



A20 显示配置 和接口说明文档

V1.0

2014-06-05



Revision History

Version	Date	Changes compared to previous issue
1.0	2014-04-26	

For Allwinnertech Only



目录

1. 概述.....	4
1.1. 编写目的.....	4
1.2. 使用范围（IC 平台 and 软件版本）	4
1.3. 目标读者（内部 or 外发）	4
2. Boot 阶段显示配置.....	5
2.1. sys_config.fex 配置文件.....	5
2.2. boot 显示流程.....	5
2.3. Bootlogo 的规格要求.....	6
3. Kernel 的显示驱动.....	7
4. Android 的显示.....	8
4.1. 热插拔消息处理.....	8
4.2. 显示接口.....	9
5. Declaration.....	16



1. 概述

1.1. 编写目的

让开发人员或用户可以快速的了解显示模块的基本配置。

1.2. 使用范围（IC 平台 and 软件版本）

A20 平台 sugar v2.1。

1.3. 目标读者（内部 or 外发）

For Allwinnertech Only



2. Boot 阶段显示配置

Boot 阶段显示相关的代码在：

- (1)、lichee/brandy/u-boot-2011.09/board/sunxi/de.c，显示策略与打开显示设备的代码。
- (2)、lichee/brandy/u-boot-2011.09/drivers/video_sunxi/，显示驱动。

2.1.sys_config.fex 配置文件

Boot 显示初始化时，会读取该文件的配置项来初始化输出模式和是否支持热插拔，该文件位于 lichee/tools/pack/chips/sun6i/configs/android/XXX/ 下（其中 XXX 是具体方案，比如 sugar-ref001）。

```
;-----  
;boot display configuration  
;output_type (0:none; 1:lcd; 2:tv; 3:hdmi; 4:vga)  
;output_mode (used for tv/hdmi output, 0:480i 1:576i 2:480p 3:576p 4:720p50 5:720p60 6:1080i50  
              7:1080i60 8:1080p24 9:1080p50 10:1080p60 11:pal 14:ntsc)  
;output_mode (used for vga output, 0:1680*1050 1:1440*900 2:1360*768 3:1280*1024 4:1024*768  
              5:800*600 6:640*480 10:1920*1080 11:1280*720)  
;auto_hpd (auto detect hdmi/cvbs/ypbpr plug in)  
;-----  
  
Example:  
[boot_disp]  
output_type      = 2  
output_mode      = 5  
auto_hpd         = 1
```

sys_config.fex 关于 boot 阶段显示的配置项如下：

其中，output_type 用于在没有插入任何输出设备的情况下默认输出设备类型，output_mode 为默认输出设备的输出模式，auto_hpd 表示该机器是否支持输出设备的热插拔检测，auto_hpd = 1 表示支持热插拔检测。

在 boot 代码中，通过宏 script_parser_fetch，去读取该配置文件的 boot_disp 下的选项，分别初始化变量 output_type、output_mode、auto_hpd。

2.2.boot 显示流程

- (1) 若主键[boot_disp]配置了子键 auto_hpd = 1,则会根据热插拔检测当前显示设备。
 - ① 首先检测 hdmi 热插拔，如果检测到 hdmi 设备插入，则把显示设备输出类型设置为 hdmi。
 - ② 如果没有检测到 hdmi 设备插入，接着检测 cvbs 热插拔，如果检测到 cvbs 设备插入，则输出类型设置为 cvbs。
- (2) 若主键[boot_disp]没有配置子键 auto_hpd = 1，则会把显示设备的输出类型设置为子键 output_type 的值。
- (3) 确定输出模式。



①根据输出类型从 bootloader 私有分区的文件 config.fex 读取对应的显示模式。Config.fex 文件里面的显示模式由 android 系统写入,从 config.fex 文件里读取是为了以上次 android 系统使用的显示模式进行开机。

②若从 config.fex 读取显示模式有误,则使用子键 output_mode 的值作为显示设备的输出模式。

Boot 的显示策略的具体代码实现详见 lichee/brandy/u-boot-2011.09/board/sunxi/de.c 的 board_display_device_open(void)函数。显示流程如下图:

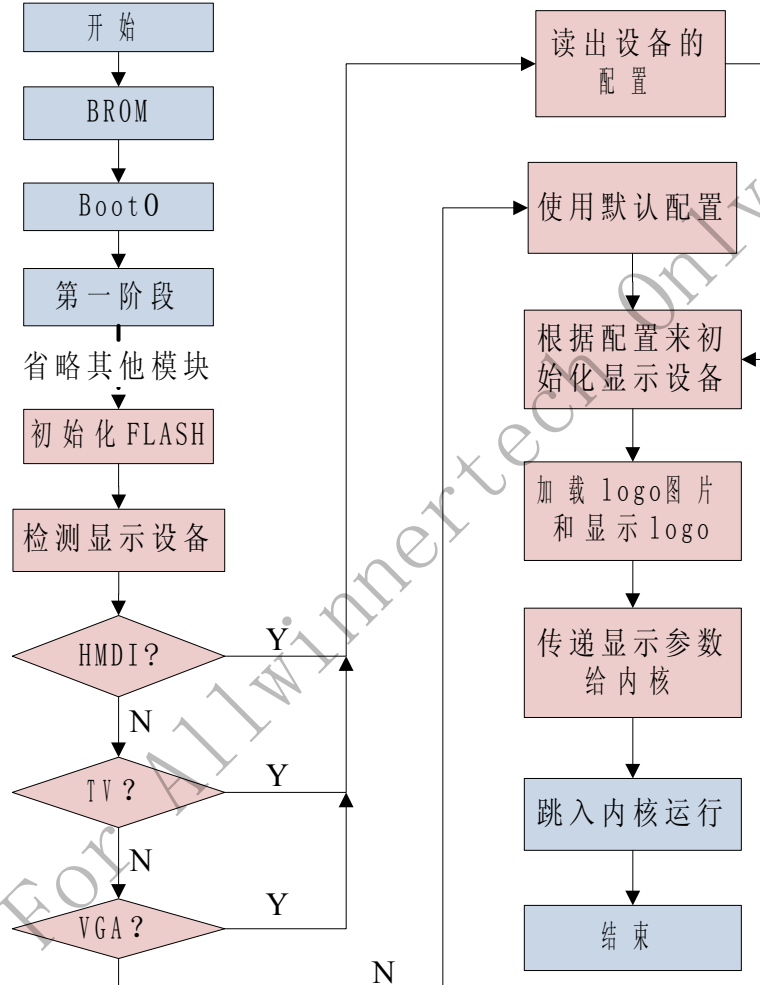


图 1 boot 的显示流程

2.3. Bootlogo 的规格要求

- (1) Bootlogo 格式必须是 32 位的 bmp 格式的图片。
- (2) Bootlogo 图片的最大支持分辨率为 1920*1080, 建议使用不超过分辨率 1280*720 的 bmp 图片。
- (3) Bootlogo 图片的存放位置为: lichee/tools/pack_brandy/chips/sun7i/boot-resource/boot-resource, 必须以 bootlogo.bmp 为名字命名。



3. Kernel 的显示驱动

显示驱动加载有两种方式，一种是 built in ，另一种是编译成 ko 库，在内核启动后，加载这些 ko 库。现在 A20 的显示驱动用的就是第一种方式。如果要编译成 ko，可以在 lichee/linux3.3/目录下，运行：make ARCH=arm menuconfig，对其进行配置。

当加载完驱动时，就会进行初始化工作，首先，读取 sys_config.fex 的一些配置项，并初始化相应的变量，并按照配置打开相应的输出设备。用于驱动初始化的配置在 sys_config.fex 中，如下：

```
;-----  
;disp init configuration  
;  
;disp_mode  
;  
;          0:screen0<screen0,fb0>  
;  
;          1:screen1<screen1,fb0>  
;  
;          2:two_diff_screen_diff_contents<screen0,screen1,fb0,fb1>  
;  
;          3:two_same_screen_diff_contets<screen0,screen1,fb0>  
;  
;          4:two_diff_screen_same_contents<screen0,screen1,fb0>  
;screenx_output_type (0:none; 1:lcd; 2:tv; 3:hdm; 4:vga)  
;screenx_output_mode (used for tv/hdmi output, 0:480i 1:576i 2:480p 3:576p 4:720p50 5:720p60 6:1080i50  
;  
;          7:1080i60 8:1080p24 9:1080p50 10:1080p60 11:pal 14:ntsc)  
;screenx_output_mode (used for vga output, 0:1680*1050 1:1440*900 2:1360*768 3:1280*1024 4:1024*768  
;  
;          5:800*600 6:640*480 10:1920*1080 11:1280*720)  
;fbx format (4:RGB655 5:RGB565 6:RGB556 7:ARGB1555 8:RGBA5551 9:RGB888  
;  
;          10:ARGB8888 12:ARGB4444)  
;fbx pixel sequence (0:ARGB 1:BGRA 2:ABGR 3:RGBA) --- 0 for linux, 2 for android  
;-----  
[disp_init]  
disp_init_enable = 1  
disp_mode = 4  
screen0_output_type = 3  
screen0_output_mode = 5
```

如果需要自定义，可阅读配置项说明。

显示驱动程序加载后，会进行初始化。如果 boot 阶段正常初始化并显示，kernel 阶段的初始化就不会重新对硬件进行初始化，也不重新打开显示设备。否则以 sys_config.fex 配置项的模式输出。

特别说明：screen0_output_type 和 screen0_output_mode 的配置会影响到 framebuffer 的 size。例如，配置了 HDMI720P，那么初始化时 framebuffer 的大小为 720P。一般用默认配置就可以，不建议修改。

4. Android 的显示

4.1. 热插拔消息处理

A20 目前支持两种显示设备的热插拔消息处理：HDMI 和 CVBS。处理策略如下：

A20显示的热插拔消息处理策略		
1. 优先级：hdmi 高，cvbs 低。		
2. 默认输出：hdmi-720P60HZ，cvbs-N 制。		
3. 具体操作与相应的状态转换：		
当前状态	具体操作	操作后状态
只插着 hdmi, 系统输出 hdmi	拔出 hdmi 线	输出 NONE
只插着 cvbs, 系统输出 cvbs	拔出 cvbs 线	输出 NONE
同时插着 hdmi 和 cvbs	拔出当前显示类型的设备线 (例如当前输出 hdmi, 拔出 hdmi 线)	切换到另一显示类型
	拔出非当前显示类型的设备线 (例如当前输出 hdmi, 拔出 cvbs 线)	不做任何处理
只插着 cvbs 或无插着任何设备	插入 hdmi	输出 hdmi, 输出显示类型为数据库优先, 其次使用默认
只插着 hdmi	插入 cvbs	不做任何处理
无插着任何设备	插入 cvbs	输出 cvbs, 输出显示类型为数据库优先, 其次使用默认
同时插着 hdmi 和 cvbs	手动切换 (setting 或遥控) 到设备支持的显示类型,	弹出10s 确认提示。确认后, 保存该显示类型到数据库; 否则返回切换前的显示类型。
同时插着 hdmi 和 cvbs	进入休眠, 再唤醒	唤醒后, 输出 HDMI-720P (HDMI 优先级高)
只插着 hdmi, 系统输出 hdmi	进入休眠, 再唤醒	唤醒后, 恢复休眠前的显示类型
只插着 cvbs, 系统输出 cvbs	进入休眠, 再唤醒	唤醒后, 恢复休眠前的显示类型

4.2.显示接口

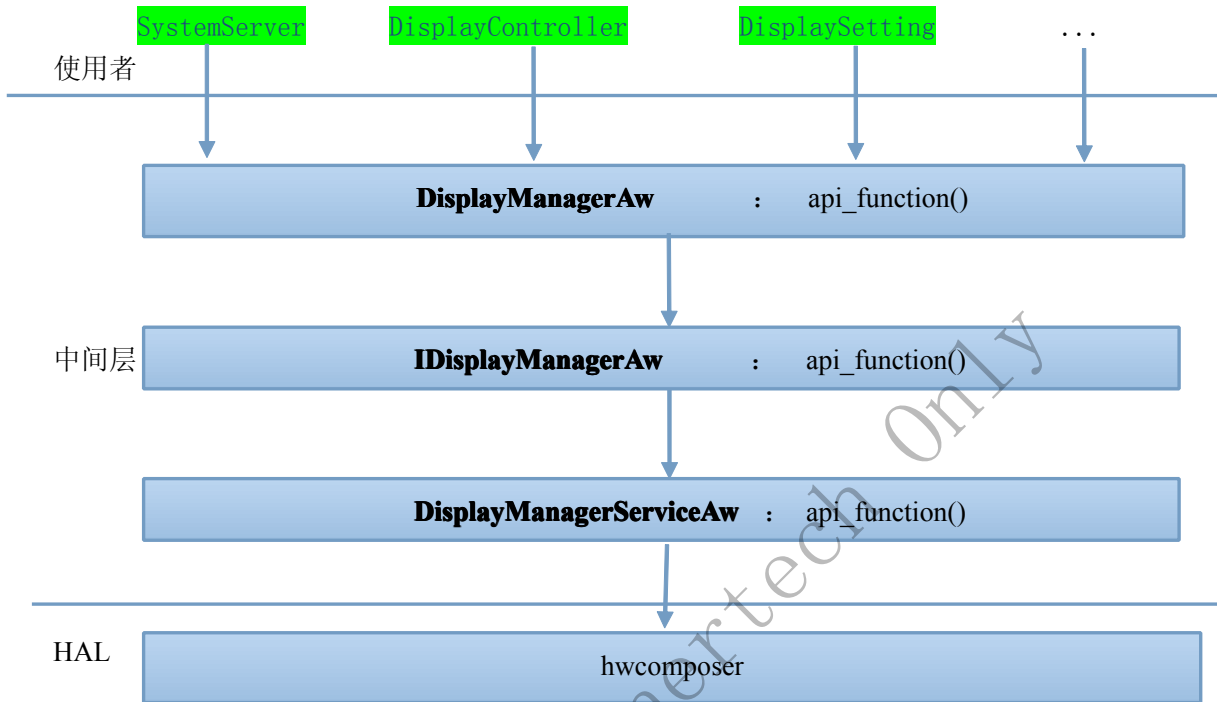


图 2 android 的显示接口调用示意图

接口说明：

1. **public int getDisplayOutputType(int mDisplay);**

➤ **ARGUMENTS**

mDisplay 显示设备序号

➤ **RETURNS**

返回 当前输出的显示类型

➤ **DESCRIPTION**

获取当前输出的显示类型。

2. **public int getDisplayWidth(int mDisplay);**

➤ **ARGUMENTS**

mDisplay 显示设备序号

➤ **RETURNS**

返回 当前显示的宽

➤ **DESCRIPTION**

获取当前显示的宽。



3. **public int getDisplayHeight(int mDisplay);**

➤ **ARGUMENTS**

mDisplay 显示设备序号

➤ **RETURNS**

返回 当前显示的高

➤ **DESCRIPTION**

获取当前显示的高。

4. **public int setDisplayOutputType(int mDisplay,int type,int format);**

➤ **ARGUMENTS**

mDisplay 显示设备序号

type 显示类型

format 显示模式

➤ **RETURNS**

返回 0

➤ **DESCRIPTION**

设置显示设备的显示模式

5. **public int isSupportHdmiMode(int mode);**

➤ **ARGUMENTS**

mode hdmi 的显示模式，例如 720p

➤ **RETURNS**

返回 0: 不支持该显示模式； 非 0: 支持该显示模式

➤ **DESCRIPTION**

查询显示设备是否支持该 hdmi 模式。

6. **public int isSupport3DMode();**

➤ **ARGUMENTS**

无

➤ **RETURNS**

返回 0: 不支持该 3D 模式； 非 0: 支持该 3D 模式

➤ **DESCRIPTION**

查询显示设备是否支持 3D 模式。



7. **public int getHdmiHotPlugStatus();**

➤ **ARGUMENTS**

无

➤ **RETURNS**

返回 0: hdmi 热插拔拔出状态; 非 0: hdmi 热插拔插入状态

➤ **DESCRIPTION**

查询 hdmi 热插拔状态。

8. **public int getTvHotPlugStatus();**

➤ **ARGUMENTS**

无

➤ **RETURNS**

返回 0: cvbs 热插拔拔出状态; 非 0: cvbs 热插拔插入状态

➤ **DESCRIPTION**

查询 cvbs 热插拔状态。

9. **public int setDisplayAreaPercent(int displayno,int percent);**

➤ **ARGUMENTS**

displayno 显示通道序号

percent 缩放比例, 单位为%

➤ **RETURNS**

返回 0

➤ **DESCRIPTION**

设置全局的显示画面的缩放比例。

10. **public int getDisplayAreaPercent(int displayno);**

➤ **ARGUMENTS**

displayno 显示通道序号

➤ **RETURNS**

返回 当前的显示的缩放比例

➤ **DESCRIPTION**

获取全局的显示画面的缩放比例。



11. public int setDisplayBright(int displayno,int bright);

➤ **ARGUMENTS**

displayno	显示通道序号
bright	亮度值

➤ **RETURNS**

返回	0
----	---

➤ **DESCRIPTION**

设置显示画面的亮度值。

12. public int getDisplayBright(int displayno);

➤ **ARGUMENTS**

displayno	显示通道序号
-----------	--------

➤ **RETURNS**

返回	亮度值
----	-----

➤ **DESCRIPTION**

获取当前显示画面的亮度值。

13. public int setDisplayContrast(int displayno,int contrast);

displayno	显示通道序号
contrast	对比度值

➤ **RETURNS**

返回	0
----	---

➤ **DESCRIPTION**

设置显示画面的对比度值。

14. public int getDisplayContrast(int displayno);

displayno	显示通道序号
-----------	--------

➤ **RETURNS**

返回	对比度
----	-----

➤ **DESCRIPTION**

获取当前显示画面的对比度值。

15. public int setDisplaySaturation(int displayno,int saturation);

displayno	显示通道序号
saturation	饱和度值

➤ **RETURNS**

返回	0
----	---

➤ **DESCRIPTION**

设置显示画面的饱和度值。



16. public int getDisplaySaturation(int displayno);

displayno 显示通道序号

➤ **RETURNS**

返回 饱和度

➤ **DESCRIPTION**

获取当前显示画面的饱和度值。

17. public int setDisplayHue(int displayno,int hue);

displayno 显示通道序号

hue 色度值

➤ **RETURNS**

返回 0

➤ **DESCRIPTION**

设置显示画面的色度值。

18. public int getDisplayHue(int displayno);

displayno 显示通道序号

➤ **RETURNS**

返回 色度

➤ **DESCRIPTION**

获取当前显示画面的色度值。

19. public int set3DLayerOffset(int displayno, int left, int right);

➤ **ARGUMENTS**

displayno 显示通道序号

left 图层的向左偏移量

Right 图层的向右偏移量

➤ **RETURNS**

返回 0

➤ **DESCRIPTION**

把 2D 的 ui 图形设置为 3D 效果显示



20. public int saveDisplayOutputTypeAndFormat(int mDisplay, int type, int format);

➤ **ARGUMENTS**

mDisplay	显示设备序号
type	显示输出类型
format	显示输出模式

➤ **RETURNS**

返回 0

➤ **DESCRIPTION**

将显示输出类型和模式保存到显示配置文件。

For Allwinnertech Only

5. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

For Allwinnertech Onl