

# 视频图形子系统

文件标识: RK-SYS1-MPI-VGS

发布版本: V0.2.1

日期: 2021.4

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

## 免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司 (“本公司”, 下同) 不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

## 商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自拥有者所有。

## 版权所有 © 2021 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: [www.rock-chips.com](http://www.rock-chips.com)

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: [fae@rock-chips.com](mailto:fae@rock-chips.com)

## 前言

### 概述

VGS ( Video Graphics Sub-System) 视频图形子系统, 主要是对输入的图像进行缩放、旋转、打OSD、打COVER、画线等操作。

### 产品版本

| 芯片名称   | 内核版本 |
|--------|------|
| RK356X | 4.19 |

### 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

修订记录

| 版本号    | 作者  | 修改日期      | 修改说明           |
|--------|-----|-----------|----------------|
| v0.1.0 | 黄晓明 | 2021-1-4  | 初始版本           |
| v0.2.0 | 黄晓明 | 2021-1-23 | 完善结构定义，新增马赛克接口 |
| v0.2.1 | 黄晓明 | 2021-4-28 | 添加Crop接口       |

目录

视频图形子系统

目录

基本概念

task

job

HANDLE

功能描述

缩放

旋转

打OSD

打COVER

画线

打mosaic

硬解规格

输入数据

输出数据

分辨率

图像格式对齐说明

API 参考

RK\_MPI\_VGS\_BeginJob

RK\_MPI\_VGS\_EndJob

RK\_MPI\_VGS\_CancelJob

RK\_MPI\_VGS\_AddScaleTask

RK\_MPI\_VGS\_AddCropTask

RK\_MPI\_VGS\_AddDrawLineTask

RK\_MPI\_VGS\_AddCoverTask

RK\_MPI\_VGS\_AddOsdTask

RK\_MPI\_VGS\_AddRotationTask

RK\_MPI\_VGS\_AddMosaicTask

RK\_MPI\_VGS\_AddDrawLineTaskArray

RK\_MPI\_VGS\_AddCoverTaskArray

RK\_MPI\_VGS\_AddOsdTaskArray

RK\_MPI\_VGS\_AddMosaicTaskArray

数据类型

VGS\_HANDLE

VGS\_TASK\_ATTR\_S

VGS\_CROP\_COORDINATE\_E

VGS\_CROP\_INFO\_S

VGS\_DRAW\_LINE\_S

VGS\_COVER\_TYPE\_E

VGS\_ADD\_COVER\_S  
VGS\_COLOR\_REVERT\_MODE\_E  
VGS\_OSD\_REVERT\_S  
VGS\_ADD\_OSD\_S  
VGS\_MOSAIC\_BLK\_SIZE\_E  
VGS\_MOSAIC\_S  
错误码

---

## 基本概念

---

### task

对一幅图像完成具体的一个或多个操作，比如打 OSD、缩放或旋转等。VGS默认的最大的task数为200。

### job

VGS管理task的结构，一个job里可以包含多个task，VGS保证task按照添加到job的顺序一次性提交硬件执行。

VGS默认的最大的job数为100。

### HANDLE

任务句柄，标识一个 job。

## 功能描述

---

VGS的功能有缩放、旋转、打osd、打cover、画线、打mosaic。

### 缩放

VGS 支持对一幅图像进行缩放，最大支持图像宽高放大和缩小32倍。

### 旋转

VGS 支持对一幅图像进行0、90、180、270角度的旋转。

### 打OSD

VGS支持在一幅图像上面叠加一张位图。

### 打COVER

VGS支持对一幅图像进行遮挡操作， 遮挡的形状为矩形。

### 画线

VGS支持对一幅图像进行画线操作。

### 打mosaic

VGS支持对一幅图像进行打mosaic操作。

# 硬解规格

## 输入数据

- ARGB/RGB888/RGB565/RGB4444/RGB5551
- YUV420/YUV422/YVYU422/YVYU420/YUV422SP10bit/YUV420SP10bit

## 输出数据

- ARGB/RGB888/RGB565/RGB4444/RGB5551
- YUV420/YUV422/YUV400/Y4/YVYU422/YVYU420

## 分辨率

- 源的最大分辨率: 8192x8192, 目标分辨率为: 4096x4096

## 图像格式对齐说明

| Format                               | Alignment                    |
|--------------------------------------|------------------------------|
| YUV420/422 YUV400/Y4 YVYU422/YUYV420 | 所有参数均须2对齐                    |
| YUV420/422 10bit                     | width stride须要16对齐， 其余参数须2对齐 |
| RGB888                               | width stride须4对齐             |

## API 参考

该功能模块为用户提供以下 MPI：

- **RK\_MPI\_VGS\_BeginJob**：启动一个 job。
- **RK\_MPI\_VGS\_AddScaleTask**：往一个已经启动的 job 添加缩放 task。
- **RK\_MPI\_VGS\_AddCropTask**：往一个已经启动的 job 添加裁剪 task。
- **RK\_MPI\_VGS\_AddDrawLineTask**：往一个已经启动的 job 添加画线 task。
- **RK\_MPI\_VGS\_AddCoverTask**：往一个已经启动的 job 添加打 COVER task。
- **RK\_MPI\_VGS\_AddOsdTask**：往一个已经启动的 job 添加打 OSD task。
- **RK\_MPI\_VGS\_AddRotationTask**：往一个已经启动的 job 里添加旋转任务。
- **RK\_MPI\_VGS\_AddMosaicTask**：往一个已经启动的 job 里添加打马赛克任务。
- **RK\_MPI\_VGS\_AddDrawLineTaskArray**：往一个已经启动的 job 里添加批量画线的任务。
- **RK\_MPI\_VGS\_AddCoverTaskArray**：往一个已经启动的 job 里添加批量打COVER的任务。
- **RK\_MPI\_VGS\_AddOsdTaskArray**：往一个已经启动的 job 里添加批量打OSD的任务。
- **RK\_MPI\_VGS\_AddMosaicTaskArray**：往一个已经启动的 job 里添加批量打马赛克的任务。

## RK\_MPI\_VGS\_BeginJob

【描述】

启动一个 job。

【语法】

```
RK_S32 RK_MPI_VGS_BeginJob(VGS_HANDLE *phHandle);
```

【参数】

| 参数名称     | 描述            | 输入\输出 |
|----------|---------------|-------|
| phHandle | 返回的job handle | 输出    |

【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

【注意】

- 可一次启动多个 job，但必须判断 RK\_MPI\_VGS\_BeginJob 函数返回成功后才能使用phHandle 返回的 HANLDE。
- phHandle 不能为空指针或非法指针。

【举例】

```
RK_S32 s32Ret = RK_SUCCESS;
VGS_HANDLE hHandle;
VGS_TASK_ATTR_S stTask;
s32Ret = RK_MPI_VGS_BeginJob(&hHandle);
if (s32Ret != RK_SUCCESS)
{
    VGS_ERROR_PROCESS(s32Ret);
}
s32Ret =RK_MPI_VGS_AddScaleTask(hHandle, &stTask, VGS_SCLCOEF_NORMAL);
if (s32Ret != RK_SUCCESS)
{
    RK_MPI_VGS_CancelJob(hHandle);
    VGS_ERROR_PROCESS(s32Ret);
}
s32Ret = RK_MPI_VGS_EndJob(hHandle);
if (s32Ret != RK_SUCCESS)
{
    RK_MPI_VGS_CancelJob(hHandle);
    VGS_ERROR_PROCESS(s32Ret);
}
```

## RK\_MPI\_VGS\_EndJob

【描述】

提交一个 job。

【语法】

```
RK_S32 RK_MPI_VGS_EndJob(VGS_HANDLE hHandle);
```

【参数】

| 参数名称    | 描述                    | 输入\输出 |
|---------|-----------------------|-------|
| hHandle | 表示一个已启动 job 的 HANDLE。 | 输入    |

【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job

## RK\_MPI\_VGS\_CancelJob

【描述】

取消一个 job。

【语法】

```
RK_S32 RK_MPI_VGS_CancelJob(VGS_HANDLE hHandle);
```

【参数】

| 参数名称    | 描述                    | 输入\输出 |
|---------|-----------------------|-------|
| hHandle | 表示一个已启动 job 的 HANDLE。 | 输入    |

【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

【注意】

- hHandle 标识的 job 必须是已经启动的 job。

## RK\_MPI\_VGS\_AddScaleTask

【描述】

往一个已经启动的job里添加缩放task。

【语法】

```
RK_S32 RK_MPI_VGS_AddScaleTask(VGS_HANDLE hHandle, const VGS_TASK_ATTR_S *pstTask, VGS_SCLCOEF_MODE_E enScaleCoefMode);
```

### 【参数】

| 参数名称            | 描述                    | 输入\输出 |
|-----------------|-----------------------|-------|
| hHandle         | 表示一个已启动 job 的 HANDLE。 | 输入    |
| pstTask         | VGS task 属性指针         | 输入    |
| enScaleCoefMode | 缩放系数模式，暂不支持           | 输入    |

### 【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

### 【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job。
- 输入图像的宽高，stride 参考硬解规格中的图像格式对齐说明

### 【举例】

- 参考RK\_MPI\_VGS\_BeginJob的举例

## RK\_MPI\_VGS\_AddCropTask

### 【描述】

往一个已经启动的job里添加缩放task。

### 【语法】

```
RK_S32 RK_MPI_VGS_AddCropTask(VGS_HANDLE hHandle, const VGS_TASK_ATTR_S
*pstTask, const VGS_CROP_INFO_S *pstVgsCrop)
```

### 【参数】

| 参数名称       | 描述                    | 输入\输出 |
|------------|-----------------------|-------|
| hHandle    | 表示一个已启动 job 的 HANDLE。 | 输入    |
| pstTask    | VGS task 属性指针         | 输入    |
| pstVgsCrop | 需要裁剪的区域               | 输入    |

### 【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job。
- 输入图像的宽高，stride 参考硬解规格中的图像格式对齐说明

【举例】

- 参考RK\_MPI\_VGS\_BeginJob的举例

## RK\_MPI\_VGS\_AddDrawLineTask

【描述】

往一个已经启动的job里添加画线task。

【语法】

```
RK_S32 RK_MPI_VGS_AddDrawLineTask(VGS_HANDLE hHandle, const VGS_TASK_ATTR_S *pstTask, const VGS_DRAW_LINE_S *pstVgsDrawLine);
```

【参数】

| 参数名称           | 描述                    | 输入\输出 |
|----------------|-----------------------|-------|
| hHandle        | 表示一个已启动 job 的 HANDLE。 | 输入    |
| pstTask        | VGS task 属性指针         | 输入    |
| pstVgsDrawLine | VGS 画线属性配置指针          | 输入    |

【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job。

【举例】

- 参考RK\_MPI\_VGS\_BeginJob的举例

## RK\_MPI\_VGS\_AddCoverTask

【描述】

往一个已经启动的job里添加打COVER task。

【语法】



```
RK_S32 RK_MPI_VGS_AddCoverTask(  
    VGS_HANDLE hHandle, const VGS_TASK_ATTR_S *pstTask, const  
    VGS_ADD_COVER_S *pstVgsAddCover)
```

#### 【参数】

| 参数名称           | 描述                    | 输入\输出 |
|----------------|-----------------------|-------|
| hHandle        | 表示一个已启动 job 的 HANDLE。 | 输入    |
| pstTask        | VGS task 属性指针         | 输入    |
| pstVgsAddCover | VGS 打 COVER 属性配置指针    | 输入    |

#### 【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

#### 【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job。

#### 【举例】

- 参考RK\_MPI\_VGS\_BeginJob的举例

## RK\_MPI\_VGS\_AddOsdTask

#### 【描述】

往一个已经启动的job里添加打OSD task。

#### 【语法】

```
RK_S32 RK_MPI_VGS_AddOsdTask(VGS_HANDLE hHandle, const VGS_TASK_ATTR_S *pstTask,  
    const VGS_ADD_OSD_S *pstVgsAddOsd)
```

#### 【参数】

| 参数名称         | 描述                    | 输入\输出 |
|--------------|-----------------------|-------|
| hHandle      | 表示一个已启动 job 的 HANDLE。 | 输入    |
| pstTask      | VGS task 属性指针         | 输入    |
| pstVgsAddOsd | VGS 打 OSD 属性配置指针      | 输入    |

#### 【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job。

【举例】

- 参考RK\_MPI\_VGS\_BeginJob的举例

## RK\_MPI\_VGS\_AddRotationTask

【描述】

往一个已经启动的job里添加旋转 task。

【语法】

```
RK_S32 RK_MPI_VGS_AddRotationTask(VGS_HANDLE hHandle, const VGS_TASK_ATTR_S *pstTask, ROTATION_E enRotationAngle)
```

【参数】

| 参数名称            | 描述                    | 输入\输出 |
|-----------------|-----------------------|-------|
| hHandle         | 表示一个已启动 job 的 HANDLE。 | 输入    |
| pstTask         | VGS task 属性指针         | 输入    |
| enRotationAngle | 旋转角度                  | 输入    |

【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job。

【举例】

- 参考RK\_MPI\_VGS\_BeginJob的举例

## RK\_MPI\_VGS\_AddMosaicTask

【描述】

往一个已经启动的job里添加打马赛克 task。

【语法】

```
RK_S32 RK_MPI_VGS_AddMosaicTask(VGS_HANDLE hHandle, const VGS_TASK_ATTR_S
*pstTask, const VGS_MOSAIC_S* pstVgsMosaic)
```

【参数】

| 参数名称         | 描述                    | 输入\输出 |
|--------------|-----------------------|-------|
| hHandle      | 表示一个已启动 job 的 HANDLE。 | 输入    |
| pstTask      | VGS task 属性指针         | 输入    |
| pstVgsMosaic | Mosaic属性配置结构体         | 输入    |

【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job。
- 做mosaic任务的时候，输入和输出的图像为同一块buffer。

【举例】

- 参考RK\_MPI\_VGS\_BeginJob的举例

## RK\_MPI\_VGS\_AddDrawLineTaskArray

【描述】

往一个已经启动的job里添加批量画线task。

【语法】

```
RK_S32 RK_MPI_VGS_AddDrawLineTaskArray(VGS_HANDLE hHandle, const VGS_TASK_ATTR_S
*pstTask,const VGS_DRAW_LINE_S astVgsDrawLine[], RK_U32 u32ArraySize);
```

【参数】

| 参数名称           | 描述                    | 输入\输出 |
|----------------|-----------------------|-------|
| hHandle        | 表示一个已启动 job 的 HANDLE。 | 输入    |
| pstTask        | VGS task 属性指针         | 输入    |
| astVgsDrawLine | VGS 画线属性配置结构体数组       | 输入    |
| u32ArraySize   | VGS 画线数目，范围[1,100]    | 输入    |

#### 【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

#### 【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job。
- 做批量画线任务时，输入和输出的图像为同一块buffer。

#### 【举例】

- 参考RK\_MPI\_VGS\_BeginJob的举例

## RK\_MPI\_VGS\_AddCoverTaskArray

#### 【描述】

往一个已经启动的job里添加批量打COVER task。

#### 【语法】

```
RK_S32 RK_MPI_VGS_AddCoverTaskArray(VGS_HANDLE hHandle, const VGS_TASK_ATTR_S
*pstTask,const VGS_ADD_COVER_S astVgsAddCover[], RK_U32 u32ArraySize)
```

#### 【参数】

| 参数名称           | 描述                    | 输入\输出 |
|----------------|-----------------------|-------|
| hHandle        | 表示一个已启动 job 的 HANDLE。 | 输入    |
| pstTask        | VGS task 属性指针         | 输入    |
| astVgsAddCover | VGS 打 COVER 属性配置结构体数组 | 输入    |
| u32ArraySize   | VGS Cover数目，范围[1,100] | 输入    |

#### 【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job。
- 做批量画线任务时，输入和输出的图像为同一块buffer。

【举例】

- 参考RK\_MPI\_VGS\_BeginJob的举例

## RK\_MPI\_VGS\_AddOsdTaskArray

【描述】

往一个已经启动的job里添加批量打OSD task。

【语法】

```
RK_S32 RK_MPI_VGS_AddOsdTaskArray(VGS_HANDLE hHandle, const VGS_TASK_ATTR_S *pstTask, const VGS_ADD_OSD_S astVgsAddOsd[], RK_U32 u32ArraySize)
```

【参数】

| 参数名称         | 描述                    | 输入\输出 |
|--------------|-----------------------|-------|
| hHandle      | 表示一个已启动 job 的 HANDLE。 | 输入    |
| pstTask      | VGS task 属性指针         | 输入    |
| astVgsAddOsd | VGS 打 OSD 属性配置结构体数组   | 输入    |
| u32ArraySize | VGS OSD数目，范围[1,100]   | 输入    |

【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job。
- 做批量叠加OSD任务时，输入和输出的图像为同一块buffer。

【举例】

- 参考RK\_MPI\_VGS\_BeginJob的举例

# RK\_MPI\_VGS\_AddMosaicTaskArray

【描述】

往一个已经启动的job里添加批量打mosaic task。

【语法】

```
RK_S32 RK_MPI_VGS_AddMosaicTaskArray(VGS_HANDLE hHandle, const VGS_TASK_ATTR_S
*pstTask, const VGS_MOSAIC_S astVgsMosaic[], RK_U32 u32ArraySize)
```

【参数】

| 参数名称         | 描述                     | 输入\输出 |
|--------------|------------------------|-------|
| hHandle      | 表示一个已启动 job 的 HANDLE。  | 输入    |
| pstTask      | VGS task 属性指针          | 输入    |
| astVgsMosaic | VGS 打 mosaic 属性配置结构体数组 | 输入    |
| u32ArraySize | mosaic 数目，范围[1,100]    | 输入    |

【返回值】

| 返回值 | 描述      |
|-----|---------|
| 0   | 成功      |
| 非0  | 失败，见错误码 |

【注意】

- 如果此接口返回失败，必须调用 RK\_MPI\_VGS\_CancelJob 接口取消掉 hHandle 标识的 job。否则会导致 hHandle 标识的 job 不能再被循环利用。
- hHandle 标识的 job 必须是已经启动的 job。
- 做批量mosaic任务时，输入和输出的图像为同一块buffer。

【举例】

- 参考RK\_MPI\_VGS\_BeginJob的举例

## 数据类型

VGS 模块相关数据类型定义如下：

- **VGS\_HANDLE** ：定义 VGS job 的句柄。
- **VGS\_TASK\_ATTR\_S**：定义 VGS task 的属性。
- **VGS\_CROP\_COORDINATE\_E**：定义 VGS 裁剪起始坐标的模式。
- **VGS\_CROP\_INFO\_S**：定义 VGS 裁剪所需要的相关配置。
- **VGS\_DRAW\_LINE\_S**：定义 VGS 画线操作的相关配置。
- **VGS\_COVER\_TYPE\_E**：定义 VGS 上的 COVER 类型。
- **VGS\_ADD\_COVER\_S**：定义 VGS 上 COVER 的配置。
- **VGS\_MOSAIC\_S**：定义 VGS 上 MOSAIC 的配置
- **VGS\_COLOR\_REVERT\_MODE\_E**：定义 VGS 上 OSD 的反色模式
- **VGS\_OSD\_REVERT\_S**：定义 VGS 上 OSD 反色的配置。

- **VGS\_ADD\_OSD\_S**: 定义 VGS 上 OSD 的配置。
- **VGS\_SCLCOEF\_MODE\_E**: 定义 VGS 缩放系数模式的配置。
- **VGS\_MOSAIC\_BLK\_SIZE\_E**: 定义 VGS mosaic 块大小的配置。

## VGS\_HANDLE

### 【说明】

定义 VGS job 的句柄。

### 【定义】

```
typedef RK_S32 VGS_HANDLE
```

### 【注意事项】

无。

### 【相关数据类型及接口】

无。

## VGS\_TASK\_ATTR\_S

### 【说明】

定义 VGS task 的属性。

### 【定义】

```
typedef struct rkVGS_TASK_ATTR_S
{
    VIDEO_FRAME_INFO_S stImgIn; /* Input picture */
    VIDEO_FRAME_INFO_S stImgOut; /* Output picture */
    RK_U64 au64privateData[4]; /* Private data of task*/
    RK_U32 reserved; /* Debug information, state of current picture*/
}VGS_TASK_ATTR_S;
```

### 【成员】

| 参数名称            | 描述                               |
|-----------------|----------------------------------|
| stImgIn         | 输入图像属性                           |
| stImgOut        | 输出图像属性                           |
| au64privateData | 与 task 相关的私有数据，VGS 不会使用和修改其中的数据。 |
| reserved        | 保留项                              |

## VGS\_CROP\_COORDINATE\_E

### 【说明】

定义 VGS 裁剪起始坐标的模式。

### 【定义】

```
typedef enum rkVGS_CROP_COORDINATE_E {
    VGS_CROP_RATIO_COOR = 0, /* Ratio coordinate. */
    VGS_CROP_ABS_COOR /* Absolute coordinate. */
} VGS_CROP_COORDINATE_E;
```

#### 【成员】

| 参数名称                | 描述   |
|---------------------|------|
| VGS_CROP_RATIO_COOR | 相对坐标 |
| VGS_CROP_ABS_COOR   | 绝对坐标 |

#### 【注意事项】

相对坐标，即起始点的坐标值是以与当前图像宽高的比率来表示，使用时需做转换，具体请参见 VGS\_CROP\_INFO\_S。

## VGS\_CROP\_INFO\_S

#### 【说明】

定义 VGS 裁剪所需要的相关配置。

#### 【定义】

```
typedef struct rkVGS_CROP_INFO_S {  
    VGS_CROP_COORDINATE_E    enCropCoordinate;  
    RECT_S                   stCropRect;  
} VGS_CROP_INFO_S;
```

#### 【成员】

| 参数名称             | 描述           |
|------------------|--------------|
| enCropCoordinate | CROP 起始点坐标模式 |
| stCropRect       | CROP 的矩形区域   |

#### 【注意事项】

- 若 enCropCoordinate 为 VPSS\_CROP\_RATIO\_COOR（相对坐标模式），使用 stCropRect 的成员时应做转换，计算公式为：  
 $s32X = \text{起始点坐标} \times \text{原始图像宽度} / 1000$ ，合法取值范围：[0, 999]，计算完成后会进行取整操作和对齐操作。公式同样适用于纵坐标计算。  
 $u32Width = \text{区域宽度} \times \text{实际图像宽度} / 1000$ ，区域宽度取值范围：[1, 1000]。计算完成后会进行取整操作和对齐操作。公式同样适用于区域高度计算。

## VGS\_DRAW\_LINE\_S

#### 【说明】

定义 VGS 画线操作的相关配置。

#### 【定义】

```
typedef struct rkVGS_DRAW_LINE_S {  
    POINT_S                stStartPoint;  
    POINT_S                stEndPoint;  
    RK_U32                 u32Thick;  
    RK_U32                 u32Color;  
} VGS_DRAW_LINE_S;
```



### 【成员】

| 参数名称         | 描述                                      |
|--------------|---|
| stStartPoint | 线的起始点坐标。                                |
| stEndPoint   | 线的结束点坐标。                                |
| u32Thick     | 线的宽度。                                   |
| u32Color     | 线的颜色，RGBA8888 格式，取值范围[0x0, 0xFFFFFFFF]。 |

## VGS\_COVER\_TYPE\_E

### 【说明】

定义 VGS 上的 COVER 类型。

### 【定义】

```
typedef enum rkVGS_COVER_TYPE_E {  
    COVER_RECT = 0,  
    COVER_QUAD_RANGLE,  
    COVER_BUTT  
} VGS_COVER_TYPE_E;
```

### 【成员】

| 参数名称              | 描述           |
|-------------------|--------------|
| COVER_RECT        | 矩形 COVER。    |
| COVER_QUAD_RANGLE | 任意四边形 COVER。 |

## VGS\_ADD\_COVER\_S

### 【说明】

定义 VGS 上 COVER 的配置。

### 【定义】

```
typedef struct rkVGS_ADD_COVER_S {  
    VGS_COVER_TYPE_E          enCoverType;  
    union {  
        RECT_S                stDstRect;  
        VGS_QUADRANGLE_COVER_S stQuadRangle;  
    };  
    RK_U32                    u32Color;  
} VGS_ADD_COVER_S;
```

### 【成员】

| 参数名称        | 描述  |
|-------------|---|
| enCoverType | COVER 类型。                                   |
| stDstRect   | 矩形 COVER 的位置和宽高。                            |
| stQuadRange | 任意四边形 COVER 的相关配置。四点坐标值和边框厚度。               |
| u32Color    | COVER的颜色，RGBA8888 格式，取值范围[0x0, 0xFFFFFFFF]。 |

## VGS\_COLOR\_REVERT\_MODE\_E

### 【说明】

定义 VGS 上 OSD 反色模式的配置。

### 【定义】

```
typedef enum rkVGS_COLOR_REVERT_MODE_E {
    VGS_COLOR_REVERT_NONE = 0,
    VGS_COLOR_REVERT_RGB,
    VGS_COLOR_REVERT_ALPHA,
    VGS_COLOR_REVERT_BOTH,
    VGS_COLOR_REVERT_BUTT
} VGS_COLOR_REVERT_MODE_E;
```

### 【成员】

| 参数名称                   | 描述                |
|------------------------|-------------------|
| VGS_COLOR_REVERT_NONE  | 不反色。              |
| VGS_COLOR_REVERT_RGB   | 仅对 RGB 反色。        |
| VGS_COLOR_REVERT_ALPHA | 仅对 alpha 反色。      |
| VGS_COLOR_REVERT_BOTH  | 对 RGB 和 alpha 反色。 |

## VGS\_OSD\_REVERT\_S

### 【说明】

定义 VGS 上 OSD 反色的配置。

### 【定义】

```
typedef struct rkVGS_OSD_REVERT_S {
    RECT_S                stSrcRect;
    VGS_COLOR_REVERT_MODE_E enColorRevertMode;
} VGS_OSD_REVERT_S;
```

### 【成员】

| 参数名称              | 描述                             |
|-------------------|--------------------------------|
| stSrcRect         | OSD 反色的起始坐标及宽高。位置和宽高值均要求 2 对齐。 |
| enColorRevertMode | OSD 反色模式。                      |

## VGS\_ADD\_OSD\_S

### 【说明】

定义 VGS 上 OSD 的配置。

### 【定义】

```
typedef struct rkVGS_ADD_OSD_S {
    MB_BLK                pMbBlk;
    RECT_S                stRect;
    PIXEL_FORMAT_E        enPixelFormat;
    RK_U32                 u32FgAlpha;
} VGS_ADD_OSD_S;
```

### 【成员】

| 参数名称          | 描述               |
|---------------|------------------|
| stRect        | OSD 的起始坐标及宽高。    |
| enPixelFormat | OSD 的像素格式。       |
| pMbBlk        | OSD 图像的物理地址。     |
| u32FgAlpha    | OSD 的前景 alpha 值。 |

## VGS\_MOSAIC\_BLK\_SIZE\_E

### 【说明】

mosaic 块大小枚举。

### 【定义】

```
typedef enum rkVGS_MOSAIC_BLK_SIZE_E {
    RK_MOSAIC_BLK_SIZE_8 = 8, /* block size 8*8 of MOSAIC */
    RK_MOSAIC_BLK_SIZE_16 = 16, /* block size 16*16 of MOSAIC */
    RK_MOSAIC_BLK_SIZE_32 = 32, /* block size 32*32 of MOSAIC */
    RK_MOSAIC_BLK_SIZE_64 = 64, /* block size 64*64 of MOSAIC */
    RK_MOSAIC_BLK_SIZE_BUT
} VGS_MOSAIC_BLK_SIZE_E;
```

### 【成员】

| 参数名称                  | 描述                |
|-----------------------|-------------------|
| RK_MOSAIC_BLK_SIZE_8  | 8x8 大小的Mosaic块。   |
| RK_MOSAIC_BLK_SIZE_16 | 16x16 大小的Mosaic块。 |
| RK_MOSAIC_BLK_SIZE_32 | 32x32 大小的Mosaic块。 |
| RK_MOSAIC_BLK_SIZE_64 | 64x64 大小的Mosaic块。 |

## VGS\_MOSAIC\_S

### 【说明】

定义 VGS 上 mosaic 的配置。

### 【定义】

```
typedef struct rkVGS_MOSAIC_S {
    VGS_MOSAIC_BLK_SIZE_E enBlkSize;
    RECT_S stDstRect;
} VGS_MOSAIC_S;
```

### 【成员】

| 参数名称      | 描述          |
|-----------|-------------|
| enBlkSize | mosaic 块大小。 |
| stDstRect | 矩形坐标。       |

## 错误码

VGS API 错误码如下

| 错误代码       | 宏定义                      | 描述                          |
|------------|--------------------------|-----------------------------|
| 0xA007800E | RK_ERR_VGS_BUF_EMPTY     | VGS的job,task 或node 节点已经使用完毕 |
| 0xA0078003 | RK_ERR_VGS_ILLEGAL_PARAM | VGS 参数设置无效                  |
| 0xA0078006 | RK_ERR_VGS_NULL_PTR      | 输入参数空指针错误                   |
| 0xA0078008 | RK_ERR_VGS_NOT_SUPPORT   | 操作不支持                       |
| 0xA0078009 | RK_ERR_VGS_NOT_PERMITTED | 操作不允许                       |
| 0xA007800D | RK_ERR_VGS_NOBUF         | 分配内存失败                      |
| 0xA0078006 | RK_ERR_VGS_NULL_PTR      | 输入参数空指针错误                   |
| 0xA0078010 | RK_ERR_VGS_SYS_NOTREADY  | 系统未初始化                      |
| 0xA007800F | RK_ERR_VGS_BUF_FULL      | 没有剩余 BUF                    |