
debugfs support for disp driver(sunxi-disp2)	* /	对接 debugfs 调试开关
composer support for disp driver(sunxi-disp2)	* /	disp2 的 fence 处理，安卓方案必须选上
HDMI Tx Driver Support(sunxi-disp2)	M / * /	HDMI 接口驱动的开关
TV Driver Support(sunxi-disp2)	M / * /	CVBS 接口驱动的开关
EDP2 Driver Support(sunxi-disp2)	M / * /	EDP 接口驱动的开关
LCD panels select	* /	LCD 屏驱动选择

## 3.1.3 Android

配置文件路径：android/device/softwinner/{CHIP}/{BOARD}/display/config.mk。

**开启双显和 HPD 检测（除了 LCD 单显设备，其余都需要开启）：**

```
DISPLAY_HOTPLUG_SUPPORT := true # false 为关闭
$(call soong_config_add,vendor,hotplugSupport,$(DISPLAY_HOTPLUG_SUPPORT))
```

**安卓设备属性配置方式：**

打开会同步 board.dts 的配置参数，关闭使用 json 参数配置。

```
DISPLAY_SYNC_DEVICE_BOOT_CONFIG:=true  
$(call soong_config_add,vendor,syncDeviceBootConfig,$(DISPLAY_SYNC_DEVICE_BOOT_CONFIG))
```

## 3.2 DTS 节点

### 3.2.1 uboot dts

路径：device/config/chips/{IC}/configs/{BOARD}/uboot-board.dts。

#### 3.2.1.1 disp

配置显示路由 & framebuffer 相关信息

```
&disp {  
    dev_num          = <2>;  
    dev0_output_type  = <1>;  
    screen0_to_lcd_index = <0>;  
    dev0_output_mode  = <4>;  
    dev0_screen_id    = <0>;  
    dev0_do_hpd       = <0>;  
  
    dev1_output_type  = <4>;  
    screen1_to_lcd_index = <2>;  
    dev1_output_mode  = <10>;  
    dev1_screen_id    = <1>;  
    dev1_do_hpd       = <1>;  
  
    fb0_format        = <0>;  
    fb0_width         = <1280>;  
    fb0_height        = <800>;  
  
    fb1_format        = <0>;  
    fb1_width         = <1920>;  
    fb1_height        = <1080>;  
    ...  
}
```

各节点参数含义：

属性	值	说明
dev_num	1/2	uboot 设备数量, AW1890 支持两个设备
dev(n)_screen_id	0/1	device n 连接到哪个 DE 核心, 0为主显, 1为副显
dev(n)_output_type	0: None 1: LCD 2: TV(cvbs) 4: HDMI 8: VGA 32: EDP	device n 输出类型
screen(n)_to_lcd_index	0 ~ 2	使用 &lcd (n) 的屏参, devn_output_type = 1 (LCD) 时有效
dev(n)_output_mode	0 ~ 92	device n 分辨率@帧率配置, 参考sunxi_display2.h: disp_tv_mode配置
dev(n)_do_hpd	0/1	device n 是否开启设备的热插拔检查
fb(n)_format	10: rgb565 bpp=16 8: rgb888 bpp=24 0: argb8888 bpp=32	fb n 存储图像 (bootlogo) 的像素格式, 与分配buf的大小相关
fb(n)_width	N/A	fb n 的宽度, 单位为像素
fb(n)_height	N/A	fb n 的高度, 单位为像素

图 3-1: uboot disp 配置节点

#### 说明

- 1.dev(n)\_screen\_id、dev(n)\_output\_type、screen(n)\_to\_lcd\_index 用来配置设备的 routing , dev(n)\_screen\_id 决定使用哪个 DE, dev(n)\_output\_type、screen(n)\_to\_lcd\_index 共同决定 DE 连接到哪个 Tcon, Tcon 连接到哪个 Interface;
- 2.dev(n)\_output\_mode、dev(n)\_do\_hpd 配置 hpd 设备的默认输出模式和热插拔检测;
- 3.fb(n)\_format、fb(n)\_width、fb(n)\_height 配置 framebuffer 属性和大小;

### 3.2.1.2 hdmi

配置 hdmi 驱动使能信息

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

各节点参数含义：可参考《AW\_HDMI20\_开发指南》

### 3.2.1.3 edp

配置 edp 的使能、timings 参数、屏参数、供电参数等信息

```
&edp0 {
    status = "okay";
    ...
};
```

各节点参数含义：可参考《AW\_EDP\_开发指南》

### 3.2.1.4 lcd

配置 lcd 的使能、timings 参数、屏参数、供电参数等信息

```
&lcd0 {
    lcd_used      = <1>;
    status        = "okay";
    /* 0: HV RGB接口 1: CPU/I80接口 2: Reserved 3: LVDS接口 4: DSI接口 */
    lcd_if        = <4>;
    ...
}
```

各节点参数含义：可参考《AW\_LCD\_开发指南》

### 3.2.1.5 tve

配置 cvbs 驱动使能信息

```
&tv0 {
    status = "okay";
    ...
};
```

各节点参数含义：可参考《Linux\_TVE\_开发指南》

## 3.2.2 kernel dts

路径：device/config/chips/{IC}/configs/{BOARD}/board.dts

### 3.2.2.1 disp

```
&disp {
    screen0_output_type = <1>;
    screen0_output_mode = <4>;
    screen0_to_lcd_index = <0>;

    screen1_output_type = <3>;
    screen1_output_mode = <5>;
    screen1_to_lcd_index = <2>;
    screen1_output_format = <0>;
    screen1_output_bits = <0>;
    screen1_output_eotf = <4>;
    screen1_output_cs = <257>;
    screen1_output_dvi_hdmi = <2>;
    screen1_output_range = <2>;
}
```

```
screen1_output_scan    = <0>;
screen1_output_aspect_ratio = <8>;

fb_format              = <0>;
fb_num                 = <2>;
/* <disp channel layer zorder> */
fb0_map                = <0 1 0 16>;
fb0_width              = <1280>;
fb0_height             = <800>;
/* <disp channel layer zorder> */
fb1_map                = <1 1 0 16>;
fb1_width              = <1920>;
fb1_height             = <1080>;

display_device_num     = <2>;
primary_display_type   = "LCD";
primary_de_id          = <0>;
primary_framebuffer_width = <1280>;
primary_framebuffer_height = <800>;
primary_dpix           = <213>;
primary_dpiy           = <213>;

extend0_display_type   = "HDMI";
extend0_de_id          = <1>;
extend0_framebuffer_width = <1920>;
extend0_framebuffer_height = <1080>;
extend0_dpix           = <160>;
extend0_dpiy           = <160>;
...
}
```

各节点参数含义：

Table:kernel 显示配置节点

属性	值	说明
screen (n)_output_type	0: None 1: LCD 2: TV(CVBS) 3: HDMI 4: VGA 6: EDP	screen n 输出类型
screen(n)_to_lcd_index	0 ~ 2	使用 lcd n 的屏参, devn_output_type = 1 (LCD) 时有效
screen(n)_output_mode	0 ~ 92	screen n 分辨率@帧率配置, 参考sunxi_display2.h: disp_tv_mode配置
fb_num	N/A	fb 的数量
fb_format	8: rgb888 bpp=24 0: argb8888 bpp=32	fb 存储图像 (bootlogo) 的像素格式, 与分配buf的大小相关
fb(n)_width	N/A	fb n 的宽度, 单位为像素
fb(n)_height	N/A	fb n 的高度, 单位为像素
display_device_num	0 ~ 2	Android 层设备数量配置
primary_display_type	LCD HDMI CVBS EDP DP	Android 层主显设备类型
primary_de_id	0/1	Android 层主显绑定的底层 DE id
primary_framebuffer_width	N/A	Android 层主显 UI 分辨率
primary_framebuffer_height	N/A	Android 层主显 UI 分辨率
primary_dpix	N/A	Android 层主显 x 方向 dpi 值
primary_dpiy	N/A	Android 层主显 y 方向 dpi 值

图 3-2: kernel disp 配置节点

#### 说明

- 1.screen (n)\_xxx 属性在开启平滑显示功能 (带 uboot 方案) 时无效, 主要用来在不带 uboot 的方案来配置默认启动参数, 如: fast boot 方案;
- 2.fb(n)\_xxx 这个属性在 uboot 和 kernel 阶段均存在, uboot fb 实现为精简版, kernel fb 对接到原生 fb 框架, 所以需要两套参数来灵活配置参数, 一般这两部分参数是同步的;
- 3.primary\_xxx、extend0\_xxx, 只有开启 **DISPLAY\_SYNC\_DEVICE\_BOOT\_CONFIG** 配置时才有效, extend0 表示配置副显

### 3.2.2.2 hdmi

配置 hdmi 驱动使能信息

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

各节点参数含义：可参考《AW\_HDMI20\_开发指南》



### 3.2.2.3 edp

配置 edp 的使能、timings 参数、屏参数、供电参数等信息

```
&edp0 {  
    status = "okay";  
    ...  
};
```

各节点参数含义：可参考《AW\_EDP\_开发指南》

### 3.2.2.4 lcd

配置 lcd 的使能、timings 参数、屏参数、供电参数等信息

```
&lcd0 {  
    lcd_used    = <1>;  
    status      = "okay";  
    /* 0: HV RGB接口 1: CPU/I80接口 2: Reserved 3: LVDS接口 4: DSI接口 */  
    lcd_if      = <4>;  
    ...  
}
```

各节点参数含义：可参考《AW\_LCD\_开发指南》

### 3.2.2.5 tve

配置 cvbs 驱动使能信息

```
&tv0 {  
    status = "okay";  
    ...  
};
```

各节点参数含义：可参考《Linux\_TVE\_开发指南》

## 3.3 Json 配置文件

显示 json 配置文件，用来配置 Android 的设备路由和输出类型。

路径：android/device/softwinner/{CHIP}/{BOARD}/display/dispconfigs/{BOARD}.json

```
{  
    "DisplayEngineVersion": 2,  
    "DisplayNumber": 3, /* Display0, Display1, Display2 */  
  
    "PrimaryDisplay": ["Display0"],  
    "ExternalDisplay": ["Display1", "Display2"],  
}
```

```
"Display0": {  
  /* Available interface type: LCD/HDMI/CVBS */  
  "InterfaceType": "LCD",  
  "DisplayEnginePortId": 0,  
  "OverrideFramebufferSize": 1,  
  "FramebufferWidth": 1280,  
  "FramebufferHeight": 800,  
  "DpiX": 213,  
  "DpiY": 213,  
  "HotplugSupported": 0,  
  "AfbcSupported": 0,  
  "HardwareRotateSupported": 0,  
  "BandwidthLimited": 338800,  
  "DefaultOutputMode": 0  
},  
  
"Display1": {  
  /* Available interface type: LCD/HDMI/CVBS */  
  "InterfaceType": "HDMI",  
  "DisplayEnginePortId": 1,  
  "OverrideFramebufferSize": 1,  
  "FramebufferWidth": 1920,  
  "FramebufferHeight": 1080,  
  "DpiX": 160,  
  "DpiY": 160,  
  "HotplugSupported": 0,  
  "AfbcSupported": 0,  
  "HardwareRotateSupported": 0,  
  "BandwidthLimited": 338800,  
  "DefaultOutputMode": 0  
},  
  
"Display2": {  
  /* Available interface type: LCD/HDMI/CVBS */  
  "InterfaceType": "CVBS",  
  "DisplayEnginePortId": 1,  
  "OverrideFramebufferSize": 1,  
  "FramebufferWidth": 1280,  
  "FramebufferHeight": 720,  
  "DpiX": 160,  
  "DpiY": 160,  
  "HotplugSupported": 0,  
  "AfbcSupported": 0,  
  "HardwareRotateSupported": 0,  
  "BandwidthLimited": 338800,  
  "DefaultOutputMode": 0  
}  
}
```

各节点参数含义:

属性	值	说明
PrimaryDisplay	Display(n)	主显使用哪个接口作为显示设备
ExternalDisplay	Display(n)	副显使用哪个接口作为显示设备，上述案例中使用Display1和Display2作为副显，但是由于实际硬件只有两个显示引擎（DE），所以只能有一个副显，会从Display1和Display2中通过检测设备接入状态来决定哪个作为副显，如实际硬件有显示器通过HDMI线接入，则使用HDMI作为副显输出。
InterfaceType	LCD HDMI CVBS EDP DP	显示设备的接口类型
DisplayEnginePortId	0/1	显示设备通过哪个显示引擎（DE）输出
FramebufferWidth	N/A	显示设备的 FB 宽度
FramebufferHeight	N/A	显示设备的 FB 高度
DpiX	N/A	显示设备的 x 方向的 dpi 值
DpiY	N/A	显示设备的 y 方向的 dpi 值

图 3-3: json 配置节点

#### 说明

1. 未在表格中说明的参数，一般无需修改，其中部分参数是预留。
2. 如果是新建的方案，或方案是从别的现有方案拷贝而来，需要注意拷贝完成后将 json 文件的名字改成 ro.product.vendor.device 属性的值一致，如：ro.product.vendor.device 值为 a527-pro2 时，json 文件需要命名为 a527-pro2.json。文件名与属性名一致后才能确保在初始化时找到对应文件进行解析。ro.product.vendor.device 的值通过 android/out/target/product/{BOARD}/vendor/build.prop 查看。

## 4 配置流程

Android 设备启动过程分为几个阶段：uboot -> kernel -> android，这几个阶段都需要显示功能，后面的阶段一般需要同步/获取前面阶段的参数来进行策略的判断和优化。所以我们可以按这个路径一步步把显示调起来：uboot -> kernel -> android。下面以 LCD + HDMI 双显为例，简述适配新的接口组合的一个流程。

### 4.1 uboot 阶段

LCD + HDMI 我们需要配置的相关节点有：`&disp`、`&hdmi`、`&lcd`。uboot 配置文件路径

#### 4.1.1 配置通路路由

配置完后的模板：

```
&disp {
    dev_num            = <2>;
    dev0_screen_id     = <0>;
    dev0_output_type   = <1>;
    screen0_to_lcd_index = <0>;

    dev1_screen_id     = <1>;
    dev1_output_type   = <4>;
    dev1_output_mode   = <10>;
    dev1_do_hpd        = <1>;

    fb0_format         = <0>;
    fb0_width          = <1280>;
    fb0_height         = <800>;

    fb1_format         = <0>;
    fb1_width          = <1920>;
    fb1_height         = <1080>;
    ...
}
```

可以分为几个部分来说明：

1) 设备数量：LCD + HDMI 是双显，配置 `dev_num = <2>`，可以在 uboot 阶段就开启两个设备，也可以配置为 `<1>`，这表示在 uboot 阶段不开启双显，后续阶段配置设备数量为 2，可以后续阶段开启双显。

2) Routing: 显示通路的路由, 模板中的路由为: de0 -> lcd0 (dev0\_screen\_id、dev0\_output\_type、screen0\_to\_lcd\_index 由这三个参数配置), de1-> hdmi (dev1\_screen\_id、dev1\_output\_type), 并配置默认输出模式和检测热插拔 (dev1\_output\_mode, dev1\_do\_hpd)

3) framebuffer: 模板中的配置表示分配两块 framebuffer0 (1280 x 800) 和 framebuffer1 (1920 x 1080), 格式为 argb8888, framebuffer0 用来存储主显 LCD 的 logo, LCD 屏幕为 1280 x 800, framebuffer1 存储副显 HDMI 的 logo, 我们设置默认模式为 10 (1080p@60), 这两个大小设置不能小于输出设备大小, 用于点对点的显示 logo。

## 4.1.2 配置设备参数

### 1) HDMI

```
&hdm1 {
    status = "okay";
    ...
};
```

一般来说, HDMI 参数在 SDK 中已配置好, 只需要打开相关节点。

### 2) LCD

```
&lcd0 {
    lcd_used      = <1>;
    status        = "okay";
    /* 0: HV RGB接口 1: CPU/I80接口 2: Reserved 3: LVDS接口 4: DSI接口 */
    lcd_if        = <3>;
    ...
}
```

LCD 节点我们需要打开相关节点, 并配置类型 lcd\_if 为对应屏幕类型, 如果使用为原厂已经调试好的屏幕, 可以直接使用原厂的参数和屏驱动配置, 如果是新的屏幕, 需要根据屏幕手册/datasheet, 来适配/移植新的屏参数和驱动。具体移植方法可以参考: 《Linux\_LCD\_开发指南》

#### 说明

这里使用的是 &lcd0 节点来配置, 是因为 screen0\_to\_lcd\_index 节点配置为 0, 表示使用 &lcd0 的参数来初始化屏, 不同的 lcd 节点, 底层对应不同的 tcon\_lcd, 支持不同的 lcd 接口类型。

## 4.1.3 uboot 调试

机器连接串口后, 上电后, 串口输入按住”s“键不松, 可以进入 uboot 命令行模式, 进行一些简单的调试。若此时可以看到 IC 型号的相关 logo 显示正常, 证明 uboot 配置成功。若显示异常可按下面步骤进行调试:

1. 输入 colorbar 1, 查看是否有显示, 若无显示, 需要检查接口设备相关参数是否正确, 有显示代表设备输出端参数正确, 继续排查;

2. 输入 `colorbar 8`，查看是否有显示，若有显示，但 logo 无显示，应该是 fb 设置大小小于 `bootlogo.bmp` 大小，可修改 `bootlogo.bmp` 大小解决，若无显示，继续排查；
3. 输入 `disp`，查看显示系统信息是否有问题，接口类型，routing，中断，若有异常，可以寻求技术支持并附上串口启动日志和 dts 配置文件。

## 4.2 kernel 阶段

方案默认开启平滑启动功能，kernel 阶段只需要同步 uboot 的部分配置即可，**kernel 配置文件**，同步 uboot 相关配置：

### 1) Disp

```
&disp {
    fb_format      = <0>;
    fb_num         = <2>;
    /* <disp channel layer zorder> */
    fb0_map        = <0 1 0 16>;
    fb0_width      = <1280>;
    fb0_height     = <800>;
    /* <disp channel layer zorder> */
    fb1_map        = <1 1 0 16>;
    fb1_width      = <1920>;
    fb1_height     = <1080>;
    ...
}
```

同步 uboot fb 大小。

### 2) HDMI

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

使能 kernel 阶段 hdmi 设备。

### 3) LCD

```
&lcd0 {
    lcd_used       = <1>;
    status         = "okay";
    /* 0: HV RGB接口 1: CPU/I80接口 2: Reserved 3: LVDS接口 4: DSI接口 */
    lcd_if         = <3>;
    ...
}
```

同步 uboot 适配好的屏参配置。

## 4.3 Android 阶段

主要需要配置通路路由和 UI 分辨率等相关属性，可以使用 json 和 dts 两种配置方式，由 `DISPLAY_SYNC_DEVICE_BOOT_CONFIG` 这个宏控制。

1) json 配置参考，[LCD + HDMI 配置](#)

2) dts 配置参考：

```
&disp {
    ...
    display_device_num    = <2>;
    primary_display_type  = "LCD";
    primary_de_id         = <0>;
    primary_framebuffer_width = <1280>;
    primary_framebuffer_height = <800>;
    primary_dpix          = <213>;
    primary_dpiy          = <213>;

    extend0_display_type  = "HDMI";
    extend0_de_id         = <1>;
    extend0_framebuffer_width = <1920>;
    extend0_framebuffer_height = <1080>;
    extend0_dpix          = <160>;
    extend0_dpiy          = <160>;
    ...
}
```

## 5 双显配置

### 5.1 时钟配置

A527&T527 平台 Tcon 数量比较多，但现有只要两个空闲的父时钟源，需要将需要使用的 Tcon，分到两个父时钟上，各 Tcon 对应接口，与可分配时钟：

表 5-1: Tcon 时钟配置

Tcon	接口类型	对应 dts 节 点	对应配置时钟	可配置父时钟
tcon0	RGB0、 LVDS0、 MIPI-DSI0	lcd0	CLK_VO0_TCONLCD0、 CLK_COMBPHY0 (DSI0/ LVDS0)	CLK_PLL_VIDEO0_4X、 CLK_PLL_VIDEO1_4X
tcon1	MIPI-DSI1	lcd1	CLK_VO0_TCONLCD1、 CLK_COMBPHY1 (DSI1)	CLK_PLL_VIDEO0_4X、 CLK_PLL_VIDEO1_4X
tcon2	HDMI	hdmi	CLK_TCONTV	CLK_PLL_VIDEO0_4X、 CLK_PLL_VIDEO1_4X
tcon3	EDP	edp	CLK_TCONTV1、CLK_EDP	CLK_PLL_VIDEO0_4X、 CLK_PLL_VIDEO1_4X
tcon4	RGB1、LVDS1	lcd2	CLK_VO1_TCONLCD0	CLK_PLL_VIDEO0_4X、 CLK_PLL_VIDEO1_4X

uboot 配置路径：longan/brandy/brandy-2.0/u-boot-2018/arch/arm/dts/{CHIP}-clk.dtsi

kernel 配置路径：device/config/chips/{IC}/configs/{BOARD}/board.dts

对应配置：

```
assigned-clocks =
    <&ccu CLK_VO0_TCONLCD0>,
    <&ccu CLK_VO0_TCONLCD1>,
    <&ccu CLK_VO1_TCONLCD0>,
    <&ccu CLK_TCONTV>,
    <&ccu CLK_TCONTV1>,
    <&ccu CLK_COMBPHY0>,
    <&ccu CLK_COMBPHY1>,
    <&ccu CLK_EDP>;
assigned-clock-parents =
    <&ccu CLK_PLL_VIDEO0_4X>,
```



```
<&ccu CLK_PLL_VIDEO1_4X>,  
<&ccu CLK_PLL_VIDEO1_4X>,  
<&ccu CLK_PLL_VIDEO1_4X>,  
<&ccu CLK_PLL_VIDEO1_4X>,  
<&ccu CLK_PLL_VIDEO0_4X>,  
<&ccu CLK_PLL_VIDEO1_4X>,  
<&ccu CLK_PLL_VIDEO1_4X>;
```

使用时钟框架 assigned-clocks, assigned-clock-parents 来手动指定父时钟

默认配置为将 CLK\_VO0\_TCONLCD0 + CLK\_COMBPHY0 (RGB0、LVDS0、MIPI-DSI0) 绑定在 CLK\_PLL\_VIDEO0\_4X, 其他 (MIPI-DSI1、RGB1、LVDS1、HDMI、EDP) 绑定在 CLK\_PLL\_VIDEO1\_4X, 这样配置可以满足大部分接口的组合。

如果需要特殊接口组合如 MIPI-DSI1 + EDP, 先看能不能切到 MIPI-DSI0 + EDP 来实现, 如果不行, 就可以切换父时钟到 CLK\_PLL\_VIDEO0\_4X 来解决可能会发生的时钟冲突问题。其他接口时钟配置情况类似。

## 5.2 电源域 (power-domains)

A523/A527/T527 平台, 支持电源域控制各模块电源, SDK 配置默认开启 “pd\_de”, “pd\_vo0” 电源, 对应可以使用 tcon 0 ~ 3 对应的接口, 如果需要使用 tcon4 的接口, 需要在 board.dts, 种加上对 “pd\_vo1” power-domains 资源的引用。参考:

```
&disp {  
    ...  
    power-domains = <&pd1 A523_PCK_DE>, <&pd1 A523_PCK_VO0>, <&pd1 A523_PCK_VO1>;  
    power-domain-names = "pd_de", "pd_vo0", "pd_vo1";  
    ...  
}
```

## 5.3 双屏异显配置

A527&T527 支持双屏同显和双屏异显, 如果需要支持双屏异显, 需要 APK 应用支持:

使用特定方法 (如 presentation) 编写 APK, 选择显示图层需要送往的屏幕。

可参考: <https://developer.android.google.cn/reference/android/app/Presentation#public-constructors>

## 6 U 盘 ini 文件动态更新配置

U 盘 ini 文件动态更新显示配置（下文统一称作 U 盘更新）功能主要用于客户有一个固件兼容多种显示接口输出的需求。在开启了 U 盘更新功能使能的情况下，用户可以事先按照特定的格式编写好配置文件，文件命名为 disp\_config.ini 并拷贝至 U 盘，在机器启动时通过长按音量+/音量-键，触发读取 U 盘参数完成更新的动作，即可做到在不修改源码和 dts，不需要重新编译打包固件的情况下修改机器的显示输出类型。

### 6.1 功能配置

#### 6.1.1 uboot 配置

- 开启 U 盘读取更新 ini 文件

配置文件路径：andriod/longan/brandy/brandy-2.0/u-boot-2018/configs/sun55iw3p1\_{BOARD}\_defconfig

```
CONFIG_AIOT_DISP_PARAM_UPDATE=y #开启  
#CONFIG_AIOT_DISP_PARAM_UPDATE is not set #关闭
```

#### 6.1.2 Android 配置

开启 HWC 从 DTS 读取设备信息 `DISPLAY_SYNC_DEVICE_BOOT_CONFIG` 宏

### 6.2 参数配置说明

#### 6.2.1 [lcd0]/[lcd1]/[lcd2]

与 `&lcd` 节点含义相同，节点名字对比去掉了 lcd。

- 支持配置节点

```
used=1  
panel_driver_name=bp101wx1  
dclk_freq=0
```

```
ht=0
x=0
hspw=0
hbp=0
vt=0
y=0
vbp=0
vspw=0
tcon_mode=0
tcon_en_odd_even_div=0
start_delay=0
backlight=0
backlight_delay=0
pwm_used=0
pwm_ch=0
pwm_freq=0
if=0
lvds_if=0
lvds_colordepth=0
lvds_mode=0
dsi_if=0
dsi_lane=0
dsi_format=0
dsi_port_num=0
dsi_eotp=0
dsi_te=0
sync_pixel_num=0
sync_line_num=0
hv_clk_phase=0
hv_sync_polarity=0
hv_data_polarity=0
hv_srgb_seq=0
hv_syuv_seq=0
hv_syuv_fdly=0
hv_if=0
```

## 6.2.2 hdmi

- 支持配置节点

status=okay ;是否开启hdmi okay 开启 disabled 关闭

## 6.2.3 edp

与 &edp 节点含义相同，节点名字对比去掉了 edp。

- 支持配置节点

```
compatible = allwinner,sunxi-edp0
status=disabled
panel_driver_name=VWX10T025J00_2560X1600
timings_type=0
```

```
ht=3504
x=2560
hbp=472
hspw=280
hfp=192
vt=1658
y=1600
vbp=49
vspw=6
vfp=3
hpolor=1
vpolor=1
fps=60
ssc_en=0
ssc_mode=0
audio_en=0
color_depth=8
color_format=0
lane_rate=1
lane_cnt=4
lane0_sw=0
lane0_pre=0
lane1_sw=0
lane1_pre=0
lane2_sw=0
lane2_pre=0
lane3_sw=0
lane3_pre=0
efficient_training=0
edp_panel_used=1
pwm_used=1
pwm_freq=50000
pwm_pol=0
default_backlight=200
```

## 6.2.4 [disp0]/[disp1]

### uboot/linux 显示参数配置

- 支持配置节点

```
dev_num=2      ;uboot/kernel需要开启的设备数量，[disp0]特有
chn_cfg_mode=3 ;硬件chn通达分配策略，双显一般为3，副显需要高规格（分辨率>=2.5k），设置为4，[disp0]特有
lcd_index=2    ;lcd设备逻辑索引，type=lcd时使用，可选0，1，2
output_type=3  ;设备类型 0: NONE 1:LCD 2:TV 3:HDMI 4:VGA 6:EDP
output_mode=10 ;TV设备mode 0:480i 1:576i 2:480p 3:576p 4:720p50 5:720p60 6:1080i50 7:1080i60 8:1080p24 9:1080p50 10:1080p60
screen_id=0    ;display engine 索引
do_hpd=1       ;设备是否需要检测热插拔
fb_width=1920  ;fb width 存储logo图片（不能小于logo图片大小）
fb_height=1080 ;fb height 存储logo图片（不能小于logo图片大小）
```

## 6.2.5 [primary\_display] [extend0\_display]

### Android 显示参数配置

- 支持配置节点

```
device_num=2      ;Android 需要开启设备属性，[primary_display]特有
display_type=HDMI  ;设备类型，支持NONE、LCD、CVBS、HDMI、VGA、VDPO、EDP、DP
de_id=0           ;设备硬件索引，一般与disp节点对应
framebuffer_width=1920 ;UI分辨率/图像源width
framebuffer_height=1080 ;UI分辨率/图像源height
dpix=160          ;设备dpix
dpiy=160          ;设备dpiy
```



## 7 调试

### 7.1 uboot

机器连接串口后，上电后，串口输入按住”s“键不松，可以进入 uboot 命令行模式。

#### 7.1.1 查看显示驱动状态

disp

```
=> disp
screen 0:
de_rate 600000000 hz, ref_fps:60
mgr0: 1280x720 fat[pw444] --[0x10] range[limit] eof[0x4] bits[0bits] err[1] force_sync[0] unblank direct_show[false] lomu[0]
hdmi output mode[5]      fps:60.6      1280x 720
err:0 skip:0 irq:394178 vsync:0 vsync_skip:0

COLOR enable ch[1] lyr[0] z[16] prem[N] a[pixel] 0] fat[ 0] fb[ 0, 0; 0, 0; 0, 0] crop[ 0, 0, 320, 720] frame[ 0, 0, 320, 720] addr[ffff0000, 0, 0] flags[0x 0] trd[0,0]
COLOR enable ch[1] lyr[1] z[16] prem[N] a[pixel] 0] fat[ 0] fb[ 0, 0; 0, 0; 0, 0] crop[ 320, 0, 320, 720] frame[ 320, 0, 320, 720] addr[ffff0000, 0, 0] flags[0x 0] trd[0,0]
COLOR enable ch[1] lyr[2] z[16] prem[N] a[pixel] 0] fat[ 0] fb[ 0, 0; 0, 0; 0, 0] crop[ 640, 0, 320, 720] frame[ 640, 0, 320, 720] addr[ffff0000, 0, 0] flags[0x 0] trd[0,0]
COLOR enable ch[1] lyr[3] z[16] prem[N] a[pixel] 0] fat[ 0] fb[ 0, 0; 0, 0; 0, 0] crop[ 960, 0, 320, 720] frame[ 960, 0, 320, 720] addr[ffff0000, 0, 0] flags[0x 0] trd[0,0]
```

图 7-1: uboot sys

可以查看显示的输出分辨率（1280 X 720），routing 路由状态（DE0 -> HDMI），初始化输出模式（mode = 5）等

#### 7.1.2 colorbar 测试

colorbar num

使用内建数据测试，num 可选参数及含义：

- 0: DE 直接输出（默认模式）
- 1 ~ 7: TCON 内建数据输出
- 8: DE 内建 colorbar 输出

## 7.2 kernel

### 7.2.1 查看显示驱动状态

```
cat /sys/class/disp/disp/attr/sys
```

```
diana-pi:/ # cat /sys/class/disp/disp/attr/sys
screen[0] -> dev[2]
de_rate 600000000 hz, ref_fps:60
mgr0: 1920x1080 fmt[yuv444] cs[0x101] range[limit] eotf[0x4] bits[8bits] err[0] force_sync[22306] unblank_direct_show[false] iommu[1] rcq_en[1]
rcq info: rcq_irq[89229] rcq_update_request[89228] rcq_update_req_irq[89516] rcq_finish_irq[89517]
dmapbuf: cache[8] cache_max[12] umap_skip[0] umap_skip_max[0]
hdmi output mode[10] fps:60.6 1920x1080
err:0 skip:0 lrq:89517 vsync:89288 vsync_skip:0
BUF enable ch[0] lyr[0] z[0] prem[N] a[global 255] fmt[ 72] fbd_type[none] fb[1920,1088; 960, 544; 960, 544] crop[ 0, 0,1920,1080] frame[ 0, 0
,1920,1080] addr[fc00000,fc7d800,fc7fe000] flags[0x 0] metadata_flag[0x 0] trd[0,0] depth[ 0] transf[0] sampling[1920:1920,1080:1080]
BUF enable ch[2] lyr[0] z[1] prem[Y] a[global 255] fmt[ 1] fbd_type[afbd] fb[1920,1088; 0, 0; 0, 0] crop[ 0, 0,1920,1080] frame[ 0, 0
,1920,1080] addr[fc400000,fc5fe000,fc7fc000] flags[0x 0] metadata_flag[0x 10] trd[0,0] depth[ 0] transf[0] sampling[1920:1920,1080:1080]
disp0 all:89224, sub:89224, cur:89224, free:89206, skip:0
```

图 7-2: kernel\_sys

可以查看显示的输出分辨率（1920 X 1080），routing 路由状态（DE0 -> HDMI），初始化输出模式（mode = 10）等

#### 调试说明：

可以重点关注这几个信息来判断系统是否正常工作：

1. 连续 cat sys 信息，对比 “err:xxx”、“vsync:xxx” 信息，如果 err 数有连续增加，大概率会有闪烁/花屏现象。vsync 计数是否一直递增，若 vsync 不递增，代表 vsync 中断刷新异常；
2. 查看 “ref\_fps:xxx”、“fps:xxx” 是否与输出设备刷新率相同，如果 ref\_fps 与输出设备刷新率不同时，需要检测 DTS 的 timing 配置是否合理；

## 7.3 Android

### 7.3.1 查看启动设备类型

```
# logcat -s displayd
----- beginning of main
01-01 08:00:19.462 332 387 W displayd: xxxxxxx: threadedStartup begin
01-01 08:00:19.462 332 387 W displayd: xxxxxxx: HAL device config from DTS
01-01 08:00:19.462 332 387 W displayd: Displays (3 entries)
01-01 08:00:19.462 332 387 W displayd: + Display0{logicid=0, framebuffer=1280x800(override=1), dpi=213x213}
01-01 08:00:19.462 332 387 W displayd: DisplayEnginePortId=0
01-01 08:00:19.462 332 387 W displayd: InterfaceType=LCD
01-01 08:00:19.462 332 387 W displayd: DefaultOutputMode=0
01-01 08:00:19.462 332 387 W displayd: HotplugSupported=0
01-01 08:00:19.462 332 387 W displayd: AfbSupported=0
01-01 08:00:19.462 332 387 W displayd: HardwareRotateSupported=0
01-01 08:00:19.462 332 387 W displayd: BandwidthLimited=0 Bytes per frame
01-01 08:00:19.462 332 387 W displayd: + Display1{logicid=1, framebuffer=1920x1080(override=1), dpi=160x160}
```

```
01-01 08:00:19.462 332 387 W displayd: DisplayEnginePortId=1
01-01 08:00:19.462 332 387 W displayd: InterfaceType=HDMI
01-01 08:00:19.462 332 387 W displayd: DefaultOutputMode=0
01-01 08:00:19.462 332 387 W displayd: HotplugSupported=0
01-01 08:00:19.462 332 387 W displayd: AfbSupported=0
01-01 08:00:19.462 332 387 W displayd: HardwareRotateSupported=0
01-01 08:00:19.462 332 387 W displayd: BandwidthLimited=0 Bytes per frame
01-01 08:00:19.462 332 387 W displayd: + Display2{logicid=1, framebuffer=1920x1080(override=1), dpi=160x160}
01-01 08:00:19.462 332 387 W displayd: DisplayEnginePortId=1
01-01 08:00:19.462 332 387 W displayd: InterfaceType=DP
01-01 08:00:19.462 332 387 W displayd: DefaultOutputMode=0
01-01 08:00:19.462 332 387 W displayd: HotplugSupported=0
01-01 08:00:19.462 332 387 W displayd: AfbSupported=0
01-01 08:00:19.462 332 387 W displayd: HardwareRotateSupported=0
01-01 08:00:19.462 332 387 W displayd: BandwidthLimited=0 Bytes per frame
...
```

如果 log 信息为” HAL device config file: /vendor/etc/dispconfigs/{BOARD}.json” ，表示使用 json 配置更新

如果 log 信息为 “xxxxxxx: HAL device config from DTS” ，表示使用 DTS 配置更新

可以检查对应设备的 InterfaceType 等信息是否与预期一致。








## 著作权声明

版权所有 © 2024 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。