

TDE

API 参考

文档版本 V1.5

发布日期 2025-12-04

概述

本文档主要介绍 TDE 的 API 和数据类型以及 Proc 调试信息。



说明

未经特殊说明，以 XM7606V13 指代 XM7606V1 系列所有芯片，以 XM7206V11A 指代 XM7206V1 系列所有芯片。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示如不可避免则将会导致死亡或严重伤害的具有高等级风险的危害。
 警告	表示如不可避免则可能导致死亡或严重伤害的具有中等级风险的危害。
 注意	表示如不可避免则可能导致轻微或中度伤害的具有低等级风险的危害。
须知	用于传递设备或环境安全警示信息。如不可避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
 说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2024-12-21	V1.0	DI 版本发布。
2025-03-04	V1.1	1.标准术语大小写统一。 2.文档语句通顺性整改。
2025-04-15	V1.2	3.2 xmedia_tde_blend_mode 纠正 sa、da 描述信息。 6.1 软件流程 更换软件实现流程图。
2025-06-04	V1.3	增加对 XM7206V11A 芯片的支持。
2025-08-20	V1.4	整个文档描述规范性更改。

修订日期	版本	修订说明
		2.2 xmedia_tde_add_task_single_blit 添加缩放，色域转换操作与 rop/blending 的互斥情况描述。
2025-12-04	V1.5	2.2 xmedia_tde_add_task_double_blit 添加背景位图与目标位图共 buffer 时的限制。

目 录

前 言.....	i
1 概述.....	1
1.1 概述	1
1.2 模块加载	1
1.2.1 加载的命令	1
1.2.2 参数	1
1.3 参考域说明	2
1.3.1 API 参考域.....	2
1.3.2 数据类型参考域.....	2
2 API 参考	4
2.1 API 概述.....	4
2.2 功能函数参考	5
3 数据类型	43
3.1 映射表.....	43
3.2 详细描述	44
4 错误码	65
5 Proc 调试信息.....	66
5.1 调试命令	66
5.2 调试信息	66
6 实例.....	73
6.1 软件流程	73
6.2 代码参考	74

插图目录

图 2-1 rop 运算的搬移操作示意图 (src1: R,G,B=0xFF,0xFF,0; src2: R,G,B=0,0,0xFF)	28
图 2-2 对前景位图进行 colorkey 运算的搬移操作示意图	29
图 2-3 对背景位图进行 colorkey 运算的搬移操作示意图	29
图 2-4 区域内 clip 示意图	30
图 2-5 区域外 clip 示意图	31
图 3-1 位图与位图中的操作区域的关系	48
图 3-2 画线线宽效果	50
图 5-1 XM7206V11A 的 proc 信息	66
图 5-2 XM7606V13 的 proc 信息	67
图 6-1 软件实现的流程图 (主流程)	74

表格目录

表 1-1 API 参考域说明	2
表 1-2 数据类型参考域说明	3
表 3-1 TDE 数据结构映射表	43
表 4-1 TDE API 错误码	65

1 概述

1.1 概述

TDE (Two Dimensional Engine) 利用硬件为 OSD (On Screen Display) 和 GUI (Graphics User Interface) 提供快速的图形绘制功能, 主要有快速位图搬移、快速色彩填充、快速搬移、快速位图缩放、画点、画水平/垂直线、位图格式转换、位图 alpha 叠加、位图按位布尔运算、ColorKey 操作。

1.2 模块加载

1.2.1 加载的命令

加载命令为 `insmod xm_tde.ko` 参数。

1.2.2 参数

参数 `g_max_tde_job`

该参数决定 TDE 内部可用 job 的最大个数。XM7606V13 默认不配置就是 128 个, XM7206V11A 默认不配置就是 20 个。使用 TDE 模块的每个功能都是承载在 job 上, 用户需要先获取一个空闲的 job, 添加一个或多个 task 到 job 上, 针对 job 提交 tde 硬件加速处理。用户可以根据负载情况通过模块参数设置 `g_max_tde_job`, 可配范围[20, 400]。

举例:

`insmod xm_tde.ko g_max_tde_job=200` // 按 200 个 job 分配

参数 g_max_tde_task

该参数决定 TDE 内部可用 task 的最大个数。XM7606V13 默认不配置就是 200 个，XM7206V11A 默认不配置就是 24 个。使用 TDE 模块对应的每一个功能，如画线，快速填充，快速拷贝，alpha blending，rop 运算等均对应一个 task，用户通过配置指定的 task，并添加到 job 上，提交给 tde 硬件进行加速处理。用户可以根据负载情况通过模块参数进行设置 g_max_tde_task，可配范围为[20, 800]。

1.3 参考域说明

1.3.1 API 参考域

本手册使用 9 个参考域描述 API 的相关信息，它们的作用如表 1-1 所示。

表1-1 API 参考域说明

参考域	含义
目的	简要描述 API 的主要功能。
语法	列出调用 API 应包括的头文件以及 API 的原型声明。
参数	列出 API 的参数、参数说明及参数属性。
描述	简要描述 API 的工作过程。
返回值	列出 API 所有可能的返回值及其含义。
需求	列出 API 包含的头文件和 API 编译时要链接的库文件。
注意	列出使用 API 时应注意的事项。
举例	列出使用 API 的实例。
相关接口	列出与本 API 相关联的其他接口。

1.3.2 数据类型参考域

本手册使用 5 个参考域描述数据类型的相关信息，它们的作用如表 1-2 所示。

表1-2 数据类型参考域说明

参考域	含义
说明	简要描述数据类型的主要功能。
定义	列出数据类型的定义语句。
成员	列出数据结构的成员及含义。
注意事项	列出使用数据类型时应注意的事项。
相关数据类型和接口	列出与本数据类型相关联的其他数据类型和接口。

2 API 参考

2.1 API 概述

TDE (Two Dimension Engine) 功能函数参考提供 2D 加速相关操作。

该功能模块提供以下 API:

- [xmedia_tde_init](#): 初始化 tde 模块。
- [xmedia_tde_exit](#): 去初始化 tde 模块。
- [xmedia_tde_create_job](#): 创建 1 个 TDE 工作组。
- [xmedia_tde_submit_job](#): 提交添加任务完成的 TDE 工作组。
- [xmedia_tde_add_task_draw_line](#): 添加画线任务。
- [xmedia_tde_add_task_quick_fill](#): 添加快速填充任务。
- [xmedia_tde_add_task_quick_copy](#): 添加快速拷贝任务。
- [xmedia_tde_add_task_scale](#): 添加缩放任务。
- [xmedia_tde_add_task_rotation](#): 添加旋转任务。
- [xmedia_tde_add_task_single_blit](#): 添加单源搬移任务(可带附加操作)
- [xmedia_tde_add_task_double_blit](#): 添加双源叠加搬移任务(可带附加操作)
- [xmedia_tde_set_alpha_threshold_value](#): 设置 alpha 判决阈值。
- [xmedia_tde_get_alpha_threshold_value](#): 获取 alpha 判决阈值。
- [xmedia_tde_cancel_job](#): 取消指定的 TDE 工作组。
- [xmedia_tde_wait_job](#): 等待指定的 TDE 工作组完成。

2.2 功能函数参考

xmedia_tde_init

【目的】

初始化 TDE 设备。

【语法】

```
xmedia_s32 xmedia_tde_init(xmedia_void);
```

【描述】

调用此接口初始化 TDE 设备。

【参数】

无。

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

接口返回值	含义
XMEDIA_SUCCESS	成功。
XMEDIA_ERRCODE_NOT_READY	TDE 驱动未加载。

【需求】

- 头文件：xmedia_tde.h
- 库文件：libxmedia_tde.a

【注意】

- 在进行 TDE 相关操作前应该首先调用此接口，保证 TDE 设备处于已初始化状态。
- TDE 设备允许支持多进程。

【举例】

```
/*declaration*/
xmedia_s32 ret = 0;

/* init TDE device */
ret = xmedia_tde_init();
if (ret != XMEDIA_SUCCESS) {
    return ret;
}
/* exit TDE device*/
xmedia_tde_exit();
```

xmedia_tde_exit

【目的】

去初始化 TDE 设备。

【语法】

```
xmedia_void xmedia_tde_exit(xmedia_void);
```

【描述】

调用此接口去初始化 TDE 设备。

【参数】

无。

【返回值】

无。

【错误码】

无。

【需求】

- 头文件：xmedia_tde.h
- 库文件：libxmedia_tde.a

【注意】

- 同一进程中，调用 `xmedia_tde_init` 与 `xmedia_tde_exit` 的次数需要对应。
- `xmedia_tde_exit` 会等待本进程提交任务完成后，退出，销毁本进程相关资源。

【举例】

无。

`xmedia_tde_create_job`

【目的】

创建 1 个 TDE 工作组。

【语法】

```
xmedia_s32 xmedia_tde_create_job(xmedia_s32 *handle);
```

【描述】

调用此接口创建 1 个 TDE 工作组（Job）。TDE 运行的最小单位是工作组，以工作组管理 TDE 任务：1 个 TDE 工作组可以管理一个或多个 TDE 任务；成功创建 TDE 工作组并添加完 TDE 任务后，通过 `xmedia_tde_submit_job` 提交该 Job；同一工作组中的 TDE 任务是按顺序执行(保序)。

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。一个句柄对应一个 Job。	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

接口返回值	含义
XMEDIA_SUCCESS	函数执行成功。

接口返回值	含义
XMEDIA_ERRCODE_NULL_PTR	空指针参数错误。
XMEDIA_ERRCODE_NOT_INIT	TDE 设备未初始化。
XMEDIA_ERRCODE_BUFFER_EMPTY	没有空闲的 job 资源。

【需求】

- 头文件: xmedia_tde.h
- 库文件: libxmedia_tde.a

【注意】

- 在调用此接口前应确保 TDE 设备已被 [xmedia_tde_init](#) 初始化过。
- [xmedia_tde_create_job](#) 必须和 [xmedia_tde_submit_job](#) 或 [xmedia_tde_cancel_job](#) 配套使用, 否则会造成 job 处于已创建状态资源无法回收。

【举例】

```

/* declaration */
xmedia_s32 ret;
xmedia_s32 handle;

ret = xmedia_tde_init();
if (ret != XMEDIA_SUCCESS) {
    return ret;
}
/* create a TDE job */
ret = xmedia_tde_create_job(&handle);
if (ret != XMEDIA_SUCCESS) {
    xmedia_tde_exit();
    return ret;
}
/* .....add task */
/* submit the job */
ret = xmedia_tde_submit_job(handle);
if (ret != XMEDIA_SUCCESS) {
    xmedia_tde_cancel_job(handle);
    return ret;
}

```

```
}
```

xmedia_tde_submit_job

【目的】

提交已创建的 TDE 工作组。

【语法】

```
xmedia_s32 xmedia_tde_submit_job(xmedia_s32 handle);
```

【描述】

调用此接口提交 1 个 job。此接口为异步非阻塞接口，接口返回成功不代表硬件处理完成，可配合 [xmedia_tde_wait_job](#) 进行同步。

【参数】

参数名称	描述	输入/输出
handle	TDE 任务句柄。一个句柄对应一个 job。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_SUCCESS	工作组提交 TDE 模块成功。
XMEDIA_ERRCODE_NOT_INIT	tde 模块未初始化。
XMEDIA_ERRCODE_INVALID_PARAM	参数不合法。
XMEDIA_ERRCODE_NOT_PERMITTED	操作不允许。

【需求】

- 头文件: xmedia_tde.h
- 库文件: libxmedia_tde.a

【注意】

- 在调用此接口前应保证调用 `xmedia_tde_init` 初始化 TDE 设备, 并且调用 `xmedia_tde_create_job` 获得了有效的任务句柄。
- 此接口为异步接口, 提交工作组给 TDE 成功返回, 不阻塞用户其他操作, 此时 TDE 会立即进行硬件处理也可能因负载进行排队延后处理, 可通过 `xmedia_tde_wait_job` 进行同步等待指定任务硬件处理完成。
- 提交工作组成功后, 可以调用 wait 接口进行同步。
- 如果 `xmedia_tde_submit_job` 返回非成功, 用户需要主动通过 `xmedia_tde_cancel_job` 进行资源释放。

【举例】

无。

xmedia_tde_add_task_draw_line

【目的】

向指定工作组中添加画线任务。

【语法】

```
xmedia_s32 xmedia_tde_add_task_draw_line(xmedia_s32 handle, const xmedia_tde_surface_info *dst, xmedia_tde_line_attr *line, xmedia_u32 line_num);
```

【描述】

向指定 handle 的工作组 job 中, 添加任务 task: 在目标 surface 的指定区域上画线。

【参数】

参数名称	描述	输入/输出
handle	由 <code>xmedia_tde_create_job</code> 接口获取的句柄, 一个 handle 对应一个工作组 job, 唯一对应。	输入
dst	目标位图和操作区域。	输入
line	画线属性指针。	输入

参数名称	描述	输入/输出
line_num	画线数量。数量[1, 100]。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_ERRCODE_NOT_INIT	模块未初始化。
XMEDIA_ERRCODE_NOT_PERMITTED	操作不允许。(状态机不对)
XMEDIA_ERRCODE_NULL_PTR	空指针参数错误。
XMEDIA_ERRCODE_INVALID_PARAM	无效参数错误。
XMEDIA_ERRCODE_BUFFER_EMPTY	没有空闲的 task 资源。
XMEDIA_ERRCODE_NOT_SUPPORT	不支持的操作。

【需求】

- 头文件：xmedia_tde.h
- 库文件：libxmedia_tde.a

【注意】

支持的格式如下：

目标位图格式	XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_RGB_888 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_888
--------	--

【举例】

无。

xmedia_tde_add_task_quick_fill

【目的】

向指定工作组中添加快速填充任务。

【语法】

```
xmedia_s32 xmedia_tde_add_task_quick_fill (xmedia_s32 handle, const xmedia_tde_surface_info
*dst, xmedia_u32 fill_color);
```

【描述】

向指定 handle 的工作组 job 中，添加任务 task：将数据值 fill_color 填充到目标 surface 指定的 rect 区域范围，可实现颜色填充的功能。

【参数】

参数名称	描述	输入/输出
handle	由 xmedia_tde_create_job 接口获取的句柄，一个 handle 对应一个工作组 job，唯一对应。	输入
dst	目标位图和操作区域。	输入

参数名称	描述	输入/输出
fill_color	填充颜色值，按 32bit 位配置。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_ERRCODE_NOT_INIT	模块未初始化。
XMEDIA_ERRCODE_NOT_PERMITTED	操作不允许。(状态机不对)
XMEDIA_ERRCODE_NULL_PTR	空指针参数错误。
XMEDIA_ERRCODE_INVALID_PARAM	无效参数错误。
XMEDIA_ERRCODE_BUFFER_EMPTY	没有空闲的 task 资源。
XMEDIA_ERRCODE_NOT_SUPPORT	不支持的操作。

【需求】

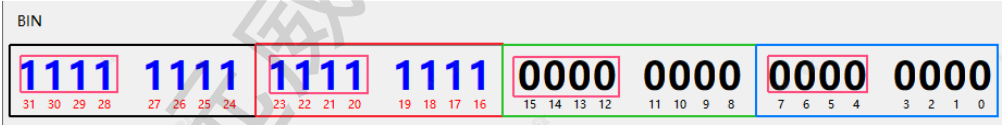
- 头文件：xmedia_tde.h
- 库文件：libxmedia_tde.a

【注意】

- 支持的格式如下：

目标位图格式	XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_RGB_888 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_888
--------	--

- fill_color 固定为 32bits。
- 当目标位图像素格式非 32 位位深时，取高位为有效位。如：红色填充值 0xffff0000，目标位图为 ARGB4444 格式的话，即每 8bits 分量取高 4 位，内部为 0xff00。(ARGB8888:0XFFFF0000→ARGB4444:0XFF00)



【举例】

无。

xmedia_tde_add_task_quick_copy

【目的】

向指定工作组中添加快速拷贝任务。

【语法】

```
xmedia_s32 xmedia_tde_add_task_quick_copy(xmedia_s32 handle, const
xmedia_tde_surface_info *src, const xmedia_tde_surface_info *dst);
```

【描述】

向指定 handle 的工作组 job 中，添加任务 task：将源 surface 中指定的区域 rect 的内容拷贝到目标 surface 中指定的区域 rect 的位置上。

【参数】

参数名称	描述	输入/输出
handle	由 xmedia_tde_create_job 接口获取的句柄，一个 handle 对应一个工作组 job，唯一对应。	输入
src	源位图和操作区域。	输入
dst	目标位图和操作区域。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_ERRCODE_NOT_INIT	模块未初始化。
XMEDIA_ERRCODE_NOT_PERMITTED	操作不允许。(状态机不对)
XMEDIA_ERRCODE_NULL_PTR	空指针参数错误。
XMEDIA_ERRCODE_INVALID_PARAM	无效参数错误。
XMEDIA_ERRCODE_BUFFER_EMPTY	没有空闲的 task 资源。
XMEDIA_ERRCODE_NOT_SUPPORT	不支持的操作。

【需求】

- 头文件：xmedia_tde.h
- 库文件：libxmedia_tde.a

【注意】

- 支持的格式如下：

源位图格式	XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_CLUT8
目标位图格式	XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_CLUT8

- 此接口使用的是 DMA 搬移，因此性能优于 [xmedia_tde_add_task_single_blit](#) 和 [xmedia_tde_add_task_double_blit](#) 搬移。
- 快速拷贝操作不支持格式转换，源位图和目标位图格式必须一致。
- 快速拷贝不支持缩放功能，用户需保障源与目标操作区域大小一致。（真实操作区域是源 rect 与源 surface 的有效交集，目标 rect 与目标 surface 的有效交集）。

- 指定的操作区域要和指定的位图有公共区域，否则会返回错误；其他操作均有此要求。
- 位图基地址按照 16 字节对齐，行间距 16 字节对齐。
- RGB 域和 clut 查表的位图格式的水平起始位置和宽度按照像素对齐。

【举例】

无。

xmedia_tde_add_task_scale

【目的】

向指定工作组中添加缩放任务。

【语法】

```
xmedia_s32 xmedia_tde_add_task_scale(xmedia_s32 handle, const xmedia_tde_surface_info *src,
const xmedia_tde_surface_info *dst);
```

【描述】

向指定 handle 的工作组 job 中，添加任务 task：将源 surface 中指定的区域 rect 的内容进行放大或缩小到目标 surface 中指定的区域 rect 的位置上。

【参数】

参数名称	描述	输入/输出
handle	由 xmedia_tde_create_job 接口获取的句柄，一个 handle 对应一个工作组 job，唯一对应。	输入
src	源位图和操作区域。	输入
dst	目标位图和操作区域。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_ERRCODE_NOT_INIT	模块未初始化。
XMEDIA_ERRCODE_NOT_PERMITTED	操作不允许。(状态机不对)
XMEDIA_ERRCODE_NULL_PTR	空指针参数错误。
XMEDIA_ERRCODE_INVALID_PARAM	无效参数错误。
XMEDIA_ERRCODE_BUFFER_EMPTY	没有空闲的 task 资源。
XMEDIA_ERRCODE_NOT_SUPPORT	不支持的操作。

【需求】

- 头文件: xmedia_tde.h
- 库文件: libxmedia_tde.a

【注意】

- 支持的格式如下:

源位图格式	XMEDIA_VIDEO_PIXEL_FMT_YUV_SEMIPLANAR_420 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_420 XMEDIA_VIDEO_PIXEL_FMT_YUV_SEMIPLANAR_422 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_422 XMEDIA_VIDEO_PIXEL_FMT_YUV_SEMIPLANAR_444 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_444 XMEDIA_VIDEO_PIXEL_FMT_YUYV_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YVYU_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_UYVY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_VYUY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YYUV_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YYVU_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_UVYY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_VUYY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_VY1UY0_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YUV_400 XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_CLUT8
-------	---

目标位图格式	XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_RGB_888 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_888
--------	--

- 缩小倍数小于 128 倍，范围 w:[32, 4096]; h:[32, 4096]; 放大倍数则没有限制(不超过最大分辨率)。 [xmedia_tde_add_task_single_blit](#) 和 [xmedia_tde_add_task_double_blit](#) 接口的缩放倍数限制与此一致，不另作说明。
- 缩放时源位图和目标位图可以为同一位图，但操作区域不能有重叠，否则效果会异常。

【举例】

无。

xmedia_tde_add_task_rotation

【目的】

向指定工作组中添加旋转任务。

【语法】

```
xmedia_s32 xmedia_tde_add_task_rotation(xmedia_s32 handle, const xmedia\_tde\_surface\_info
*src, const xmedia\_tde\_surface\_info *dst, xmedia_video_rotation rot);
```

【描述】

向指定 handle 的工作组 job 中，添加任务 task：将源 surface 中指定的区域 rect 的内容旋转固定角度到目标 surface 中指定的区域 rect 的位置上，可以做 90 度，180 度和 270 度顺时针旋转。

【参数】

参数名称	描述	输入/输出
handle	由 xmedia_tde_create_job 接口获取的句柄，一个 handle 对应一个工作组 job，唯一对应。	输入
src	源位图和操作区域。	输入
dst	目标位图和操作区域。	输入
rot	旋转角度。顺时针 0,90,180,270 度。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_ERRCODE_NOT_INIT	模块未初始化。
XMEDIA_ERRCODE_NOT_PERMITTED	操作不允许。(状态机不对)
XMEDIA_ERRCODE_NULL_PTR	空指针参数错误。
XMEDIA_ERRCODE_INVALID_PARAM	无效参数错误。
XMEDIA_ERRCODE_BUFFER_EMPTY	没有空闲的 task 资源。
XMEDIA_ERRCODE_NOT_SUPPORT	不支持的操作。

【需求】

- 头文件: xmedia_tde.h
- 库文件: libxmedia_tde.a

【注意】

- 支持的格式如下:

源位图格式	XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_CLUT8
目标位图格式	XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_CLUT8

- 旋转不支持格式转换, 即输入像素格式与输出像素格式必须一致。

- 旋转不支持缩放，即输入区域宽高与输出区域宽高需要对应角度的变化，如输入为 1920x1080，旋转 90 度，输出需设置为 1080x1920。

【举例】

无。

xmedia_tde_add_task_single_blit

【目的】

向指定工作组中添加单源搬移任务(可附加操作)。附加操作可以是缩放，输出剪切，镜像，翻转，填充色块，与色块做 alpha blending 或 rop 运算等。

【语法】

```
xmedia_s32 xmedia_tde_add_task_single_blit(xmedia_s32 handle, const
xmedia_tde_surface_info*src, const xmedia_tde_surface_info *dst, xmedia_u32 fill_color,
xmedia_tde_operation_attr *opt);
```

【描述】

向指定 handle 的工作组 job 中，添加任务 task：实现对源 surface 中指定区域内容，搬移到目标 surface 中的指定区域，期间伴随一定附加操作（opt 指定附加操作）。实际应用场景可以是：

- 像素格式转换，源位图格式转换为目标位图格式。
- 色域转换，YUV 源位图转换为 RGB 域目标位图。
- 带附加操作的缩放，指定输出剪切域或镜像翻转。
- 填充色作为前景与位图做 rop 运算或 alpha blending 混合。

【参数】

参数名称	描述	输入/输出
handle	由 xmedia_tde_create_job 接口获取的句柄，一个 handle 对应一个工作组 job，唯一对应。	输入
src	源位图和操作区域。	输入
dst	目标位图和操作区域。	输入
fill_color	填充色。固定 ARGB8888，范围[0x0, 0xffffffff]。	输入
opt	附加操作集合。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_ERRCODE_NOT_INIT	模块未初始化。
XMEDIA_ERRCODE_NOT_PERMITTED	操作不允许。(状态机不对)
XMEDIA_ERRCODE_NULL_PTR	空指针参数错误。
XMEDIA_ERRCODE_INVALID_PARAM	无效参数错误。
XMEDIA_ERRCODE_BUFFER_EMPTY	没有空闲的 task 资源。
XMEDIA_ERRCODE_NOT_SUPPORT	不支持的操作。

【需求】

- 头文件：xmedia_tde.h
- 库文件：libxmedia_tde.a

【注意】

- 支持的格式如下：

源位图格式	XMEDIA_VIDEO_PIXEL_FMT_YUV_SEMIPLANAR_420 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_420 XMEDIA_VIDEO_PIXEL_FMT_YUV_SEMIPLANAR_422 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_422 XMEDIA_VIDEO_PIXEL_FMT_YUV_SEMIPLANAR_444 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_444 XMEDIA_VIDEO_PIXEL_FMT_YUYV_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YVYU_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_UYVY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_VYUY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YYUV_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YYVU_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_UVYY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_VUYY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_VY1UY0_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YUV_400 XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_CLUT8
-------	---

目标位图格式	XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_RGB_888 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_888
--------	--

- 在调用此接口前应保证调用 `xmedia_tde_int` 对 TDE 模块进行了初始化，并且调用 [xmedia_tde_create_job](#) 获得了有效的任务句柄。如无特殊说明，所有 `xmedia_tde_add_task_xxxx` 接口都必须满足此前提。
- 源位图 `src` 与目标位图 `dst` 不能为 NULL。如有任何一个为空，接口返回错误码 `XMEDIA_ERRCODE_NULL_PTR`。
- 附加操作集合 `opt` 可以为 NULL，一般只是做格式转换、色域转换、简单缩放时可以不配 `opt`。此时 `fill_color` 也无效。
- 当附加操作集合 `opt` 不为 NULL，且开启了 `rop` 或 `alpha blending` 功能，此时 `fill_color` 用户配置有效，填充色作为 S2 与源位图作为 S1 进行运算并输出结果，输出结果可做镜像和翻转，还可指定输出剪切区域。
- 当指定 `colorkey` 操作时，只能对背景 S1 做 `colorkey`，也就是对位图做 `colorkey`。
- `alpha blending` 和 `rop` 功能不支持同时开启。一个 task 只能二选一。
- `rop` 或 `alpha blending` 功能开启时，不支持对背景 S1 做缩放操作，也不支持对背景 S1 做色域转换操作。
- `clip` 操作时
 - 若为区域内 `clip`，则裁减区域必须与操作区域有公共交集，否则会返回错误。
 - 若为区域外 `clip`，则裁减区域不可完全覆盖操作区域，否则会返回错误码。也就是说，实际更新区域不能为空。
 - `clip` 和 `mirror` 功能不支持同时开启。一个 task 只能二选一。

- clip 和 flip 功能不支持同时开启。一个 task 只能二选一。

【举例】

无。

xmedia_tde_add_task_double_blit

【目的】

向指定工作组中添加双源搬移任务(可附加操作)。附加操作可以是缩放，双源 alpha blending 混合叠加，双源 rop 逻辑运算，colorkey，输出剪切，镜像，翻转，指定输出 alpha 等。

【语法】

```
xmedia_s32 xmedia_tde_add_task_double_blit(xmedia_s32 handle, const
xmedia_tde_surface_info *bg, const xmedia_tde_surface_info *fg, const xmedia_tde_surface_info
*dst, xmedia_tde_operation_attr *opt);
```

【描述】

向指定 handle 的工作组 job 中，添加任务 task：将前景 surface 指定区域内容与背景 surface 指定区域内容进行指定的运算操作(如 alpha blending, rop 运算)，将运算后的结果拷贝到目标 surface 指定区域中。opt 是指定运算的配置，如：是否进行 rop 操作和 rop 命令码；是否作色键（colorkey）及 colorkey 的配置值；是否作区域剪切（clip 操作）及指定 clip 区域；是否镜像、是否翻转、是否进行 alpha 混合等配置。实际应用场景可以是：

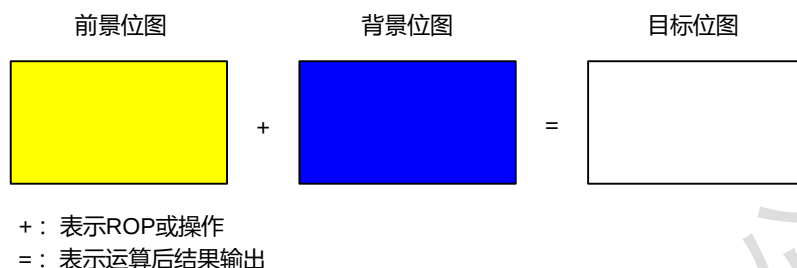
- 双源 alpha blending 混合叠加搬移输出
- 双源 rop 布尔运算搬移输出
- 双源叠加场景关键色处理
- 双源运算后指定输出剪切域、镜像翻转
- 双源运算后指定输出 alpha 来源

xmedia_tde_operation_attr 结构中的配置项涉及到的概念解释如下：

- 按位布尔运算，即 rop

rop 操作是指在将前景位图的 RGB 颜色分量和 alpha 分量值与背景位图的 RGB 颜色分量值和 alpha 分量值进行按位的布尔运算（包括按位与，按位或等），将结果输出，如图 2-1 所示。

图2-1 rop 运算的搬移操作示意图 (src1: R,G,B=0xFF,0xFF,0; src2: R,G,B=0,0,0xFF)



- Alpha 混合操作

Alpha 混合操作是基于 porter-duff 运算公式，将前景位图和背景位图根据透明度 alpha 值进行加权求和，得到 1 个 alpha 混和后的输出位图，达到两个位图按照一定透明度叠加的效果。如何取 alpha 的比例及运算取决于用户设置的 alpha 混合命令，具体的计算方法参考 [xmedia_tde_blend_mode](#) 里的介绍。

说明

预乘:

- 前景或背景位图数据是预乘后的数据，则无需配置对应的预乘使能，即 `bg_premulti_en` 或 `fg_premulti_en` 配 0。
- 前景或背景位图数据没有预乘，这时需要选择仅对应前景或背景开启预乘，即 `bg_premulti_en` 和 `fg_premulti_en` 配 1。

像素 alpha:

- 背景像素 alpha 固定使能，用户不可配。如果背景位图本身没有像素 alpha，则使用全局 alpha 替代，如果全局 alpha 用户没配，默认 0xff。
- 前景像素 alpha 用户可配，如 `pixel_alpha_en = 1`，则前景直接用像素 alpha 参与运算；如 `pixel_alpha_en = 0`，则用全局 alpha 替代，如果全局 alpha 用户没配，默认 0xff。

全局 alpha:

- 用户可以配置全局 alpha 使能与对应的数值。
- 当像素 alpha 使能情况下，会在对应的像素 alpha 基础上再乘个全局 alpha，用于全局调整形成一个新的像素 alpha 用于参与混合运算。如：前景 alpha 为 0xff，全局 alpha 为 0x80，则参与混合运算的是 0xff 乘 0x80，最终为 0x80。

- ColorKey 操作

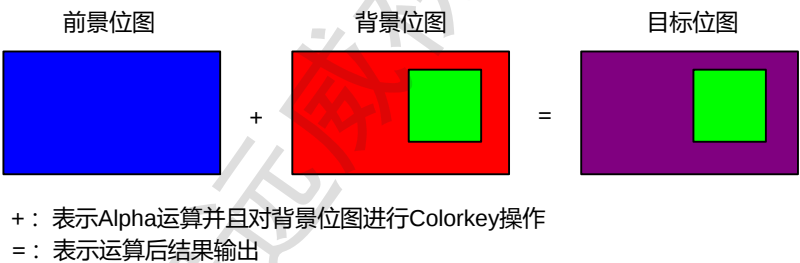
Colorkey 操作是指满足关键色范围的像素不参与 TDE 运算。Colorkey 设置需要根据像素格式对每个分量单独设置过滤条件。所有分量满足过滤条件的颜色称为关键色。Colorkey 操作有 2 种模式：

- 对前景进行 colorkey，其设置的含义是前景位图中的关键色不参与运算，将背景位图保留，即背景位图相应区域直接拷贝至输出位图，如图 2-2 所示。
- 对背景进行 colorkey，其设置的含义是背景位图中的关键色区域直接拷贝至输出位图，其他区域是运算结果，如图 2-3 所示。

图2-2 对前景位图进行 colorkey 运算的搬移操作示意图



图2-3 对背景位图进行 colorkey 运算的搬移操作示意图



- 缩放操作

当背景位图、前景位图中的操作区域若和目的位图的操作区域大小不一致时：

- 背景位图会与目标位图取交集，保障操作尺寸一致。
- 基于背景与目标位图交集尺寸已定情况下，前景位图的操作区域尺寸与目标位图尺寸不一致，就进行缩放。先前景缩放到目标位图操作尺寸，再进行混合或 rop 等运算操作。

- 镜像/翻转功能

调用者可通过 `xmedia_tde_operation_attr` 中 `mirror_en/flip_en` 配置项指定镜像，翻转功能：

- 镜像，按照水平方向进行对称拷贝。
- 翻转，按照垂直方向进行对称拷贝。
- 颜色扩展

颜色扩展功能指将精度低的色彩格式通过调色板（称为 clut 表）扩展到真彩色。如 clut8 格式的位图仅有 256 色，调用者可通过构造合适的 clut 表，将 clut 表首地址配置给位图 `xmedia_tde_surface` 的 `clut_phys_addr` 属性，TDE 就可以依靠检索 clut 表来实现 clut 8 到真彩色 ARGB 的扩展。

为了实现颜色扩展，调用者需要配置的选项为：

 - 位图 surface 结构中的 clut 表首地址 `clut_phys_addr`，该地址指向的内存必须物理连续。
 - clut 表固定为 RGB 域，即颜色表为 256 个 ARGB8888 颜色值。
- 输出图像的剪切功能，即 clip 功能

经过 TDE 处理的图像直接输出到目的位图的指定区域。而 clip 功能则可以在输出图像时，指定输出其中的一部分到目的位图，即对输出结果做了裁剪后才输出。

clip 支持两种裁剪模式：

 - 区域内裁剪：指仅更新 clip 指定范围内的区域为 TDE 运算结果。如图 2-4 所示，clip 区域与目的位图的操作区域相交，区域内裁剪就导致仅将灰色区域更新为 TDE 运算结果，目标操作区域的其它地方保持不变。
 - 区域外裁剪：指 TDE 运算结果仅更新 clip 指定范围外的区域。如图 2-5 所示，clip 区域与目的位图的操作区域相交，区域外裁剪就导致仅将灰色区域更新为 TDE 运算结果，clip 区域内部的地方保持不变。

图2-4 区域内 clip 示意图

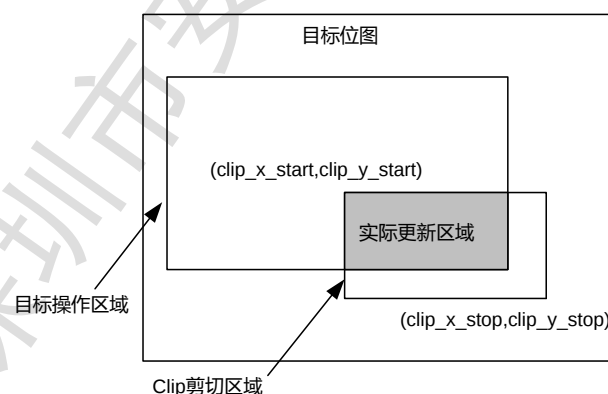
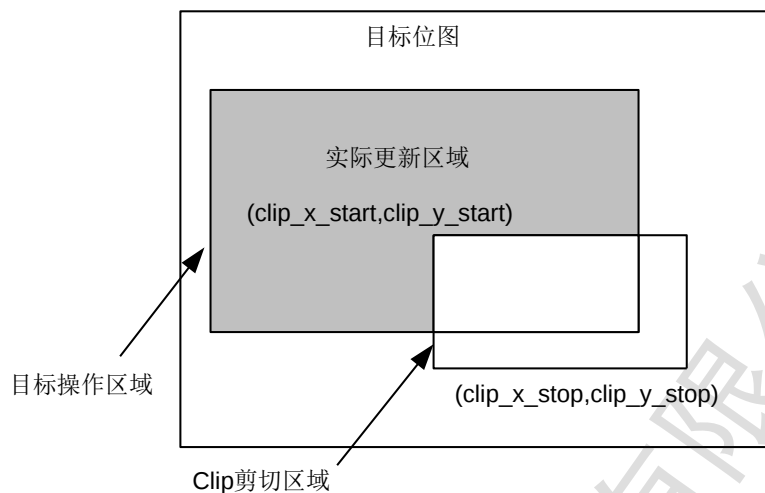


图2-5 区域外 clip 示意图



- 输出 Alpha 来源
有 4 种模式供选择：
 - 来源于运算结果
 - 来源于前景位图
 - 来源于背景位图
 - 来源于全局 Alpha



说明

对于 alpha 叠加运算，则需要选择来源于运算结果模式。

【参数】

参数名称	描述	输入/输出
handle	由 xmedia_tde_create_job 接口获取的句柄，一个 handle 对应一个工作组 job，唯一对应。	输入
bg	背景位图和操作区域。	输入
fg	前景位图和操作区域。	输入
dst	目标位图和操作区域。	输入
opt	附加操作配置集合。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_ERRCODE_NOT_INIT	模块未初始化。
XMEDIA_ERRCODE_NOT_PERMITTED	操作不允许。(状态机不对)
XMEDIA_ERRCODE_NULL_PTR	空指针参数错误。
XMEDIA_ERRCODE_INVALID_PARAM	无效参数错误。
XMEDIA_ERRCODE_BUFFER_EMPTY	没有空闲的 task 资源。
XMEDIA_ERRCODE_NOT_SUPPORT	不支持的操作。

【需求】

- 头文件：xmedia_tde.h
- 库文件：libxmedia_tde.a

【注意】

- 支持的格式如下：

背景位图格式	XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555
	XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444
	XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888
	XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555
	XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444
	XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888
	XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555
	XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444
	XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888
	XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555
	XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444
	XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888
	XMEDIA_VIDEO_PIXEL_FMT_RGB_565
	XMEDIA_VIDEO_PIXEL_FMT_BGR_565
	XMEDIA_VIDEO_PIXEL_FMT_CLUT8

前景位图格式	XMEDIA_VIDEO_PIXEL_FMT_YUV_SEMIPLANAR_420 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_420 XMEDIA_VIDEO_PIXEL_FMT_YUV_SEMIPLANAR_422 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_422 XMEDIA_VIDEO_PIXEL_FMT_YUV_SEMIPLANAR_444 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_444 XMEDIA_VIDEO_PIXEL_FMT_YUYV_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YVYU_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_UYVY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_VYUY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YYUV_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YYVU_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_UVYY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_VUYY_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_VY1UY0_PACKAGE_422 XMEDIA_VIDEO_PIXEL_FMT_YUV_400 XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_CLUT8
--------	---

目标位图格式	XMEDIA_VIDEO_PIXEL_FMT_ARGB_1555 XMEDIA_VIDEO_PIXEL_FMT_ARGB_4444 XMEDIA_VIDEO_PIXEL_FMT_ARGB_8888 XMEDIA_VIDEO_PIXEL_FMT_ABGR_1555 XMEDIA_VIDEO_PIXEL_FMT_ABGR_4444 XMEDIA_VIDEO_PIXEL_FMT_ABGR_8888 XMEDIA_VIDEO_PIXEL_FMT_RGBA_1555 XMEDIA_VIDEO_PIXEL_FMT_RGBA_4444 XMEDIA_VIDEO_PIXEL_FMT_RGBA_8888 XMEDIA_VIDEO_PIXEL_FMT_BGRA_1555 XMEDIA_VIDEO_PIXEL_FMT_BGRA_4444 XMEDIA_VIDEO_PIXEL_FMT_BGRA_8888 XMEDIA_VIDEO_PIXEL_FMT_RGB_565 XMEDIA_VIDEO_PIXEL_FMT_RGB_888 XMEDIA_VIDEO_PIXEL_FMT_BGR_565 XMEDIA_VIDEO_PIXEL_FMT_BGR_888
--------	--

- 仅支持单个源的格式配置为 clut8 格式，用于颜色扩展。即不能两个源都同时为 clut8 格式，且输出不能是 clut8 格式。
- [xmedia_tde_add_task_double_blit](#) 必须背景与前景均配置有效 surface，不支持配置为 NULL。如果是单源(带附加操作)搬移，使用 [xmedia_tde_add_task_single_blit](#) 接口。
- 在调用此接口前应保证调用 [xmedia_tde_init](#) 初始化 TDE 模块，并且调用 [xmedia_tde_create_job](#) 获得了有效的工作组句柄。
- 当前景源位图与目标位图尺寸不一致时，会先将前景源位图缩放到目标位图尺寸大小，再基于缩放后的图像镜像附加操作运算。
- global_alpha 和 alpha0、alpha1 的设置值统一按照[0, 255]的范围进行设置。
- 背景位图可以与目标位图为同一位图，但不允许重叠。
- clip 操作时
 - 若为区域内 clip，则裁减区域必须与操作区域有公共交集，否则会返回错误。
 - 若为区域外 clip，则裁减区域不可完全覆盖操作区域，否则会返回错误码。也就是说，实际更新区域不能为空。
 - clip 和 mirror 功能不支持同时开启。一个 task 只能二选一。
 - clip 和 flip 功能不支持同时开启。一个 task 只能二选一。

- 在第一次作颜色扩展操作（源为 clut 格式，目的为非 clut 格式）时，需要打开 clut_reload 标记。
- rop 操作时，通过操作结构体 `xmedia_tde_rop_attr` 中的成员 `alpha_rop` 和 `color_rop` 分别指定透明度分量和颜色分量进行的 rop 操作。其中，rop 操作类型中的 S1 指背景位图 bg，S2 指前景位图 fg。

【举例】

无。

xmedia_tde_get_alpha_threshold_value

【目的】

获取 alpha 判决阈值。

【语法】

```
xmedia_s32 xmedia_tde_get_alpha_threshold_value(xmedia_u8 *value);
```

【描述】

获取 alpha 判决阈值。用于结果图片像素格式为 ARGB1555 的情况。若前景位图和背景位图的 alpha 运算结果小于此阈值，结果像素的 alpha 位取 0；大于或等于此阈值，像素的 alpha 位取 1。

【参数】

参数名称	描述	输入/输出
value	指向 alpha 判决阈值的指针。	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_ERRCODE_NOT_INIT	tde 模块未初始化。
XMEDIA_ERRCODE_NULL_PTR	空指针参数错误。
XMEDIA_ERRCODE_INVALID_PARAM	参数不合法。

【需求】

- 头文件: xmedia_tde.h
- 库文件: libxmedia_tde.a

【注意】

无。

【举例】

无。

xmedia_tde_set_alpha_threshold_value

【目的】

设置 alpha 判决阈值。

【语法】

```
xmedia_s32 xmedia_tde_set_alpha_threshold_value(xmedia_u8 value);
```

【描述】

设置 alpha 判决阈值。当前景和背景做 alpha blending 操作时，不管前景位图和背景位图的格式为什么，硬件都会生成 ARGB8888 的中间位图格式，若目标图片像素格式为 ARGB1555，则若前景位图和背景位图的 alpha 运算结果小于此阈值，结果像素的 alpha 位取 0；大于或等于此阈值，像素的 alpha 位取 1。

【参数】

参数名称	描述	输入/输出
value	alpha 判决阈值。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_ERRCODE_NOT_INIT	TDE 设备未打开，API 调用失败。
XMEDIA_ERRCODE_INVALID_PARAM	无效的参数设置。

【需求】

- 头文件：xmedia_tde.h
- 库文件：libxmedia_tde.a

【注意】

无。

【举例】

无。

xmedia_tde_cancel_job

【目的】

取消指定的工作组，并取消对应已经添加到工作组中的任务。

【语法】

```
xmedia_s32 xmedia_tde_cancel_job(xmedia_s32 handle);
```

【描述】

向 TDE 工作组添加任务时，如果出现当前的操作参数非法等错误，程序需要返回退出时，可调用此接口取消当前工作组。

【参数】

参数名称	描述	输入/输出
handle	由 xmedia_tde_create_job 接口获取的句柄，一个 handle 对应一个工作组 job，唯一对应。	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_ERRCODE_NOT_INIT	tde 模块未初始化
XMEDIA_ERRCODE_INVALID_PARAM	参数不合法。
XMEDIA_ERRCODE_NOT_PERMITTED	操作不允许。

【需求】

- 头文件：xmedia_tde.h
- 库文件：libxmedia_tde.a

【注意】

- 在调用此接口前应保证调用 [xmedia_tde_init](#) 初始化 TDE 模块，并且调用 [xmedia_tde_create_job](#) 获得了有效的任务句柄，否则返回值无效。
- 已经提交的工作组不能够取消。
- 取消后的工作组不能再向其添加任务，也不能再次提交该工作组。
- 在向 TDE 工作组(job)中添加任务（如 task A）时出错可以有以下两种处理方式：
 - 忽略出错的 task A，继续向 TDE 工作组中添加其余任务，并提交该工作组。若该工作组成功执行，则说明所有成功添加的任务都完成了，A 任务因未添加成功，不会有对应 A 任务的处理。

- 因添加任务 A 出错而取消整个工作组，则说明该任务连同其下所有已成功添加的任务，连同工作组都被取消。

【举例】

```
/* declaration */
xmedia_s32 ret;
xmedia_s32 handle;
xmedia_tde_surface_info src;
xmedia_tde_surface_info dst;
xmedia_tde_operation_attr opt = {0};

/* 初始化TDE模块 */
ret = xmedia_tde_init();
if (ret != XMEDIA_SUCCESS) {
    return ret;
}
/* create a TDE job */
ret = xmedia_tde_create_job(&handle);
if (ret != XMEDIA_SUCCESS) {
    xmedia_tde_exit();
    return ret;
}

/* add serial commands to job successfully*/
/* prepare arguments of bitblt command */

/* if fail to add one more bitblt command to the job, cancel the job */
ret = xmedia_tde_add_task_double_blit(handle, &src, &dst, &opt);
if (ret != XMEDIA_SUCCESS) {
    xmedia_tde_cancel_job(handle);
    xmedia_tde_exit();
    return ret;
}
```

xmedia_tde_wait_job

【目的】

等待指定的工作组完成。

【语法】

```
xmedia_s32 xmedia_tde_wait_job(xmedia_s32 handle, xmedia_u32 timeout);
```

【描述】

当使用 `xmedia_tde_submit_job` 提交 TDE 工作组后，可以使用此接口等待指定的 TDE 工作组完成。此接口会判断硬件是否将指定工作组完成，如果未完成会阻塞进程直到硬件完成工作或超时。

【参数】

参数名称	描述	输入/输出
handle	由 <code>xmedia_tde_create_job</code> 接口获取的句柄，一个 handle 对应一个工作组 job，唯一对应。	输入
timeout	等待时间(ms)	输入

【返回值】

返回值	描述
0	指定的 TDE 任务完成。
非 0	失败，其值为错误码。

【错误码】

返回值	描述
XMEDIA_ERRCODE_NOT_INIT	tde 模块未初始化。
XMEDIA_ERRCODE_INVALID_PARAM	参数不合法。
XMEDIA_ERRCODE_TIMEOUT	等待超时。

【需求】

- 头文件：xmedia_tde.h
- 库文件：libxmedia_tde.a

【注意】

- 此接口为阻塞接口，会阻塞等待指定的工作组完成。

- 不允许等待一个未提交的工作组，否则返回相应错误码。

【举例】

无。

仅限深圳市安远威视科技有限公司使用

3数据类型

3.1 映射表

本章详细描述了 API 中涉及的数据结构，如表 3-1 所示。

表3-1 TDE 数据结构映射表

函数	说明
xmedia_tde_surface	surface 结构体，含图层信息：像素格式、宽高、行间距、颜色表、内存首地址等。
xmedia_tde_surface_info	surface 结构体和操作区域 rect。
xmedia_tde_line_attr	画线的线条属性。
xmedia_tde_operation_attr	附加操作集合，含：alpha blending、rop、colorkey、clip、mirror、flip 等。
xmedia_tde_blend_attr	alpha blending 功能的属性集合，含：全局 alpha、像素 alpha、预乘、叠加方式等属性配置。
xmedia_tde_blend_mode	alpha blending 功能的叠加模式。
xmedia_tde_rop_attr	rop 逻辑运算功能的属性集合，含：alpha 的运算模式、颜色 rgb 的运算模式。
xmedia_tde_rop_mode	rop 逻辑运算模式
xmedia_tde_colorkey_attr	关键色属性集合，含：colorkey 模式，colorkey 值，colorkey 分量。

函数	说明
xmedia_tde_colorkey_mode	关键色模式。
xmedia_tde_colorkey_value	关键色值。
xmedia_tde_colorkey_comp	关键色分量。
xmedia_tde_clip_attr	剪切域属性，含：剪切模式、剪切区域。
xmedia_tde_clip_mode	剪切模式。
xmedia_tde_out_alpha_mode	输出 alpha 来源模式。

3.2 详细描述

xmedia_tde_surface

【说明】

位图 surface 结构体。

【定义】

```
#define XMEDIA_COLOR_COMPONENT_NUM 3
typedef struct {
    xmedia_u64 phys_addr[XMEDIA_COLOR_COMPONENT_NUM];
    xmedia_u32 stride[XMEDIA_COLOR_COMPONENT_NUM];
    xmedia_u32 width;
    xmedia_u32 height;
    xmedia_video_pixel_format pixel_format;
    xmedia_bool is_alpha_ext_1555;
    xmedia_u8 alpha0;
    xmedia_u8 alpha1;
    xmedia_u64 clut_phys_addr;
    xmedia_bool clut_reload;
} xmedia_tde_surface;
```

【成员】

成员	描述
phys_addr	位图首地址。数组下标： 0: Y 或 RGB 1: UV 2: U 或 V
stride	行间距。数组下标： 0: Y 或 RGB 1: UV 2: U 或 V
width	位图宽度。
height	位图高度。
pixel_format	像素格式。
is_alpha_ext_1555	是否使能 1555 的 Alpha 扩展。 当位图格式为 ARGB1555 时，该项有效。
alpha0	Alpha0 值。 取值范围：[0, 255]。 当位图格式为 ARGB1555 且 is_alpha_ext_1555 为 TRUE 时，该项有效。 在 ARGB1555 格式下，当像素的最高位为 0 时，选择该值作为 alpha 叠加的 alpha 值。
alpha1	Alpha1 值。 取值范围：[0, 255]。 当位图格式为 ARGB1555 且 is_alpha_ext_1555 为 TRUE 时，该项有效。 在 ARGB1555 格式下，当像素的最高位为 1 时，选择该值作为 alpha 叠加的 alpha 值。
clut_phys_addr	clut 表首地址，用于颜色扩展查表。支持 256 个 ARGB8888 的颜色。
clut_reload	颜色表是否重载。

【注意事项】

- RGB 域内各像素格式的位图的水平起始位置、宽度、高度按照像素对齐。
- 像素格式位深不足 Byte 的位图起始地址、宽度、高度必须按照单字节对齐。
- YUV 域内个像素格式的位图的水平起始位置、宽度、高度必须像素 2 对齐。
- clut 到真彩色 ARGB 的扩展是依靠检索 clut 表来实现的。故颜色扩展功能（如 clut 格式位图扩展到 ARGB8888 格式位图）或颜色校正时，需要配置 clut 表首地址 clut_phys_addr，且其指向的 clut 表内存必须是物理连续的。
- stride 按 16 字节对齐，最大不超过 65535。

【相关数据类型和接口】

[xmedia_tde_surface_info](#)

xmedia_tde_surface_info

【说明】

位图 surface 和操作区域 rect 的集合。

【定义】

```
typedef struct {  
    xmedia_tde_surface surface;  
    xmedia_video_rect rect;  
} xmedia_tde_surface_info;
```

【成员】

成员	描述
surface	位图 surface。

成员	描述
rect	指定操作区域。 XM7606V13: rect.x 范围: [-4064, 4064] rect.y 范围: [-4064, 4064] rect.width 范围: [32, 4096] rect.height 范围: [32, 4096] XM7206V11A: rect.x 范围: [0, 4064] rect.y 范围: [0, 4064] rect.width 范围: [32, 4096] rect.height 范围: [32, 4096]

【注意事项】

- 操作区域与位图的关系如图 3-1 所示。
- [xmedia_tde_surface_info](#) 包括 surface 和 rect 两部分属性。surface 主要描述位图的基本信息，包括：位图的像素宽度、像素高度、位图每行的跨度、像素格式、位图存放的物理地址等。rect 描述本次操作指定的矩形范围，包括：起始位置和尺寸信息。

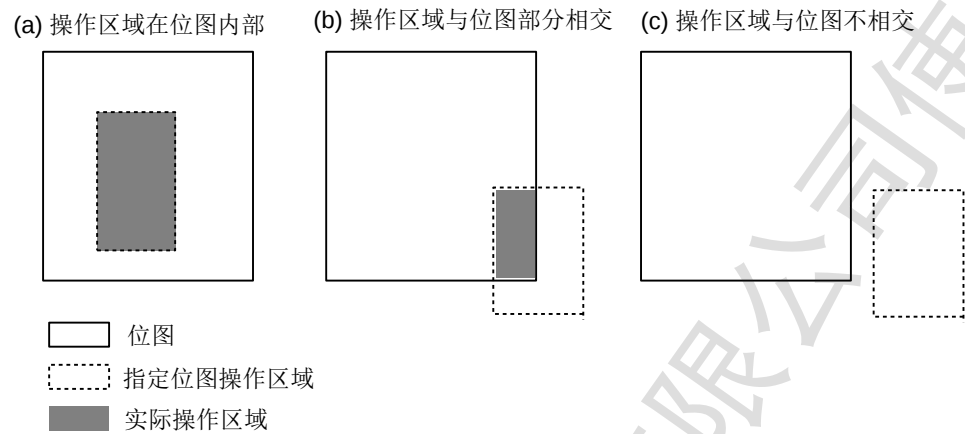
通过指定不同的操作区域，用户可指定位图的全部或部分参与操作。

- 若希望整个位图都参与操作，则指定操作区域起点为 (0, 0)，宽高分别为位图的宽高。
- 若希望位图的部分区域参与操作，则指定适当的操作区域大小。如图 3-1 中的情况 (a)，指定的操作区域即为有效的操作区域。注意：如果指定的操作区域与位图部分相交，如情况(b)，则自动裁剪指定操作区域，故有效的操作区域为灰色相交部分。
- 若指定的操作区域与位图不相交，如情况(c)，则认为配置错误，返回错误码 XMEDIA_ERRCODE_INVALID_PARAM。

📖 说明

有效操作区域：指调用者指定的操作区域与位图的相交部分。

图3-1 位图与位图中的操作区域的关系



【相关数据类型和接口】

[xmedia_tde_surface](#)

[xmedia_video_rect](#)

所有 [xmedia_tde_add_task_xxxx](#) 接口

[xmedia_tde_line_attr](#)

【说明】

画线属性。

【定义】

```
typedef struct {  
    xmedia_video_point start_point;  
    xmedia_video_point end_point;  
    xmedia_u32 thick;  
    xmedia_u32 color;  
} xmedia_tde_line_attr;
```

【成员】

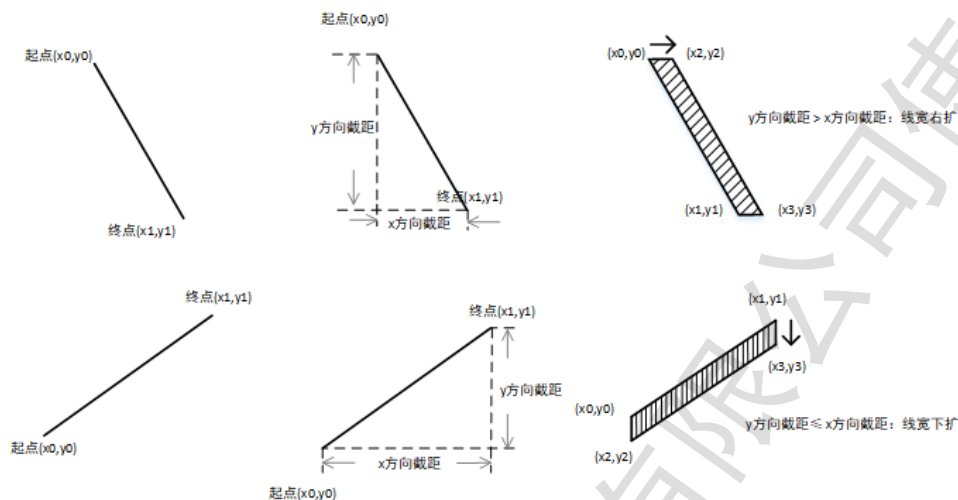
成员	描述
start_point	画线起点。 start_point.x 范围：[0, 65535] start_point.y 范围：[0, 65535]

成员	描述
end_point	画线终点。 end_point.x 范围：[0, 65535] end_point.y 范围：[0, 65535]
thick	线宽。 范围：(0, 255]
color	线条颜色。 范围：[0x0, 0xffffffff], 32bit 位。

【注意事项】

- 起点不能与终点重合。
- XM7606V13 要求起点和终点不能同时在图像有效区外，XM7206V11A 要求起点和终点均不能在图像有效区外。
- 起点和终点不能同时在图像右边界上，如：图像区域宽高为 1920x1080，起点 (1920, 0) 终点 (1920, 1080)。
- 起点和终点不能同时在图像下边界上，如：图像区域宽高为 1920x1080，起点 (0, 1080) 终点 (1920, 1080)
- 画线是由起终点根据线宽扩展得到四点的四边形绘制，不支持四边形任意一点超过[0, 65535]。
- 线宽效果如图 3-2，沿线如果垂直方向变化大于水平方向（y 方向截距大于 x 方向截距），则水平向右扩展线的宽度。如果水平方向变化大于或等于垂直方向，则垂直向下扩展线的宽度。

图3-2 画线线宽效果



- 画线起点、终点、线宽按照像素对齐。

【相关数据类型和接口】

xmedia_video_point

[xmedia_tde_add_task_draw_line](#)

xmedia_tde_operation_attr

【说明】

位图搬移附加操作集合。

【定义】

```
typedef struct {
    xmedia_bool blend_en;
    xmedia\_tde\_blend\_attr blend_attr;
    xmedia_bool rop_en;
    xmedia\_tde\_rop\_attr rop_attr;
    xmedia_bool colorkey_en;
    xmedia\_tde\_colorkey\_attr ck_attr;
    xmedia_bool clip_en;
    xmedia\_tde\_clip\_attr clip_attr;
    xmedia_bool mirror_en;
    xmedia_bool flip_en;
    xmedia\_tde\_out\_alpha\_mode alpha_mode;
    xmedia_u8 global_alpha;
} xmedia_tde_operation_attr;
```

【成员】

成员	描述
blend_en	alpha blending 功能使能。 0: 不开 1: 开
blend_attr	alpha blending 功能配置属性。
rop_en	rop 运算使能。 0: 不开 1: 开
rop_attr	rop 运算功能配置属性。
colorkey_en	关键色使能。 0: 不开 1: 开
ck_attr	关键色功能配置属性。
clip_en	输出剪切域使能。 0: 不开 1: 开
clip_attr	输出剪切域配置属性。
mirror_en	水平镜像使能。 0: 不开 1: 开
flip_en	垂直翻转使能。 0: 不开 1: 开
alpha_mode	输出结果 alpha 来源。
global_alpha	全局 alpha 值。 范围: [0x0, 0xff]

【注意事项】

- alpha blending 功能与 rop 功能互斥，只能二选一，两者都开会返回相应错误码。
- 关键色功能是在 alpha blending 功能中生效，具体效果详见接口描述。

【相关数据类型和接口】

[xmedia_tde_add_task_single_blit](#)

[xmedia_tde_add_task_double_blit](#)

xmedia_tde_blend_attr

【说明】

Alpha blending 功能配置属性。

【定义】

```
typedef struct {  
    xmedia_bool global_alpha_en;  
    xmedia_bool pixel_alpha_en;  
    xmedia_bool bg_premulti_en;  
    xmedia_bool fg_premulti_en;  
    xmedia\_tde\_blend\_mode blend_mode;  
} xmedia_tde_blend_attr;
```

【成员】

成员	描述
global_alpha_en	全局 alpha 使能。 0：不开 1：开
pixel_alpha_en	像素 alpha 使能。 0：不开 1：开
bg_premulti_en	背景像素预乘使能。 0：不开 1：开

成员	描述
fg_premulti_en	前景像素预乘使能。 0: 不开 1: 开
blend_mode	alpha blending 的运算公式，基于 potter-duff 公式。

【注意事项】

- 上表的 pixel_alpha_en 特指前景像素 alpha 使能配置，背景是固定像素 alpha 使能开启(不可配)。
- 背景/前景预乘使能正确理解与使用场景：用户确定输入背景/前景位图的图像数据已经是做了预乘的就配 0，未做预乘可以配 1 让硬件在 alpha blending 运算前对位图进行预乘。

【相关数据类型和接口】

[xmedia_tde_operation_attr](#)

xmedia_tde_blend_mode

【说明】

alpha blending 运算公式。

【定义】

```
typedef enum {
    XMEDIA_TDE_BLEND_MODE_CLEAR = 0,
    XMEDIA_TDE_BLEND_MODE_SRC,
    XMEDIA_TDE_BLEND_MODE_DST,
    XMEDIA_TDE_BLEND_MODE_SRCOVER,
    XMEDIA_TDE_BLEND_MODE_DSTOVER,
    XMEDIA_TDE_BLEND_MODE_SRCIN,
    XMEDIA_TDE_BLEND_MODE_DSTIN,
    XMEDIA_TDE_BLEND_MODE_SRCOUT,
    XMEDIA_TDE_BLEND_MODE_DSTOUT,
    XMEDIA_TDE_BLEND_MODE_SRCATOP,
    XMEDIA_TDE_BLEND_MODE_DSTATOP,
    XMEDIA_TDE_BLEND_MODE_XOR,
    XMEDIA_TDE_BLEND_MODE_DARKEN,
```

```

XMEDIA_TDE_BLEND_MODE_LIGHTEN,
XMEDIA_TDE_BLEND_MODE_MULTIPLY,
XMEDIA_TDE_BLEND_MODE_SCREEN,
XMEDIA_TDE_BLEND_MODE_MAX
} xmedia_tde_blend_mode;

```

【成员】

pixel = (foreground x fs + background x fd), 其中:

- fs: foreground blend coefficient.
- fd: destination blend coefficient.
- pixel: 运算以后的像素值。
- foreground: 前景位图的像素值(预乘后的像素值)。
- background: 背景位图的像素值(预乘后的像素值)。
- sa: foreground alpha, 前景位图 alpha 归一化值。
- da: background alpha, 背景位图 alpha 归一化值。

fs 和 fd 分别为源位图像素和目的位图像素的系数, 选择下表中不同的项代表不同的系数。

成员	描述
XMEDIA_TDE_BLEND_MODE_CLEAR	fs: 0.0 fd: 0.0 pixel = 0
XMEDIA_TDE_BLEND_MODE_SRC	fs: 1.0 fd: 0.0 pixel = fg
XMEDIA_TDE_BLEND_MODE_DST	fs: 0.0 fd: 1.0 pixel = bg
XMEDIA_TDE_BLEND_MODE_SRCOVER	fs: 1.0 fd: 1.0-sa pixel = fg*1 + bg*(1-sa) = fg + bg - bg*sa
XMEDIA_TDE_BLEND_MODE_DSTOVER	fs: 1.0-da fd: 1.0 pixel = fg*(1-da) + bg = fg + bg - fg*da
XMEDIA_TDE_BLEND_MODE_SRCIN	fs: da fd: 0.0 pixel = fg*da
XMEDIA_TDE_BLEND_MODE_DSTIN	fs: 0.0 fd: sa pixel = bg*sa

成员	描述
XMEDIA_TDE_BLEND_MODE_SRCOUT	fs: 1.0-da fd: 0.0 pixel = fg(1-da) = fg - fg*da
XMEDIA_TDE_BLEND_MODE_DSTOUT	fs: 0.0 fd: 1.0-sa pixel = bg(1-sa) = bg - bg*sa
XMEDIA_TDE_BLEND_MODE_SRCATOP	fs: da fd: 1.0-sa pixel = fg*da + bg(1-sa) = bg + fg*da - bg*sa
XMEDIA_TDE_BLEND_MODE_DSTATOP	fs: 1.0-da fd: sa pixel = fg(1-da)+ bg*sa = fg + bg*sa - fg*da
XMEDIA_TDE_BLEND_MODE_XOR	fs: 1.0-da fd: 1.0-sa pixel = fg(1-da) + bg(1-sa) = fg + bg - (fg*da + bg*sa)
XMEDIA_TDE_BLEND_MODE_DARKEN	在 XOR 的基础上+min(foreground, background) pixel = fg + bg + min(fg,bg) - (fg*da + bg*sa)
XMEDIA_TDE_BLEND_MODE_LIGHTEN	在 XOR 的基础上+max(foreground, background) pixel = fg + bg + max(fg,bg) - (fg*da + bg*sa)
XMEDIA_TDE_BLEND_MODE_MULTIPLY	正片叠底 pixel = fg*bg
XMEDIA_TDE_BLEND_MODE_SCREEN	滤色 pixel = 1-(1-fg)(1-bg)= fg + bg- fg*bg
XMEDIA_TDE_BLEND_MODE_MAX	枚举尾巴。

【注意事项】

无。

【相关数据类型和接口】

[xmedia_tde_blend_attr](#)

[xmedia_tde_rop_attr](#)

【说明】

rop 位运算属性。

【定义】

```
typedef struct {  
    xmedia_tde_rop_mode alpha_rop;  
    xmedia_tde_rop_mode color_rop;  
} xmedia_tde_rop_attr;
```

【成员】

成员	描述
alpha_rop	alpha 分量的 rop 运算模式。
color_rop	rgb 分量的 rop 运算模式。

【注意事项】

无。

【相关数据类型和接口】

[xmedia_tde_operation_attr](#)

xmedia_tde_rop_mode

【说明】

rop 运算公式。

【定义】

```
typedef enum {  
    XMEDIA_TDE_ROP_MODE_BLACKNESS = 0,  
    XMEDIA_TDE_ROP_MODE_NOTMERGEPEN,  
    XMEDIA_TDE_ROP_MODE_MASKNOTPEN,  
    XMEDIA_TDE_ROP_MODE_NOTCOPYPEN,  
    XMEDIA_TDE_ROP_MODE_MASKPENNOT,  
    XMEDIA_TDE_ROP_MODE_NOT,  
    XMEDIA_TDE_ROP_MODE_XORPEN,  
    XMEDIA_TDE_ROP_MODE_NOTMASKPEN,  
    XMEDIA_TDE_ROP_MODE_MASKPEN,  
    XMEDIA_TDE_ROP_MODE_NOTXORPEN,  
    XMEDIA_TDE_ROP_MODE_NOP,  
    XMEDIA_TDE_ROP_MODE_MERGEOTPEN,  
}
```

```

XMEDIA_TDE_ROP_MODE_COPYPEN,
XMEDIA_TDE_ROP_MODE_MERGEPENNOT,
XMEDIA_TDE_ROP_MODE_MERGEPEN,
XMEDIA_TDE_ROP_MODE_WHITE,
XMEDIA_TDE_ROP_MODE_MAX
} xmedia_tde_rop_mode;

```

【成员】

所有位运算都是基于 ARGB8888 格式下进行运算，其中：

- S1：背景像素。
- S2：前景像素。

成员	描述
XMEDIA_TDE_ROP_MODE_BLACKNESS	Blackness
XMEDIA_TDE_ROP_MODE_NOTMERGEPEN	$\sim(S2 \mid S1)$
XMEDIA_TDE_ROP_MODE_MASKNOTPEN	$\sim S2 \& S1$
XMEDIA_TDE_ROP_MODE_NOTCOPYPEN	$\sim S2$
XMEDIA_TDE_ROP_MODE_MASKPENNOT	$S2 \& \sim S1$
XMEDIA_TDE_ROP_MODE_NOT	$\sim S1$
XMEDIA_TDE_ROP_MODE_XORPEN	$S2 \wedge S1$
XMEDIA_TDE_ROP_MODE_NOTMASKPEN	$\sim(S2 \& S1)$
XMEDIA_TDE_ROP_MODE_MASKPEN	$S2 \& S1$
XMEDIA_TDE_ROP_MODE_NOTXORPEN	$\sim(S2 \wedge S1)$
XMEDIA_TDE_ROP_MODE_NOP	$S1$
XMEDIA_TDE_ROP_MODE_MERGENOTPEN	$\sim S2 \mid S1$
XMEDIA_TDE_ROP_MODE_COPYPEN	$S2$
XMEDIA_TDE_ROP_MODE_MERGEPENNOT	$S2 \mid \sim S1$
XMEDIA_TDE_ROP_MODE_MERGEPEN	$S2 \mid S1$
XMEDIA_TDE_ROP_MODE_WHITE	Whiteness
XMEDIA_TDE_ROP_MODE_MAX	枚举尾巴

【注意事项】

无。

【相关数据类型和接口】

[xmedia_tde_rop_attr](#)

xmedia_tde_colorkey_attr

【说明】

关键色属性。

【定义】

```
typedef struct {  
    xmedia_tde_colorkey_mode ck_mode;  
    xmedia_tde_colorkey_value ck_value;  
} xmedia_tde_colorkey_attr;
```

【成员】

成员	描述
ck_mode	关键色模式。
ck_value	关键色指定值。

【注意事项】

关键色功能是针对 alpha blending 功能的一个扩展，使用场景就是当用户需要通过 alpha blending 功能叠加图像时，可以通过指定关键色更灵活的做出更多的效果。

【相关数据类型和接口】

[xmedia_tde_operation_attr](#)

xmedia_tde_colorkey_mode

【说明】

关键色模式。

【定义】

```
typedef enum {  
    XMEDIA_TDE_COLORKEY_MODE_FOREGROUND = 0,  
    XMEDIA_TDE_COLORKEY_MODE_BACKGROUND,  
    XMEDIA_TDE_COLORKEY_MODE_MAX
```

```
} xmedia_tde_colorkey_mode;
```

【成员】

成员	描述
XMEDIA_TDE_COLORKEY_MODE_FOREGROUND	对前景位图做关键色。
XMEDIA_TDE_COLORKEY_MODE_BACKGROUND	对背景位图做关键色。
XMEDIA_TDE_COLORKEY_MODE_MAX	枚举尾巴。

【注意事项】

无。

【相关数据类型和接口】

[xmedia_tde_colorkey_attr](#)

[xmedia_tde_colorkey_value](#)

【说明】

关键色数值。

【定义】

```
typedef union {
    struct {
        xmedia_tde_colorkey_comp alpha;
        xmedia_tde_colorkey_comp red;
        xmedia_tde_colorkey_comp green;
        xmedia_tde_colorkey_comp blue;
    } argb_color_key;
    struct {
        xmedia_tde_colorkey_comp alpha;
        xmedia_tde_colorkey_comp y;
        xmedia_tde_colorkey_comp cb;
        xmedia_tde_colorkey_comp cr;
    } yuv_color_key;
    struct {
        xmedia_tde_colorkey_comp alpha;
        xmedia_tde_colorkey_comp clut;
    } clut_color_key;
} xmedia_tde_colorkey_value;
```

【成员】

联合体中的数据结构 `argb_color_key`:

成员	描述
alpha	alpha 分量 key 属性。
red	red 分量 key 属性。
green	green 分量 key 属性。
blue	blue 分量 key 属性。

联合体中的数据结构 `yuv_color_key`:

成员	描述
alpha	alpha 分量 key 属性。
y	y 分量 key 属性。
cb	cb 分量 key 属性。
cr	cr 分量 key 属性。

联合体中的数据结构 `clut_color_key`:

成员	描述
alpha	alpha 分量 key 属性。
clut	clut 索引分量 key 属性。

【注意事项】

配置 key 值按照位图的像素格式进行匹配，如位图是 RGB 域的格式就对应 `argb_color_key`；如位图是 YUV 域的格式就对应 `yuv_color_key`；如果位图是索引就对应 `clut_color_key`。

【相关数据类型和接口】

[xmedia_tde_colorkey_attr](#)

xmedia_tde_colorkey_comp

【说明】

关键色各分量的属性。

【定义】

```
typedef struct {
    xmedia_u8 component_min;
    xmedia_u8 component_max;
    xmedia_u8 component_mask;
    xmedia_bool is_component_out;
    xmedia_bool is_component_ignore;
} xmedia_tde_colorkey_comp;
```

【成员】

成员	描述
component_min	key 值左边界。 范围：[0x0, 0xff]
component_max	key 值右边界。 范围：[0x0, 0xff]
component_mask	key 值掩码。 范围：[0x0, 0xff]
is_component_out	关键色生效范围。 0: 范围内，即左边界与右边界之间 1: 范围外，左边界以左，右边界以右
is_component_ignore	分量是否忽略。 0: 此分量参与关键色处理 1: 此分量不参与关键色处理

【注意事项】

无。

【相关数据类型和接口】

[xmedia_tde_colorkey_value](#)

xmedia_tde_clip_attr

【说明】

输出剪切域属性。

【定义】

```
typedef struct {  
    xmedia\_tde\_clip\_mode clip_mode;  
    xmedia_video_rect clip_rect;  
} xmedia_tde_clip_attr;
```

【成员】

成员	描述
clip_mode	剪切模式。
clip_rect	剪切区域。 clip_rect.x 范围：[0, 4064] clip_rect.y 范围：[0, 4064] clip_rect.width 范围：[32, 4096] clip_rect.height 范围：[32, 4096]

【注意事项】

- 区域内剪切，裁剪区域必须与操作区域有公共交集；
- 区域外剪切，裁剪区域不可完全覆盖操作区域；

【相关数据类型和接口】

[xmedia_tde_operation_attr](#)

xmedia_tde_clip_mode

【说明】

输出剪切域模式。

【定义】

```
typedef enum {
```

```

XMEDIA_TDE_CLIP_MODE_INSIDE = 0,
XMEDIA_TDE_CLIP_MODE_OUTSIDE,
XMEDIA_TDE_CLIP_MODE_MAX
} xmedia_tde_clip_mode;

```

【成员】

成员	描述
XMEDIA_TDE_CLIP_MODE_INSIDE	仅更新 surface 与 clip_rect 的交集范围。
XMEDIA_TDE_CLIP_MODE_OUTSIDE	仅更新 surface 与 clip_rect 的交集以外的 surface 有效范围。
XMEDIA_TDE_CLIP_MODE_MAX	枚举尾巴。

【注意事项】

无。

【相关数据类型和接口】

[xmedia_tde_operation_attr](#)

xmedia_tde_out_alpha_mode

【说明】

输出结果 alpha 分量来源。

【定义】

```

typedef enum {
    XMEDIA_TDE_OUT_ALPHA_MODE_NORMAL = 0,
    XMEDIA_TDE_OUT_ALPHA_MODE_FROM_BACKGROUND,
    XMEDIA_TDE_OUT_ALPHA_MODE_FROM_FOREGROUND,
    XMEDIA_TDE_OUT_ALPHA_MODE_FROM_GLOBALALPHA,
    XMEDIA_TDE_OUT_ALPHA_MODE_MAX
} xmedia_tde_out_alpha_mode;

```

【成员】

成员	描述
XMEDIA_TDE_OUT_ALPHA_MODE_NORMAL	输出 alpha 来源于运算结果。

成员	描述
XMEDIA_TDE_OUT_ALPHA_MODE_FROM_BACKGROUND	输出 alpha 来源于背景 alpha。
XMEDIA_TDE_OUT_ALPHA_MODE_FROM_FOREGROUND	输出 alpha 来源于前景 alpha。
XMEDIA_TDE_OUT_ALPHA_MODE_FROM_GLOBALALPHA	输出 alpha 来源于全局 alpha。

【注意事项】

- alpha blending 或 rop 运算固定输出结果的 alpha 分量来源于运算结果。
- 如果输入是 YUV 域无 alpha 分量的图像，输出有 alpha 分量的 RGB 域图像，可以通过指定输出 alpha 来源来决定输出 alpha 值。如：YUV to ARGB，可以配置输出 alpha 来源于全局 alpha 并配置全局 alpha 值。或前景是 YUV，背景是 ARGB，做叠加输出，可以配置输出 alpha 来源背景 alpha。

【相关数据类型和接口】

[xmedia_tde_operation_attr](#)

4 错误码

TDE API 错误码如表 4-1 所示。

表4-1 TDE API 错误码

错误代码	宏定义	描述
0x80010001	XMEDIA_ERRCODE_INVALID_PA RAM	无效的参数设置。
0x80010002	XMEDIA_ERRCODE_NULL_PTR	参数中有空指针错误。
0x80010004	XMEDIA_ERRCODE_NOT_INIT	模块未初始化。
0x80010005	XMEDIA_ERRCODE_NOT_SUPP ORT	不支持的操作。
0x80010006	XMEDIA_ERRCODE_TIMEOUT	操作超时。
0x8001000C	XMEDIA_ERRCODE_BUFFER_E MPTY	节点资源用尽。
0x8001000F	XMEDIA_ERRCODE_NOT_PERMI TTED	操作不允许，状态机不对。
0x80010012	XMEDIA_ERRCODE_NOT_READ Y	驱动未加载。

Proc 调试信息

5.1 调试命令

可通过命令 `cat /proc/umap/tde` 查看节点资源和任务配置情况。

5.2 调试信息

图5-1 XM7206V11A 的 proc 信息

[illegible]

【参数说明】	
参数	
module param info (模块参数信息(加载 ko 时决定))	max_job_n
	max_task

【参数说明】

参数	描述
module param info (模块参数信息(加载 ko 时决定))	max_job_num tde 模块最大工作组个数；只能在加载驱动时进行配置。可配范围是[20,400]。 XM7606V13 默认配置 128 个， XM7206V1 系列默认配置 20 个。
	max_task_num tde 模块最大任务个数；只能在加载驱动时进行配置。可配范围是[20,800]。 XM7606V13 默认配置 200 个， XM7206V11A 默认配置 24 个。

参数		描述
global job state (全局实时 job 的状态)	mod_name	tde 模块的使用者： 目前仅支持 USER。
	created	已经创建的工作组个数。
	wait	等待提交硬件的工作组个数。
	busy	硬件正在处理的工作组个数。
	proc	硬件处理完毕的工作组个数。
the newest job info (记录的最新的 8 个 job 的配置)	handle	与工作组唯一对应的句柄 id。
	mod_name	模块名，同上。
	finish_sta	硬件完成状态： 1：正常完成 2：失败 3：被取消
	task_num	对应 handle 的工作组涵盖多少个 task。
	in_size (XM7606V13 有此参数)	对应 handle 的工作组的大致输入数据量。
	out_size (XM7606V13 有此参数)	对应 handle 的工作组的大致输出数据量。
	fg_size_h (XM7206V11A 有此参数)	前景位图有效区域的高度
	fg_size_w (XM7206V11A 有此参数)	前景位图有效区域的宽度
	bg_size_h (XM7206V11A 有此参数)	背景位图有效区域的高度
	bg_size_w (XM7206V11A 有此参数)	背景位图有效区域的宽度

参数		描述
	dst_size_h (XM7206V11A 有此参数)	目标位图有效区域的高度
	dst_size_w (XM7206V11A 有此参数)	目标位图有效区域的宽度
	total_time	从提交工作组到接口返回总耗时(软件+硬件)。
	hw_time	从启动硬件到中断上报耗时(硬件)。
the newest task info(记录的最新的 8 个 job 对应的 task 的具体配置)	handle	句柄 id, 同上。
	line	对应工作组是否有画线任务: 0: 没有 1: 有
	fill	对应工作组是否有填充任务: 0: 没有 1: 有
	copy	对应工作组是否有拷贝任务: 0: 没有 1: 有
	scale	对应工作组是否有缩放任务: 0: 没有 1: 有
	rot	对应工作组是否有旋转任务: 0: 没有 1: 有

参数		描述
	blit1	对应工作组是否有单源搬移任务： 0：没有 1：有
	blit2	对应工作组是否有双源搬移任务： 0：没有 1：有
	mirror	对应工作组是否有镜像任务： 0：没有 1：有
	flip	对应工作组是否有翻转任务： 0：没有 1：有
	blend	对应工作组是否有叠加任务： 0：没有 1：有
	rop	对应工作组是否有布尔运算任务： 0：没有 1：有
	ck	对应工作组是否有关键色任务： 0：没有 1：有

参数		描述
	clip	对应工作组是否有输出剪切任务： 0: 没有 1: 有
abnormal history job info(记录异常 job 信息，历史最新 8 个)	handle	同上。
	mod_name	同上。
	finish_sta	同上。
	task_num	同上。
	in_size (XM7606V13 有此参数)	同上。
	out_size (XM7606V13 有此参数)	同上。
	fg_size_h (XM7206V11A 有此参数)	同上。
	fg_size_w (XM7206V11A 有此参数)	同上。
	bg_size_h (XM7206V11A 有此参数)	同上。
	bg_size_w (XM7206V11A 有此参数)	同上。
	dst_size_h (XM7206V11A 有此参数)	同上。
	dst_size_w (XM7206V11A 有此参数)	同上。
	total_time	同上。
	hw_time	同上。
Abnormal history task	Handle	同上。

参数		描述
info(记录对应的异常 job 的 task 配置信息)	Line	同上。
	fill	同上。
	copy	同上。
	scale	同上。
	rot	同上。
	blit1	同上。
	blit2	同上。
	Mirror	同上。
	flip	同上。
	Blend	同上。
	rop	同上。
	ck	同上。
	clip	同上。

6 实例

6.1 软件流程

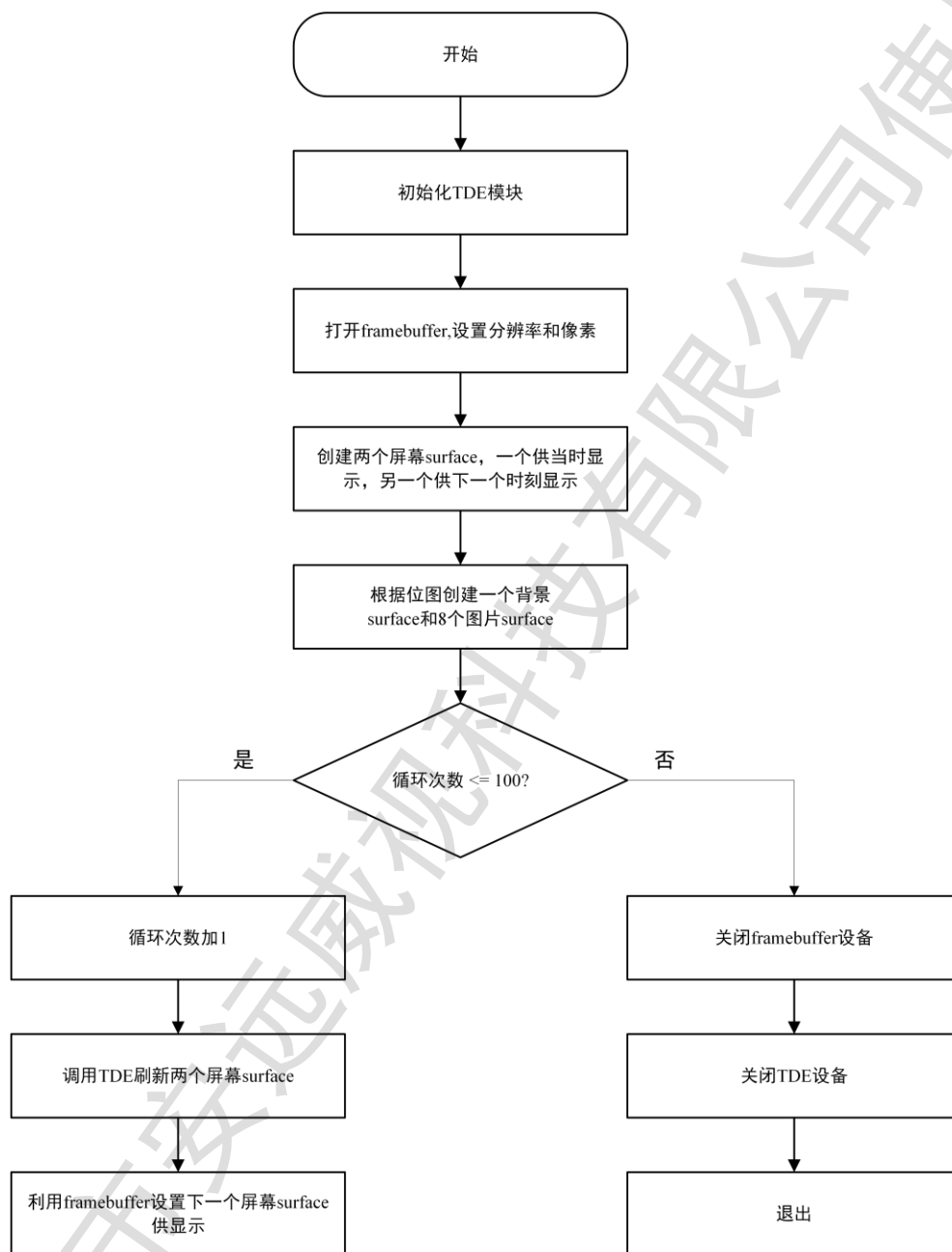


说明

本章以利用 TDE 实现图片动画旋转效果为例。在应用 TDE 前应保证 TDE 和 FB 驱动已经加载，视频输出设备处于工作状态。运行本示例至少需要给叠加图像层 0 分配 1658880 字节的显存，有关 FB 的加载请参见《FB 开发参考》。

该例的软件实现流程如图 6-1 所示。

图6-1 软件实现的流程图（主流程）



6.2 代码参考

详细的代码实现请见 SDK 的 sample 目录下的 tde/sample_tde.c。