



# CV184x TDE 用户使用指南

Version: 1.0.0

Release date: 2025/03

©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

<b>1</b>	<b>声明</b>	<b>2</b>
<b>2</b>	<b>功能概述</b>	<b>3</b>
2.1	目的 . . . . .	3
2.2	术语和定义 . . . . .	3
<b>3</b>	<b>API 参考</b>	<b>4</b>
3.1	API 概述 . . . . .	4
3.2	功能函数参考 . . . . .	4
<b>4</b>	<b>数据类型</b>	<b>14</b>
4.1	映射表 . . . . .	14
4.2	详细描述 . . . . .	14
<b>5</b>	<b>错误码</b>	<b>18</b>

修订记录

Revision	Date	Description
1.0.0	2025/03/05	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 功能概述

TDE 是一个轻量级的 2D 图形 API，用于实现图像的旋转、画线、拷贝操作。

## 2.1 目的

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，主要有旋转、画线、拷贝操作，目前仅支持 ARGB8888 格式图像。

## 2.2 术语和定义

表 2.1: 术语和定义列表

序号	术语	定义说明
1.	TDE	Two Dimensional Engine 二维引擎；本文档中表示硬件上的二维图像渲染引擎。
2.	Job	TDE 软件定义的工作集合。
3.	task	一次操作（旋转、画线、拷贝），代表硬件工作一次，一个 Job 可以包含多个 task。
4.	Rotate	顺时针旋转，支持 90, 270 度。
5.	DrawLine	画线，支持画横线和竖线。
6.	QuickCopy	快速拷贝。

# 3 API 参考

## 3.1 API 概述

该功能模块为用户提供以下 API:

- `CVI_TDE_Open`: 打开 TDE 设备。
- `CVI_TDE_Close`: 关闭 TDE 设备。
- `CVI_TDE_BeginJob`: 创建 1 个 TDE 任务。
- `CVI_TDE_EndJob`: 提交添加操作完成的 TDE 任务。
- `CVI_TDE_WaitAllDone`: 等待 TDE 的所有任务完成。
- `CVI_TDE_CancelJob`: 取消指定的 TDE 任务。
- `CVI_TDE_Rotate`: 向任务中添加旋转操作。
- `CVI_TDE_DrawLine`: 向任务中添加画线操作。
- `CVI_TDE_QuickCopy`: 向任务中添加快速拷贝操作。

## 3.2 功能函数参考

### 3.2.1 `CVI_TDE_Open`

#### 【目的】

打开 TDE 设备。

#### 【语法】

```
CVI_S32 CVI_TDE_Open(CVI_VOID);
```

#### 【描述】

调用此接口打开 TDE 设备。

#### 【参数】

无

**【返回值】**

返回值	描述
CVI_SUCCESS	成功。
错误码	失败，其值是错误码。

**【需求】**

需求	描述
头文件	cvi_tde.h
库文件	libtde.a

**【注意】**

- 在进行 TDE 相关操作前应该首先调用此接口，保证 TDE 设备处于打开状态。
- TDE 设备允许多进程重复打开。

**【举例】**

```
/*declaration*/
CVI_S32 ret = 0;
/* open TDE device*/
ret = CVI_TDE_Open ();
if (ret != CVI_SUCCESS) {
    return -1;
}
/* close TDE device*/
CVI_TDE_Close ();
```

## 3.2.2 CVI\_TDE\_Close

**【目的】**

关闭 TDE 设备，将释放所有 CVI\_TDE\_Open 函数申请的资源。

**【语法】**

```
CVI_VOID CVI_TDE_Close(CVI_VOID);
```

**【描述】**

调用此接口关闭 TDE 设备。

**【参数】**

无

**【返回值】**

无

**【需求】**

需求	描述
头文件	cvi_tde.h
库文件	libtde.a

**【注意】**

调用 CVI\_TDE\_Open 与 CVI\_TDE\_Close 的次数需要对应。

**【举例】**

无

### 3.2.3 CVI\_TDE\_BeginJob

**【目的】**

创建 1 个 TDE 任务。

**【语法】**

```
TDE_HANDLE CVI_TDE_BeginJob(CVI_VOID);
```

**【描述】**

调用此接口创建 1 个 TDE 任务 (Job)。TDE 以任务的形式管理 TDE 命令：1 个 TDE 任务是一系列 TDE 命令的集合，它可以包含 1 个或多个 TDE 操作；一个 TDE 命令对应一个 TDE 操作；成功创建 TDE 任务添加完 TDE 操作后，通过 CVI\_TDE\_EndJob 提交该 Job；同一任务中的 TDE 命令是顺序执行。

**【参数】**

无

**【返回值】**

返回值	描述
句柄	成功
TDE_INVALID_HANDLE	失败

**【需求】**

需求	描述
头文件	cvi_tde.h
库文件	libtde.a

**【注意】**

- 在调用此接口前应确保 TDE 设备处于打开状态。
- 应判断返回值，确保获得 1 个正确的任务句柄。
- TDE 能够缓存的任务数没有限制。
- CVI\_TDE\_BeginJob 必须和 CVI\_TDE\_EndJob 配套使用，否则会造成内存泄漏。



- 注意判断返回值。

**【举例】**

```
/* declaration */
CVI_S32 ret;
TDE_HANDLE handle;
/* open TDE device*/
ret = CVI_TDE_Open ();
if (ret != CVI_SUCCESS) {
    return -1;
}
/* create a TDE job */
handle = CVI_TDE_BeginJob ();
if(handle == TDE_INVALID_HANDLE) {
    CVI_TDE_Close ();
    return -1;
}
/* submit the job */
ret = CVI_TDE_EndJob (handle, TD_FALSE, TD_TRUE, 20);
if(ret != CVI_SUCCESS) {
    CVI_TDE_Close ();
    return -1;
}
/* close TDE device*/
CVI_TDE_Close ();
```

### 3.2.4 CVI\_TDE\_EndJob

**【目的】**

提交已创建的 TDE 任务。

**【语法】**

```
CVI_S32 CVI_TDE_EndJob(TDE_HANDLE s32Handle, CVI_BOOL bSync, CVI_BOOL bBlock, ↵
↵CVI_U32 u32TimeOut);
```

**【描述】**

调用此接口提交该 Job;

同一任务中的 TDE 命令是顺序执行。

- 阻塞

指该函数调用不会立刻返回，只有在以下情况下才会返回：

- TDE 任务中的命令都完成
- 等待超时
- 等待被打断

- 非阻塞

指该函数调用会立刻返回，而不关心 TDE 任务中的命令是否已经完成。

阻塞时可以设置一个最长等待时间，如果等待时间到了，TDE 任务中的命令还没有完成，函数就会提前返回，但是任务中的命令还是会在未来的某个时刻完成。

#### 【参数】

参数名称	描述	输入/输出
s32Handle	TDE 任务句柄。	输入
bSync	保留参数，暂不使用。	输入
bBlock	阻塞标志。CVI_TURE: 阻塞, CVI_FALSE: 非阻塞。	输入
u32TimeOut	超时时间，单位：ms。	输入

#### 【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

#### 【需求】

需求	描述
头文件	cvl_tde.h
库文件	libtde.a

#### 【注意】

- 在调用此接口前应保证调用 CVI\_TDE\_Open 打开 TDE 设备，并且调用 CVI\_TDE\_BeginJob 获得了有效的任务句柄。
- 若设置为阻塞操作，函数超时返回或被中断返回时应该注意：此时 TDE 操作的 API 函数提前返回，但执行的操作仍会完成。
- 提交任务后，此任务对应的 handle 会变为无效，再次提交会出现错误码 CVI\_ERR\_TDE\_UNEXIST。

#### 【举例】

无

### 3.2.5 CVI\_TDE\_WaitAllDone

#### 【目的】

等待 TDE 所有任务完成。

#### 【语法】

```
CVI_S32 CVI_TDE_WaitAllDone(CVI_VOID);
```

#### 【描述】

调用此接口等待 TDE 所有任务完成后返回。

**【参数】**

无

**【返回值】**

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

**【需求】**

需求	描述
头文件	cvi_tde.h
库文件	libtde.a

**【注意】**

此接口为阻塞接口，会阻塞等待所有的 TDE 任务完成。

**【举例】**

无

## 3.2.6 CVI\_TDE\_CancelJob

**【目的】**

取消 TDE 任务及已经成功加入到该任务中的操作。

**【语法】**

```
CVI_S32 CVI_TDE_CancelJob(TDE_HANDLE s32Handle);
```

**【描述】**

向 TDE 任务添加操作时，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任务及其下的所有操作。

**【参数】**

参数名称	描述	输入/输出
s32Handle	TDE 任务句柄。	输入

**【返回值】**

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

**【需求】**

需求	描述
头文件	cvi_tde.h
库文件	libtde.a

**【注意】**

- 在调用此接口前应保证调用 CVI\_TDE\_Open 打开 TDE 设备，并且调用 CVI\_TDE\_BeginJob 获得了有效的任务句柄，否则返回值无效。
- 已经提交的任务不能够再取消。
- 取消后的任务不再有效，故不能再向其添加操作，也不能提交该任务。
- 在向 TDE 任务中添加操作（如操作 A）时出错可以有以下两种处理方式：

忽略出错的操作 A，继续向 TDE 任务中添加其余命令，并提交该任务。若该任务成功执行，则说明所有成功添加的操作都完成了，A 操作因未添加成功而没有执行。

因添加操作 A 出错而取消整个任务，则说明该任务连同其下所有已成功添加的操作都被取消。

**【举例】**

```
/* declaration */
CVI_S32 ret;
TDE_HANDLE handle;
/* open TDE device*/
ret = CVI_TDE_Open ();
if (ret != CVI_SUCCESS) {
    return -1;
}
/* create a TDE job */
handle = CVI_TDE_BeginJob ();
if (handle == TDE_INVALID_HANDLE) {
    CVI_TDE_Close ();
    return -1;
}
/* add task */
TDE_SURFACE_S stSrc;
TDE_SURFACE_S stDst;
ret = CVI_TDE_Rotate(handle, &stSrc, &stDst, TDE_ROTATE_90);
if (ret != CVI_SUCCESS) {
    CVI_TDE_CancelJob(handle);
    CVI_TDE_Close ();
    return -1;
}
/* submit the job */
ret = CVI_TDE_EndJob (handle, TD_FALSE, TD_TRUE, 20);
if (ret != CVI_SUCCESS) {
    CVI_TDE_Close ();
    return -1;
}
/* close TDE device*/
CVI_TDE_Close ();
```

### 3.2.7 CVI\_TDE\_Rotate

#### 【目的】

向任务中添加旋转操作。

#### 【语法】

```
CVI_S32 CVI_TDE_Rotate(TDE_HANDLE s32Handle, TDE_SURFACE_S *pstSrc, TDE_
→SURFACE_S *pstDst, TDE_ROTATE_ANGLE_E enRotateAngle);
```

#### 【描述】

将 pstSrc 中的位图以指定的尺寸旋转，将结果输出到以 pstDst 中指定的目的地址内存中，可以做 90 度，270 度顺时针旋转。

#### 【参数】

参数名称	描述	输入/输出
s32Handle	TDE 任务句柄。	输入
pstSrc	源地址集合。	输入
pstDst	目的地址集合。	输入
enRotateAngle	旋转的角度	输入

#### 【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

#### 【需求】

需求	描述
头文件	cvi_tde.h
库文件	libtde.a

#### 【注意】

- 仅支持 PIXEL\_FORMAT\_ARGB\_8888 的格式。
- 图像宽高都需要 16 像素对齐。
- 位图物理地址需要 16 对齐。
- 旋转 90 度和 270 度，目的图像的宽高和源图像宽高调换。
- 源图像 u32Stride 等于源图像宽度的 4 倍，目的图像 u32Stride 等于目的图像宽度的 4 倍。

#### 【举例】

无

### 3.2.8 CVI\_TDE\_DrawLine

#### 【目的】

向任务中添加画线操作，仅支持画直线。

#### 【语法】

```
CVI_S32 CVI_TDE_DrawLine(TDE_HANDLE s32Handle, TDE_SURFACE_S *pstSrc, TDE_
→SURFACE_S *pstDst, TDE_LINE_S *pstLine);
```

#### 【描述】

将 pstSrc 中的位图画线，将结果输出到以 pstDst 中指定的目的地址内存中，仅支持画横竖直线。

#### 【参数】

参数名称	描述	输入/输出
s32Handle	TDE 任务句柄。	输入
pstSrc	源地址集合。	输入
pstDst	目的地址集合。	输入
pstLine	单一线条属性，参考数据结构 TDE_LINE_S。	输入

#### 【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

#### 【需求】

需求	描述
头文件	cvi_tde.h
库文件	libtde.a

#### 【注意】

- 仅支持 PIXEL\_FORMAT\_ARGB\_8888 的格式。
- 图像宽高都需要 16 像素对齐。
- 位图物理地址需要 16 对齐。
- 目的图像的宽高和源图像宽高相等。
- 源图像 u32Stride 等于源图像宽度的 4 倍，目的图像 u32Stride 等于目的图像宽度的 4 倍。

#### 【举例】

无

### 3.2.9 CVI\_TDE\_QuickCopy

#### 【目的】

向指定任务中添加快速拷贝操作。

#### 【语法】

```
CVI_S32 CVI_TDE_QuickCopy(TDE_HANDLE s32Handle, TDE_SURFACE_S *pstSrc, TDE_
→SURFACE_S *pstDst);
```

#### 【描述】

将 pstSrc 中的位图拷贝到以 pstDst 中指定的目的地址内存中。

#### 【参数】

参数名称	描述	输入/输出
s32Handle	TDE 任务句柄。	输入
pstSrc	源地址集合。	输入
pstDst	目的地址集合。	输入

#### 【返回值】

返回值	描述
CVI_SUCCESS	成功
错误码	失败，其值是错误码

#### 【需求】

需求	描述
头文件	cvi_tde.h
库文件	libtde.a

#### 【注意】

- 仅支持 PIXEL\_FORMAT\_ARGB\_8888 的格式。
- 图像宽高都需要 16 像素对齐。
- 位图物理地址需要 16 对齐。
- 目的图像的宽高和源图像宽高相等。
- 源图像 u32Stride 等于源图像宽度的 4 倍，目的图像 u32Stride 等于目的图像宽度的 4 倍。

#### 【举例】

无

# 4 数据类型

本章列举并描述可配置的数据结构。

## 4.1 映射表

表 4.1: 数据类型映射表

数据类型	说明
TDE_INVALID_HANDLE	无效的句柄
TDE_HANDLE	TDE 任务句柄
TDE_SURFACE_S	位图 surface 结构体
TDE_ROTATE_ANGLE_E	旋转角度枚举
TDE_LINE_S	画线结构体

## 4.2 详细描述

### 4.2.1 TDE\_INVALID\_HANDLE

#### 【说明】

无效的句柄。

#### 【定义】

```
#define TDE_INVALID_HANDLE (-1)
```

#### 【成员】

无

#### 【注意事项】

无

#### 【相关数据类型及接口】

CVI\_TDE\_BeginJob



## 4.2.2 TDE\_HANDLE

### 【说明】

TDE 任务句柄。

### 【定义】

```
typedef CVI_S32 TDE_HANDLE;
```

### 【成员】

无

### 【注意事项】

无

### 【相关数据类型及接口】

无

## 4.2.3 TDE\_SURFACE\_S

### 【说明】

位图 surface 结构体。

### 【定义】

```
typedef struct _TDE_SURFACE_S {  
    CVI_U64 u64PhyAddr; /* Header address of a bitmap or the Y component */  
    CVI_U32 u32Height; /* Bitmap height */  
    CVI_U32 u32Width; /* Bitmap width */  
    CVI_U32 u32Stride; /* Stride of a bitmap */  
    PIXEL_FORMAT_E enColorFmt;  
} TDE_SURFACE_S;
```

### 【成员】

成员	描述
u64PhyAddr	物理地址
u32Height	图像高度 [16, 65532]
u32Width	图像宽度 [16, 65532]
u32Stride	行跨距 [64, 262143]
enColorFmt	像素格式。

### 【注意事项】

- 像素格式仅支持 PIXEL\_FORMAT\_ARGB\_8888 的格式。
- 图像宽高都需要 16 像素对齐。
- 位图物理地址需要 16 对齐。
- u32Stride 一般等于宽度的 4 倍。

## 【相关数据类型及接口】

CVI\_TDE\_Rotate CVI\_TDE\_DrawLine CVI\_TDE\_QuickCopy

## 4.2.4 TDE\_ROTATE\_ANGLE\_E

## 【说明】

旋转角度。

## 【定义】

```
typedef enum _TDE_ROTATE_ANGLE_E {
    TDE_ROTATE_NONE = 0, /* No rotate */
    TDE_ROTATE_90, /* rotate 90 */
    TDE_ROTATE_270, /* rotate 270 */
    TDE_ROTATE_MAX
} TDE_ROTATE_ANGLE_E;
```

## 【成员】

成员	描述
TDE_ROTATE_NONE	不旋转
TDE_ROTATE_90	旋转 90 度
TDE_ROTATE_270	旋转 270 度

## 【注意事项】

- 顺时针旋转。

## 【相关数据类型及接口】

CVI\_TDE\_Rotate

## 4.2.5 TDE\_LINE\_S

## 【说明】

画线结构体。

## 【定义】

```
typedef struct _TDE_LINE_S {
    CVI_S32 s32StartX;
    CVI_S32 s32StartY;
    CVI_S32 s32EndX;
    CVI_S32 s32EndY;
    CVI_U32 u32Thick;
    CVI_U32 u32Color;
} TDE_LINE_S;
```

## 【成员】

成员	描述
s32StartX	线条起点坐标 x
s32StartY	线条起点坐标 y
s32EndX	线条终点坐标 x
s32EndY	线条终点坐标 y
u32Thick	线条宽度
u32Color	线条颜色

**【注意事项】**

- 只能画横线或者竖线。
- 如果画横线, (s32StartX, s32StartY) 表示左上顶点坐标, (s32EndX, s32EndY) 表示右上顶点坐标, 线条向下延伸, 厚度为 u32Thick, 构成一个矩形区域, 那么 s32StartY 和 s32EndY 必须相等。
- 如果画竖线, (s32StartX, s32StartY) 表示左上顶点坐标, (s32EndX, s32EndY) 表示左下顶点坐标, 线条向右延伸, 厚度为 u32Thick, 构成一个矩形区域, 那么 s32StartX 和 s32EndX 必须相等。
- u32Color 颜色为 ARGB8888 格式表示。
- 坐标不能超出图像范围。

**【相关数据类型及接口】**

CVI\_TDE\_DrawLine

# 5 错误码

列举此软件模块接口返回的错误码

表 5.1: API 错误码表

错误代码	宏定义	描述
0xC0258003	CVI_ERR_TDE_ILLEGAL_PARAM	无效参数
0xC0258005	CVI_ERR_TDE_UNEXIST	无效 handle
0xC0258006	CVI_ERR_TDE_NULL_PTR	空指针
0xC0258008	CVI_ERR_TDE_NOT_SUPPORT	操作不支持
0xC0258009	CVI_ERR_TDE_NOT_PERM	操作不允许
0xC025800C	CVI_ERR_TDE_NOMEM	分配内存失败
0xC025800E	CVI_ERR_TDE_BUF_EMPTY	job 中没有操作
0xC025800F	CVI_ERR_TDE_BUF_FULL	job 中 task 已满
0xC0258010	CVI_ERR_TDE_NOTREADY	TDE 没有打开
0xC0258012	CVI_ERR_TDE_BUSY	TDE 硬件忙