



CV184x 开机画面使用手册

Version: 1.2.3

Release date: 2023-07-28

©2022 北京晶视智能科技有限公司
本文件所含信息归北京晶视智能科技有限公司所有。
未经授权，严禁全部或部分复制或披露该等信息。

目录

1	声明	2
2	开机画面使用指南	3
3	uboot	4
3.1	uboot 命令	4
3.2	uboot 函数相关代码	5
3.3	uboot 命令范例	6
3.4	使用储存装置并启用开机画面	6
3.4.1	准备开机图片	6
3.5	注意事项	7
4	alios	9
4.1	启用 alios 开机画面	9
4.1.1	准备开机图片	10
4.2	alios 开机画面原理	11

修订记录

Revision	Date	Description
1.0.0	2025/03/30	初稿

1 声明



法律声明

本数据手册包含北京晶视智能科技有限公司（下称“晶视智能”）的保密信息。未经授权，禁止使用或披露本数据手册中包含的信息。如您未经授权披露全部或部分保密信息，导致晶视智能遭受任何损失或损害，您应对因之产生的损失/损害承担责任。

本文件内信息如有更改，恕不另行通知。晶视智能不对使用或依赖本文件所含信息承担任何责任。本数据手册和本文件所含的所有信息均按“原样”提供，无任何明示、暗示、法定或其他形式的保证。晶视智能特别声明未做任何适销性、非侵权性和特定用途适用性的默示保证，亦对本数据手册所使用、包含或提供的任何第三方的软件不提供任何保证；用户同意仅向该第三方寻求与此相关的任何保证索赔。此外，晶视智能亦不对任何其根据用户规格或符合特定标准或公开讨论而制作的可交付成果承担责任。

联系我们

地址 北京市海淀区丰豪东路 9 号院中关村集成电路设计园（ICPARK）1 号楼

深圳市宝安区福海街道展城社区会展湾云岸广场 T10 栋

电话 +86-10-57590723 +86-10-57590724

邮编 100094（北京）518100（深圳）

官方网站 <https://www.sophgo.com/>

技术论坛 <https://developer.sophgo.com/forum/index.html>

2 开机画面使用指南

此指南用以说明如何在 uboot 及 alios 下，显示出开机画面。

3 uboot

uboot 提供以下功能：

- 提供 boot 环境下 VO 设备的开关，包含 VO 不同接口和时序。
- 提供 boot 环境下 VL 视频层的开关。
- 提供 boot 环境下 VO 设备背景色的设置。
- VL 视频层默认格式为 YUV420 PLANAR。

3.1 uboot 命令

- **startvo**：启动 VO 设备 参数：设备号，接口型别，时序。

```
cv1835# help startvo
startvo - open vo device with a certain interface.
```

- <dev> 设备号，请参考图表 1-1
- <intf-type> 接口型别，请参考图表 1-1
- <timing> 时序

<> MIPI_TX、LVDS、I80 不参考时序变量，会根据目前对应 driver 来设置时序
CV184X 上的标准时序如下：

```
2(1080P24), 3(1080P25), 4(1080P30), 5(720P50), 6(720P60),
7(1080P50), 8(1080P60), 9(576P50), 10(480P60), 11(800x600)
```

- **stopvo**：关闭 VO 设备 参数：设备号

```
cv1835# help stopvo
stopvo - close interface of vo device.
```

- <dev> 设备号，请参考图表 1-1

- **startvl**：启动 VL 视频层 参数：视频层号，图文件地址，视频地址，图文件大小，VO 对齐

```
cv1835# help startvl
startvl - open video layer of the vo
```

- <layer> 视频层号，请参考图表 1-1

- <addr_in> 图文件地址
- <addr_out> 视频地址
- <size> 图档大小
- <alignment> VO 对齐

· **stopvl**: 关闭 VL 视频层 参数: 视频层号

```
cv1835# help stopvl
stopvl - close video layer of the vo
```

- <layer> 视频层号, 请参考图表 1-1

· **setvobg**: 设定 VO 设备背景色 参数: 设备号, 背景色

- <dev> 设备号, 请参考图表 1-1
- <bgcolor> 背景色 (10bit RGB 排列, bit[29:20] 为 R, bit[19:10] 为 G, bit[9:0] 为 B)

图表 1-1

处理器类型	设备	视频层	图形层	接口类别
CV184X	[0]	[0]	[0]	64(BT.1120), 1024(LCD_18BIT), 2048(LCD_24BIT), 4096(LCD_30BIT), 8192(MIPI_TX), 65536(I80)

图表 1-2

处理器类型	视频层最大分辨率	图形层最大分辨率
CV184X	1920x1080	1920x1080

3.2 uboot 函数相关代码

```
cmd/Makefile
cmd/cvi_vo.c
drivers/video/Makefile
drivers/video/cvitek/ (包含以下子目录)
include/cvitek/cvi_disp.h
include/cvitek/cvi_mipi.h
include/cvitek/cvi_lvds.h
include/cvitek/cvi_i80.h
include/cvitek/cvi_panels/ (包含以下子目录)
```

3.3 uboot 命令范例

以下以 CV184X 处理器操作，配置设备 DHD 的时序 MIPI_TX 720*1080@60 输出为例。

<> 各 DDR 放置图片地址不同，请根据处理器来使用 DDR 地址。

- 把 JPEG 档载入到内存

```
fatload mmc 1:1 0x84080000 logo.jpg
```

- 解码 JPEG 到内存 (cvi_jpeg jpg_buf_addr dest_buf_addr jpg_size)

```
cvi_jpeg 0x84080000 0x82080000 0x80000
```

- DHD0 设备启动

```
startvo 0 8192 0 (MIPI_TX)
startvo 0 1024 0 (单路6bit LVDS)
startvo 0 2048 0 (单路8bit LVDS)
startvo 0 4096 0 (单路10bit LVDS, 暂不支持)
startvo 0 65536 0 (I80)
```

- VL 视频层启动

```
startvl 0 0x84080000 0x82080000 0x80000 16
```

- 设置 VO 背景色为黑色

```
setvobg 0 0x00000000
```

- VL 视频层关闭

```
Stopvl 0
```

- DHD0 设备关闭

```
Stopvo 0
```

3.4 使用储存装置并启用开机画面

3.4.1 准备开机图片

SDK 提供了 build/tools/common/bootlogo/process_images.py 工具，可将原始图片转换为 SDK 所需的 logo.jpg 格式 (MJPEG 裸流)。

安装依赖：

```
apt install ffmpeg python3-pil
```


使用示例:

单张图片 (uboot 仅支持单张), 竖屏 720x1280 旋转 90 度:

```
python build/tools/common/bootlogo/process_images.py \  
-i my_logo.png -o bootlogo/ -r 720 1280 --rotate 90
```

单张图片, 不旋转, 使用原图分辨率:

```
python build/tools/common/bootlogo/process_images.py \  
-i my_logo.png -o bootlogo/
```

参数说明:

- -i 输入图片路径 (支持 jpg/png)
- -o 输出目录 (默认当前目录, 生成 logo.jpg)
- -r 输出分辨率 WIDTH HEIGHT (不指定则以输入图片为准)
- --rotate 旋转角度: 0/90/180/270

生成 logo.jpg 后, 继续以下配置步骤。

1. 将开机图档 logo.jpg (I80 屏需要 BMP 格式图文件) 拷贝至板卡目录下 bootlogo/logo.jpg。
2. 在 build/boards/cv184x/cv184x_wevb/u-boot/cv184x_defconfig 配置 CONFIG_BOOTLOGO 为 y。
3. 如需使用 reset、power、backlight 功能, 需在 build/boards/default/dts/cv184x/cv184x_base.dtsi 中的 vo 节点添加 gpio 相关信息如:

```
reset-gpio = <&porte 2 GPIO_ACTIVE_LOW>;  
pwm-gpio = <&porte 0 GPIO_ACTIVE_HIGH>;  
power-ct-gpio = <&porte 1 GPIO_ACTIVE_HIGH>;
```

4. 编译时使用 menuconfig 在 SDK options 中使能 Pack BOOTLOGO, 该配置会将 logo.jpg 打包到烧录文件。
5. 使用下列命令编译 BSP。

```
source build/envsetup_soc.sh  
defconfig cv184x_wevb  
menuconfig  
Build_all
```

3.5 注意事项

- 配置开机画面, 通过 BT.1120/656 接口显示时, 外接处理器的驱动需自行移植实现。
- 如果开机画面使用的是 MIPI_TX、LVDS 或 I80 接口时, 若有不支持的 mipi_dsi、lvds 或者 i80 panel, 可参考 u-boot-2021.10/include/cvitek/cvi_panels 内的 headers, 新增相对应的 header。只要按 u-boot-2021.10/include/cvitek/cvi_panels/cvi_panels.h 参照其他 panel 修改即可对应不同的 mipi_dsi、lvds 或 i80 panel。

- 使用储存装置并储存开机画面时，需于 CV184x_asic.dtsi 配置一块内存空间（默认为 0x82080000），并确保 u-boot/include/configs/CV184x-asic.h 中的 LOGO_RESERVED_ADDR 设置为同样内存空间。

4 alios

- alios 开机画面目前仅支持 MIPI_DSI 接口，提供同 linux 一致的 mipi_tx_xx 接口 (可参考 Screen_Docking_Guide.pdf 的 MIPI_DSI 章节)。用户通过在 solution 中调用这些接口实现 VO 设备的初始化。

4.1 启用 alios 开机画面

1. 在 cvi_alios/components/cvi_mmf_sdk/cvi_middleware/cvi_panel 下添加 panel 的 header，可参考 Screen_Docking_Guide.pdf 或已支持 panel 实现 combo_dev_cfg_s 等数据结构。
2. 在 cvi_alios/components/cvi_mmf_sdk/cvi_middleware/cvi_panel/dsi_panels.h 中实现新 panel 的 panel_desc_s 结构体。
3. 在 cvi_alios/solutions/normboot/package_yamls/package.yaml.turnkey 中添加 config 选项并开启如：

```
CONFIG_PANEL_HX8394: 1 //可添加其他屏幕
CONFIG_RTOS_INIT_MEDIA: 1
CONFIG_SUPPORT_VO: 1
CONFIG_ALIOSLOGO: 1
```

4. 在 cvi_alios/solutions/normboot/customization/cv1842hp_gc8613/pipeline/custom_moduleparam.c 中修改配置如：

```
.alios_sys_mode = 0,
.alios_vi_mode = 0,
.alios_vpss_mode = 0,
.alios_venc_mode = 0,
.alios_vo_mode = 1,
```

5. 在 cvi_alios/solutions/normboot/customization/cv1842hp_gc8613/pipeline/custom_voparam.c 中修改配置如：

```
.u8Bindmode = false,

//可添加其他屏幕
#if (CONFIG_PANEL_HX8394 == 1 || CONFIG_PANEL_ILI9488 == 1 ||
    CONFIG_PANEL_OT7290B == 1)
    .u8VoCnt = 1,
```

(下页继续)

(续上页)

```
#else
    .u8VoCnt = 0,
#endif
```

6. 如需使用 reset、power、backlight 功能,需在 cv1_alios/solutions/normboot/customization/cv1842hp_gc861 中 PLATFORM_PanelInit() 接口自行添加 gpio 相关信息如:

```
#if (!defined(CONFIG_SUPPORT_VO) || (CONFIG_SUPPORT_VO))
#if CONFIG_PANEL_HX8394
    u8 pw_port, pw_pin, bl_port, bl_pin, rst_port, rst_pin;
    pw_port = 4;
    pw_pin = 1;
    bl_port = 4;
    bl_pin = 0;
    rst_port = 4;
    rst_pin = 2;
    _GPIOSetValue(pw_port, pw_pin, 1);
    _GPIOSetValue(bl_port, bl_pin, 1);
    _GPIOSetValue(rst_port, rst_pin, 1);
    udelay(20 * 1000);
    _GPIOSetValue(rst_port, rst_pin, 0);
    udelay(20 * 1000);
    _GPIOSetValue(rst_port, rst_pin, 1);
    udelay(20 * 1000);
```

4.1.1 准备开机图片

SDK 提供了 build/tools/common/bootlogo/process_images.py 工具来生成 logo.jpg。

安装依赖:

```
apt install ffmpeg python3-pil
```

使用示例:

单张图片:

```
python build/tools/common/bootlogo/process_images.py \
-i my_logo.png -o bootlogo/ -r 720 1280 --rotate 90
```

多张图片 (alios 支持动画 logo, 轮流播放多张 JPEG):

```
python build/tools/common/bootlogo/process_images.py \
-i ./frames/ -o bootlogo/ -r 720 1280 --rotate 90 -f 15
```

参数说明:

- -i 输入图片或文件夹路径
- -o 输出目录
- -r 输出分辨率 WIDTH HEIGHT
- --rotate 旋转角度: 0/90/180/270

- -f 帧率 (fps), 仅多张图片模式有效, 默认 10

重要: 必须设置 “CONFIG_RTOS_LOGO_SIZE “, 该值需大于实际 logo.jpg 文件的大小, 且该内存不会被释放。

生成 logo.jpg 后, 继续以下配置步骤。

7. 将开机图档 logo.jpg 拷贝至板卡目录下 bootlogo/logo.jpg, 编译 SDK, 编译时使用 menuconfig 在 SDK options 中使能 Pack BOOTLOGO, 指令如下:

```
source build/envsetup_soc.sh
defconfig cv184x_wevb
menuconfig
Build_all
```

4.2 alios 开机画面原理

1. 将开机图档 logo.jpg 拷贝至 \$BOOTLOGO_PATH, 编译 SDK。
2. 在 `cvi_alios/components/cvi_platform/media/src/media_logo.c` 代码中, 构建 `VDEC_STREAM_S` 结构体, 并指定 `pu8Buf` 等于宏 `CVIMMAP_RTOS_LOGO_ADDR`, 该地址保存着 logo.jpg 数据。
3. 调用 `CVI_VB_Init()` 初始化 VB。
4. 初始化 VDEC 设备 (可参考 `MediaProcessingSoftwareDevelopmentReference_zh.pdf` 第八章)。
5. 调用 `CVI_VDEC_SendStream()` 将 jpeg 数据送 VDEC 解码。
6. 调用 `CVI_VDEC_GetFrame()` 获取解码完成的 `VIDEO_FRAME_INFO_S` 结构体。
7. 调用 `CVI_VO_SendLogoFromIon()` 送显, 该接口目前仅支持 NV21 格式。