



旷明智能

旷明 X0S 显示配置指导

文档编号:		版本号:	V1.0.0
日期:	2025/11/20	负责人:	多媒体 Team
文档类型:	接口文档		
版权声明:	本文件系旷明智能科技（“QuaMing”）的原创成果，旷明智能科技依法享有该文件的全部版权。对本文件的全部或部分复制，务必事先取得旷明智能科技的书面许可，并向版权所有者予以明确确认。任何侵犯本公司版权等知识产权的行为，本公司均保留依法追究其法律责任的权利。 在法律许可的范畴内，特此声明，使用前，请审慎阅读合同条款与条件以及相关说明，并严格遵循本文件中的各项说明。旷明智能科技对于不当行为所导致的后果（包括但不限于电压过高、超频或温度过高），不承担任何责任。 旷明智能科技所提供的信息仅作参考或典型应用之用，本文件中的所有声明、信息及建议，均不构成任何明示或暗示的担保。旷明智能科技保留随时变更电路设计和/或规格的权利，且无需另行通知。 客户应全权负责获取实施解决方案/产品可能所需的第三方许可，旷明智能科技不承担任何与第三方许可相关的许可费用或特许权使用费。对于任何由所需第三方许可证所涵盖的事项，旷明智能科技不承担任何保证、赔偿或其他义务。 凡以任何方式直接或间接使用本文件资料者，均视为自愿接受本文件声明的约束。		

修订记录

版本	更新人	修订日期	修订内容
V1.0.0	多媒体Team	2025/11/20	初始版本
V1.0.1	多媒体Team	2025/12/15	更新旋转设置
V1.0.2	多媒体Team	2025/12/16	更新开机动画片源定制

目录

1 概述.....	2
1.1 背景.....	2
1.2 目的.....	2
2 双屏异显配置.....	3
2.1 双屏设置.....	3
2.2 亮度设置.....	3
2.3 图片设置.....	3
2.4 视频播放.....	4
3 开机动画配置.....	5
3.1 配置编译开关.....	5
3.2 配置启动参数.....	5
3.3 转换片源.....	5
3.4 安装片源.....	5
4 旋屏配置.....	7
4.1 QM10XD 芯片平台	7
4.1.1 配置编译开关.....	7
4.1.2 配置开机 logo 旋转.....	7
4.1.3 配置开机动画旋转.....	7
4.1.4 配置应用 UI 界面旋转	7
4.1.5 配置旋转内存.....	8
4.2 QM10XH 芯片平台	8
4.3 QM10XV 芯片平台	8
5 屏幕分辨率更改.....	9
5.1 QM10XD 芯片平台	9
5.1.1 更改编译开关.....	9
5.1.2 更改开机动画启动参数.....	9
5.1.3 更改显存和压缩内存.....	9
5.2 QM10XH 芯片平台	9
5.3 QM10XV 芯片平台	9
6 屏幕类型更改.....	10
6.1 QM10XD 芯片平台	10
6.2 QM10XH 芯片平台	10
6.3 QM10XV 芯片平台	10
6.3.1 更改编译开关.....	10
6.3.2 更改显示设备类型.....	10
6.3.3 加载显示驱动模块.....	10

1 概述

1.1 背景

旷明 XOS（全称 X-AIOS）是旷明智能科技开发提供的通用 AI 操作系统，运行于富瀚和旷明的芯片平台，方便客户基于此进行产品应用开发。

1.2 目的

本文档旨在指导旷明 XOS 里显示相关的配置，包括双屏异显和开机动画等，帮助开发者快速集成显示相关功能。

2 双屏异显配置

QM10XD 芯片支持 MIPI (2x2lane) 接口的双屏异显，以下以此芯片平台为例做介绍。

2.1 双屏设置

默认发布的 SDK 版本是单屏的，可以按如下方法设置成双屏：

- 在 product/demo_ld/project_demo_ld_defconfig 文件中，打开双屏编译开关
CONFIG_XOS_LCM_2ND_SCR=y

- 在 product/demo_ld/config/display.ini 文件中增加第二个屏的配置 display1:
[id]

display0 = 0,0,0

display1 = 0,1,4

这里 display0 代表屏幕 0，display1 代表屏幕 1，等号后面三个数字分别代表屏幕类型（非 SPI 屏为 0，SPI 屏为 1）、vo 显示设备号、fb 节点号（/dev/fb0 就填 0，/dev/fb4 就填 4）。

注：MIPI 屏、RGB 屏、MCU 屏统称为非 SPI 屏。

- 在 product/demo_ld/qua_overlay/ko/loadko.sh 文件中，为第二个屏幕配置显存和压缩内存，以 KB 为单位：

insmod fhfb.ko

video="vram0_size:4000,vram1_size:0,vram2_size:0,vram3_size:0,vram4_size:4000,vram5_size:0,vram6_size:0,vram7_size:0," fbc=3000 fbc1=3000 fhfb0_start=0 fhfb0_size=0 fhfb0_fbc=1
fhfb4_start=0 fhfb4_size=0 fhfb4_fbc=1

这里显存 vram4_size 的计算是：800x1280 的屏幕分辨率，格式为 ARGB，需要的显存大小是 800*1280*4/1024=4000KB。

压缩内存 fbc1 的大小一般配置为 vram4_size 大小的四分之三，如果配得太小，会导致复杂纹理的图形压缩时出现异常；如果配得太大，造成内存空间浪费。

2.2 亮度设置

亮度设置接口在 core/package/libnative/qlibquadriv/include/qm_backlight.h 文件中声明：

int qm_backlight_setbri_index(uint32_t device_index, uint32_t brightness);

其中，device_index 是屏幕索引，从 0 开始。brightnesss 是要设置的亮度，范围从 0 到 100，值越大越亮。

示例代码：

设置第一个屏幕亮度到 50

qm_backlight_setbri_index(0, 50);

设置第二个屏幕亮度到 80

qm_backlight_setbri_index(1, 80);

2.3 图片设置

图片通过开源 LVGL 图形库的 `lv_image_set_src` 接口设置，之后经过 LVGL 框架下的图片解码和绘制流程之后写入平台的 `framebuffer`，详细用法参见官方网站（[Images（图象） — LVGL 文档](#)）。

图片显示在哪个屏幕上，取决于 `image` 控件所在的控件树根是哪个屏幕的。

调用 `lv_display_get_1st_scr` 接口可以获取第一个屏幕的 `lv_display_t` 对象指针，其 `act_scr` 成员变量即为第一个屏幕的控件树根；类似地，调用 `lv_display_get_2nd_scr` 接口可以获取第二个屏幕的 `lv_display_t` 对象指针，其 `act_scr` 成员变量即为第二个屏幕的控件树根。

示例代码：

```
extern lv_display_t * lv_display_get_1st_scr(void);
lv_display_t * _disp1 = lv_display_get_1st_scr();
if(_disp1 != NULL){
    lv_obj_t * 1st_img = lv_image_create(_disp1->act_scr);
    lv_image_set_src(1st_img, "H:/res/skin/wallpaper/wallpaper-bg-1.jpg");
}

extern lv_display_t * lv_display_get_2nd_scr(void);
lv_display_t * _disp2 = lv_display_get_2nd_scr();
if(_disp2 != NULL){
    lv_obj_t * 2nd_img = lv_image_create(_disp2->act_scr);
    lv_image_set_src(2nd_img, "H:/res/skin/wallpaper/wallpaper-bg-2.jpg");
}
```

2.4 视频播放

视频通过播放器接口播放，详细用法参见《QuaPlayer 接口使用说明》。

通过 `KEY_PARAMETER_VO_DISPLAY_ID` 参数来指定视频在哪个屏幕上播放。

示例代码：

在第一个屏幕上播放视频

```
qua_mm_player_set_parameter(player, KEY_PARAMETER_VO_DISPLAY_ID, (QUA_VOID_PTR)"id:display0");
```

在第二个屏幕上播放视频

```
qua_mm_player_set_parameter(player, KEY_PARAMETER_VO_DISPLAY_ID, (QUA_VOID_PTR)"id:display1");
```

注：这里的“`id:display0`”和“`id:display1`”其实就是对应于 2.1 节提到的 `display.ini` 文件中的配置。

3 开机动画配置

本章仍以 QM10XD 芯片平台为例做介绍。

3.1 配置编译开关

在 `product/demo_ld/project_demo_ld_defconfig` 文件中，打开开机动画编译开关
`CONFIG_XOS_PLAY_BOOTANIMATION=y`

3.2 配置启动参数

在 `product/demo_ld/rootfs_overlay/etc/init.d/S11bootanimation` 文件中，要根据产品的实际情况来配开机动画的启动参数

```
./bootanimation_server -w 800 -h 1280 -s 38 -r 60 -v '2,32768;2,2457600' &
```

其中 `-w` 和 `-h` 是开机动画片源的宽高，`-s` 是屏幕的物理分辨率，`-r` 是动画播放帧率，`-v` 是 `vb common pool` 的配置（逗号前表示一个 `pool` 包含几块 `block`，逗号后表示每块 `block` 的大小，以字节为单位。若是多个 `pool`，则用分号分隔）。

注：`-s` 后面的数字就是

`core/package/libnative/qlibquammapi/include/display/qua_mm_display.h` 文件中的枚举值，800x480、800x1280、1024x600、480x854 分辨率的屏幕分别对应于

```
QUA_VO_OUTPUT_800x480_60
```

```
QUA_VO_OUTPUT_800x1280_60
```

```
QUA_VO_OUTPUT_1024x600_60
```

```
QUA_VO_OUTPUT_480x854_60
```

3.3 转换片源

开机动画支持的格式是 h264 裸码流，并且参考帧只能有一个，不能含有 B 帧。

片源的分辨率也不能超出屏幕大小。

如果片源不满足如上条件，则需先在 PC 上用 `ffmpeg` 软件转换片源。

对于 `mp4` 文件的片源，可以用如下转换命令：

```
ffmpeg -i video_720p.mp4 -vf "scale=800:1280" -c:v libx264 -x264opts  
"ref=1:bframes=0" -preset medium -crf 23 800x1280.h264
```

其中，800:1280 是要缩放到的分辨率。

如果觉得原有片源播放时间太长，则可截断片源。比如，用如下命令可以只取前 30 帧：

```
ffmpeg -i 800x1280.h264 -frames:v 30 -c copy output.h264
```

3.4 安装片源

把转换后的片源放到 `core/res/` 对应的目录下。

然后在 `product/demo_ld/rootfs_overlay/overlay_for_precopy.sh` 文件中，拷贝转换后的片源到 `product/demo_ld/rootfs_overlay/usr/media` 目录下，并重命名为 `boot.h264`，见下：

```
install -vD
```

```
$PRJ_TOP_DIR/core/res/moto_ld/h264/800x1280.h264 ./usr/media/boot.h264
```

这样编译烧录后，板子上就存在一个/usr/media/boot.h264 文件，开机动画程序 bootanimation_server 就会打开该片源进行播放。

4 旋屏配置

4.1 QM10XD芯片平台

4.1.1 配置编译开关

在 `product/demo_ld/project_demo_ld_defconfig` 文件中，打开旋屏编译开关
`CONFIG_XOS_FB_ROTATION=y`

4.1.2 配置开机 logo 旋转

开机 logo 是在 uboot 里做的，本身并不会旋转。而是在 PC 上把图片旋转好，在 uboot 里显示这张旋转好的图片。

4.1.3 配置开机动画旋转

开机动画是根据环境变量 `landscape` 来决定是否旋转的。

在刷机脚本 `script.ini` 中，增加
`setenv landscape '1'`；

其中 `landscape` 是环境变量名称，后面的数字代表旋转的方向 (0、1、2、3 分别代表不旋转、顺时针 90 度旋转、顺时针 180 度旋转、顺时针 270 度旋转)。不设环境变量 `landscape` 等同于设为 0，即不旋转。

4.1.4 配置应用 UI 界面旋转

应用 UI 界面的旋转方向是由 `product/demo_ld/config/param.ini` 文件中的配置决定的：

```
show_hor = 1
```

`show_hor` 所赋值的含义与上一节提到的环境变量 `landscape` 的值相同。

注：屏幕的显示宽高应是实际物理分辨率，即在 `core/board/generic/qml0xd/fbdev_10xd.c` 文件中设给 `set_public_attr` 接口的 `intf_sync` 参数（不打开开机动画）或是启动脚本里的 `-s` 参数（打开开机动画时，参见 3.2 节）。

而应用 UI 界面旋转后的宽高应和屏幕的显示宽高保持一致，即在

`core/board/generic/qml0xd/conf/lv_conf.h` 文件中的 `LV_USE_HOR_SIZE` 和 `LV_USE_VER_SIZE` 宏定义。

但是应用 UI 界面布局的宽高则应调用 `lv_display_get_horizontal_resolution` 和 `lv_display_get_vertical_resolution` 接口获取。

例如：屏幕的显示宽高是 `QUA_VO_OUTPUT_480x854_60`，则 `LV_USE_HOR_SIZE` 和 `LV_USE_VER_SIZE` 应定义成 480 和 854。如果设置屏幕旋转方向为 90 度或 270 度，则

lv_display_get_horizontal_resolution 和 lv_display_get_vertical_resolution 接口会返回 854 和 480。这意味着应用 UI 界面应按横屏时的宽高布局。

4.1.5 配置旋转内存

要使 UI 界面旋转生效，需要增大显存：

在 product/demo_ld/qua_overlay/ko/loadko.sh 文件中，把 vram0_size 增大为原来的一倍：

```
insmod fhfb.ko
```

```
video="vram0_size:8000,vram1_size:0,vram2_size:0,vram3_size:0,vram4_size:0,vram5_size:0,vram6_size:0,vram7_size:0," fbc=3000 fbc1=0 fhfb0_start=0 fhfb0_size=0 fhfb0_fbc=1 fhfb4_start=0 fhfb4_size=0 fhfb4_fbc=1
```

这样，这块显存的前半部分存放旋转后的数据，用于送显；后半部分存放旋转前的数据，用于给图形库 LVGL 绘制。

要使开机动画旋转生效，还需为硬件配置旋转内存，以 KB 为单位：

```
insmod g2d.ko rotbuf=2000
```

这里旋转内存的大小配成增大前显存的一半。这是因为视频只支持 420SP 和 422SP 格式，所以只需 RGBA 格式的显存的一半。

注意：硬件旋转要求宽高 8 像素对齐。因此，对于 480x854 这样的屏幕，显存的大小计算就变成了 $480 \times 856 \times 4 \times 2 / 1024 = 3210$ ，其中 856 是 854 向上对齐 8 像素后的高度，乘以 2 即增大为原来的一倍。详细计算方法参见 2.1 节。

4.2 QM10XH芯片平台

暂不支持旋屏。

4.3 QM10XV芯片平台

如果是 SPI 屏，则驱动配置的屏幕宽高和应用 UI 配置的宽高颠倒时，会自动旋屏。

如果是非 SPI 屏，则不支持旋屏。

5 屏幕分辨率更改

5.1 QM10XD芯片平台

5.1.1 更改编译开关

默认发布的 SDK 版本适配的屏幕分辨率是 800x1280。如果更换的屏幕不是这个分辨率，则需要修改在 `product/demo_ld/project_demo_ld_defconfig` 文件中，更改下面的宏开关：

```
CONFIG_XOS_LCM_800x1280=y  
CONFIG_XOS_RES_800x1280=y
```

其中，LCM 的开关在 `core/board/generic/qm10xd/fbdev_10xd.c` 和 `core/board/generic/qm10xd/fbdev2_10xd.c` 中被用到，决定了设备分辨率。而 RES 的开关在 `core/board/generic/qm10xd/conf/lv_conf.h` 中被用到，决定了应用 UI 绘制界面的分辨率。一般情况下，这两个分辨率应该是一致的。

5.1.2 更改开机动画启动参数

屏幕分辨率变了，就要改变开机动画启动参数 `-s`。具体参见 3.2 节。

5.1.3 更改显存和压缩内存

屏幕分辨率变了，需要重配显存和压缩内存。

在 `product/demo_ld/qua_overlay/ko/loadko.sh` 文件中，更改 `vram0_size` 和 `fb` 后面的数字：

```
insmod fhfb.ko  
video="vram0_size:4000,vram1_size:0,vram2_size:0,vram3_size:0,vram4_size:0,vram5_size:0,vram6_size:0,vram7_size:0," fbc=3000 fbc1=0 fhfb0_start=0 fhfb0_size=0  
fhfb0_fbc=1 fhfb4_start=0 fhfb4_size=0 fhfb4_fbc=0
```

内存大小的计算方法参见 2.1 节。

5.2 QM10XH芯片平台

可参照 5.1 节修改。

5.3 QM10XV芯片平台

如果是 SPI 屏，则仅需参照 5.1.1 和 5.1.2 节修改。

如果是非 SPI 屏，则还需参照 5.1.3 节修改。

6 屏幕类型更改

6.1 QM10XD芯片平台

默认发布的 SDK 版本适配的是 MIPI 屏。它与 RGB 屏、MCU 屏这些非 SPI 屏之间切换时，无需特别的修改。

暂不支持 SPI 屏。

6.2 QM10XH芯片平台

默认发布的 SDK 版本适配的是 MIPI 屏。

暂不支持 SPI 屏。

6.3 QM10XV芯片平台

默认发布的 SDK 版本适配的是 SPI 屏，若更换成 RGB/MCU 等非 SPI 屏时，需做如下修改。

6.3.1 更改编译开关

FB 驱动是双缓存机制，所以需要在产品的 defconfig 配置文件中打开如下开关：

CONFIG_XOS_USE_DOUBLE_BUFFERED=y

6.3.2 更改显示设备类型

在 display.ini 文件中，把屏幕类型从 SPI 屏改成非 SPI 屏，即把下面第一个数字从 1 改成 0。具体参见 2.1 节。

display0 = 1,0,0

6.3.3 加载显示驱动模块

在 pre_load_modules_QM10XV.sh 文件中，添加以下内容，依次加载所需的显示驱动模块：

insmod lcm_module.ko

insmod lcm.ko

insmod vou.ko

insmod fyfb.ko

ABOUT QUAMING

The majority of the R&D team of the company come from well-known semiconductor companies, such as UNISOC, ASR Microelectronics, and JLQ Technology. The team masters the core technologies of chip design, such as the AI algorithm library, the self-developed AI operating system, the high-performance multimedia library, and the multimedia chip IP. It possesses leading chips and is equipped with its self-developed AI operating system XOS, meeting the demands of intelligence, display and vision. It has released multiple chip solutions, featuring intelligence and low cost.

CONTACT US

For more product info, please contact z@quaming.com, or scan the QR code to follow us on Wechat.

This brief is for reference only and has no commitment. All content Contained herein is subject to changes without notice ©2023 Quaming Intelligent Technology Co., Ltd.

