



旷明 XOS QM102D LCD 驱动接口

使用说明

部 门	
文 档 编 号	
版 本 号	V0.1.0
作 者	

版权所有

旷明智能科技（无锡）有限公司

本资料及其包含的所有内容为旷明智能科技（无锡）有限公司所有,受中国法律及适用之国际公约中有关著作权法律的保护。未经旷明智能科技（无锡）有限公司书面授权,任何人不得以任何形式复制、传播、散布、改动或以其它方式使用本资料的部分或全部内容,违者将被依法追究责任。

更新记录

日期	更新人	版本	备注
2024/12/14		V0.0.1	初版
2025/09/05		V0.1.0	添加已调试的屏信息

目录

1、 引言	4
1.1 编写目的	4
1.2 预期读者和阅读建议	4
1.3 缩略术语	4
1.4 参考资料	4
2、 文档简介	5
2.1 功能简介	5
2.2 驱动框架	5
2.3 相关文件	6
2.4 已调试的屏信息	7
3、 接口函数	8
3.1 QUA_LCM_MIPI_JD9365D_POWERON	8
3.2 QUA_LCM_MIPI_JD9365D_IDENTIFY	8
3.3 QUA_LCM_MIPI_JD9365D_POWEROFF	8
3.4 QUA_LCM_MIPI_JD9365D_DEINIT	8
3.5 QUA_LCM_MIPI_JD9365D_INIT	8
3.6 QUA_LCM_MIPI_JD9365D_SUSPEND	9
3.7 QUA_LCM_MIPI_JD9365D_RESUME	9
3.8 QUA_LCM_MIPI_JLM101B021_CTRL	9
4、 全局变量和常量	10
常量	10
数组	10
5、 注意事项	11

1、引言

1.1 编写目的

本文旨在让客户快速了解旷明的 XOS QM102D SDK 产品 LCD 驱动接口方案，并快速上手基于旷明的 XOS QM102D SDK 做 LCD 服务和驱动开发。

1.2 预期读者和阅读建议

本文档可提供给客户、研发人员、技术支持工程师和测试工程师使用。

1.3 缩略术语

词语	解释
SDK	Software Development Kit
XOS	旷明统一操作系统
BSP	板级支持包
QuaMM	旷明多媒体

1.4 参考资料

2、 文档简介

本文着重描述 LCD 驱动接口和驱动结构，涉及 PWM 和 GPIO 相关内容可参考对应模块驱动接口文档。

本驱动接口用于控制 XOS 平台支持的 LCD 显示屏，实现了电源管理、初始化、显示同步等功能。

LCD 接口上支持 MIPI, RGB, SPI, MCU 等，详细的 LCM 接口具体依赖产品平台的芯片型号支持情况。

当前文档以 MIPI 接口 JD9365D 屏驱动为例进行介绍。

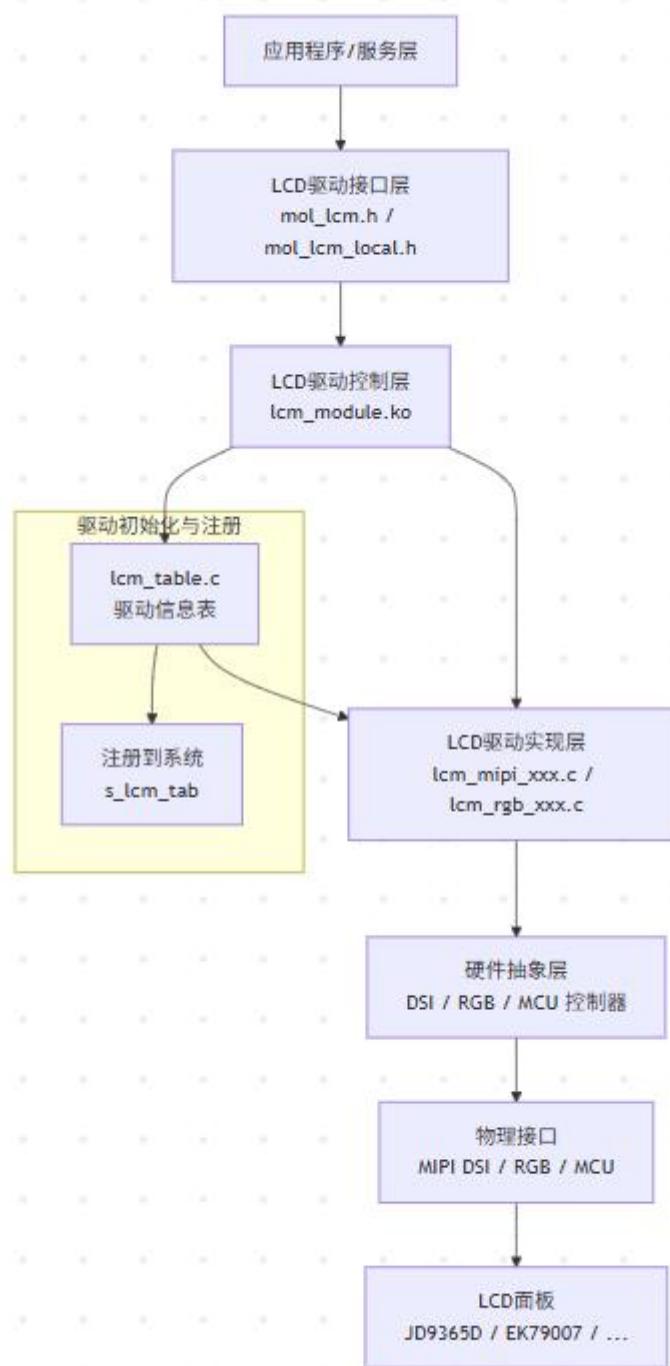
2.1 功能简介

LCD 驱动接口提供的功能主要包括：

- LCD 驱动注册
 - 添加 LCD 驱动文件
 - 添加 LCD 编译选项
 - 注册 LCD 模组信息

2.2 驱动框架

基于旷明 XOS 的 LCD 驱动框架图如下：



2.3 相关文件

base/soc/qm10xd/linux/media/driver/lcm_module:

`lcm_mipi_xxx.c`
`lcm_table.c`
`Makefile`
`lcm_module.ko`

2.4 已调试的屏信息

屏/驱动芯片	接口	屏信息	驱动文件
ST7265	RGB	5" 800(H) *480(V)	lcm_rgb_qua_st7265_tcxd050iblma.c
JD9365	MIPI	10.1" 800(H)*1280(V)	lcm_mipi_qua_jd9365d_jlm101b021.c
HX8729	MIPI	10.1" 1920(H)*1200(V)	lcm_mipi_qua_hx8729_qfh24004.c

3、 接口函数

3.1 qua_lcm_mipi_jd9365d_PowerOn

- **功能:** 开启 LCD 电源, 进行一些初始化操作, 包括设置 GPIO 引脚方向和电平, 初始化 MIPI 总线。
- **参数:**
 - `param`: 未使用, 可为空指针。
 - `ops`: 指向 `lcm_ops` 结构体的指针, 包含了操作 LCD 的函数指针。
- **返回值:** 0 表示成功, 非 0 表示失败 (当前实现中总是返回 0)。

3.2 qua_lcm_mipi_jd9365d_identify

- **功能:** 识别 LCD 设备 (当前实现中未进行实际识别操作, 直接返回 0)。
- **参数:**
 - `param`: 未使用, 可为空指针。
 - `ops`: 指向 `lcm_ops` 结构体的指针, 包含了操作 LCD 的函数指针。
- **返回值:** 0 表示成功识别或未进行识别操作, -1 表示识别失败 (当前实现中总是返回 0)。

3.3 qua_lcm_mipi_jd9365d_PowerOff

- **功能:** 关闭 LCD 电源, 包括设置 GPIO 引脚电平, 反初始化 MIPI 总线。
- **参数:**
 - `param`: 未使用, 可为空指针。
 - `ops`: 指向 `lcm_ops` 结构体的指针, 包含了操作 LCD 的函数指针。
- **返回值:** 0 表示成功, 非 0 表示失败 (当前实现中总是返回 0)。

3.4 qua_lcm_mipi_jd9365d_deinit

- **功能:** 反初始化 LCD, 调用 `qua_lcm_mipi_jd9365d_PowerOff` 关闭电源。
- **参数:**
 - `param`: 未使用, 可为空指针。
 - `ops`: 指向 `lcm_ops` 结构体的指针, 包含了操作 LCD 的函数指针。
- **返回值:** `qua_lcm_mipi_jd9365d_PowerOff` 的返回值, 0 表示成功, 非 0 表示失败。

3.5 qua_lcm_mipi_jd9365d_init

- **功能:** 初始化 LCD, 将初始化设置表推送到 LCD。

- **参数:**
 - **param:** 指向 `lcm_handle_t` 结构体的指针, 包含了 LCD 的相关信息。
 - **ops:** 指向 `lcm_ops` 结构体的指针, 包含了操作 LCD 的函数指针。
- **返回值:** 0 表示成功, 非 0 表示失败 (当前实现中返回推送设置表的结果)。

3.6 `qua_lcm_mipi_jd9365d_suspend`

- **功能:** 暂停 LCD, 反初始化 MIPI 总线。
- **参数:**
 - **param:** 未使用, 可为空指针。
 - **ops:** 指向 `lcm_ops` 结构体的指针, 包含了操作 LCD 的函数指针。
- **返回值:** 0 表示成功, 非 0 表示失败 (当前实现中总是返回 0)。

3.7 `qua_lcm_mipi_jd9365d_resume`

- **功能:** 恢复 LCD, 初始化 MIPI 总线 (当前未实现睡眠唤醒设置表的推送)。
- **参数:**
 - **param:** 未使用, 可为空指针。
 - **ops:** 指向 `lcm_ops` 结构体的指针, 包含了操作 LCD 的函数指针。
- **返回值:** 0 表示成功, 非 0 表示失败 (当前实现中总是返回 0)。

3.8 `qua_lcm_mipi_jlm101b021_ctrl`

- **功能:** `lcm_module_fun` 结构体, 包含了上述所有操作函数的指针, 用于统一管理 LCD 的各种操作。

4、全局变量和常量

常量

- `LCM_JD9365D_WIDTH`: LCD 的宽度, 值为 800。
- `LCM_JD9365D_HEIGHT`: LCD 的高度, 值为 1280。
- `LCM_JD9365D_DEBUG`: 调试开关, 值为 1 时开启调试输出。

数组

- `qua_lcm_mipi_jd9365d_jlm101b021_init_setting`: LCD 初始化设置表, 包含一系列的 DSI 命令和参数, 用于配置 LCD 的各种功能。
- `qua_lcm_mipi_jd9365d_dp_info`: `lcm_display_sync_info_t` 结构体, 定义了 LCD 的显示同步参数, 包括水平和垂直同步的有效时间、前后沿时间等。
- `qua_lcm_mipi_jd9365d_timing`: `dsi_timing_t` 结构体, 定义了 MIPI 总线的时序参数, 包括时钟和数据通道的各种时序设置。
- `qua_lcm_mipi_jd9365d_mipi_info`: `lcm_mipi_info` 结构体, 定义了 MIPI 接口的相关信息, 如工作模式、通道数、数据格式、数据包大小和时序等。
- `qua_lcm_mipi_jd9365d_jlm101b021_info`: `lcm_cfg_t` 结构体, 包含了 LCD 的基本配置信息, 如供应商名称、型号、分辨率、帧率、方向、显示同步信息、MIPI 信息和操作函数指针等。

5、 注意事项

- 部分功能（如 `lcm_mipi_jd9365d_identify` 中的设备识别操作）可能未完全实现或需要根据实际情况进行调整。
- 代码中的 `GPIO` 操作部分可能需要根据实际硬件连接进行修改。
- 在使用过程中，需确保正确调用各个接口函数，按照电源管理的流程进行操作，以避免损坏设备或出现异常。