

# RKIVS

API 参考

瑞芯微电子股份有限公司  
图形计算平台中心

## 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，由其各自所有者所有。

## 版权所有 \*\*© 2022 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

---

### 文档适用对象

- 软件开发工程师
- 技术支持工程师

### 适用平台

- RV1103
- RV1106

### 版本记录

版本信息	版本说明	日期	作者
V1.0	初始版本	2022.05.06	李煌

# 目录

- 前言.....3
  - 文档适用对象.....3
  - 适用平台.....3
  - 版本记录.....3
- 1. 概述.....5
  - 1.1. IVS 概述.....5
  - 1.2. IVS 概念.....5
- 2. API 参考.....6
  - RK\_IVS\_MD\_Init..... 7
  - RK\_IVS\_MD\_Exit..... 8
  - RK\_IVS\_MD\_CreateChn..... 9
  - RK\_IVS\_MD\_DestroyChn.....10
  - RK\_IVS\_MD\_GetBg..... 11
  - RK\_IVS\_MD\_Process..... 12
- 3. MD 数据类型..... 15
  - MD\_ALG\_MODE\_E..... 15
  - MD\_ATTR\_S..... 15
- 4. 参考示例..... 17
  - 说明..... 17

## 1. 概述

---

### 1.1. IVS 概述

RKIVS(Rockchip Intelligent Video Surveillance)是瑞芯微媒体处理芯片解决方案中比 IVE(Intelligent Video Engine, 智能加速引擎)更高层次的智能视频监控应用 API。用户基于 IVS 可以快速开发出相关智能应用。当前 IVS 支持的智能应用有: MD (Motion Detection, 移动侦测)。

### 1.2. IVS 概念

- MD 算法

MD 算法包含帧差法(MD\_ALG\_MODE\_REF)和背景法(MD\_ALG\_MODE\_BG)两种。

- 帧差法(MD\_ALG\_MODE\_REF)

直接以用户指定的图像为参考帧, 得出视频侦测分析结果的算法, 称为帧差法。

- 背景法(MD\_ALG\_MODE\_BG)

在 MD 处理的过程中, 将产生当前视频的背景图像。然后以背景图像为参考帧, 得出视频侦测分析结果的算法, 称为背景法。

- 背景更新权重

当 MD 算法选择为背景法时, 每次 MD 处理都会产生静止部分图像, 这部分图像和背景会作一次像素值叠加, 新背景 = (静止部分图像的叠加权重  $\times$  静止部分图像 + 动态部分图像的叠加权重  $\times$  旧背景)  $\gg 16$ 。

## 2. API 参考

---

RKIVS 提供以下功能接口：

- RK\_IVS\_MD\_Init: 初始化。
- RK\_IVS\_MD\_Exit: 退出。
- RK\_IVS\_MD\_CreateChn: 创建 MD 通道。
- RK\_IVS\_MD\_DestroyChn: 销毁 MD 通道。
- RK\_IVS\_MD\_GetBg: 获取背景。
- RK\_IVS\_MD\_Process: 侦测处理。

# RK\_IVS\_MD\_Init

## 【描述】

移动侦测初始化。

## 【语法】

```
RK_S32 RK_IVS_MD_Init();
```

## 【参数】

无

## 【返回值】

返回值	描述
0	成功。
非 0	失败， 参见错误码。

## 【引用】

头文件	ivs_md.h rk_md.h
库文件	libivs.a libive.a

## 【注意】

- 外部应用使用 IVS 一系列调用前，在程序初始化部分调用该接口，以完成 IVS 上下文创建及必要的初始化工作。

- 在结束 IVS 相关功能使用后，必须调用 RK\_IVS\_MD\_Exit 进行反初始化，避免系统资源未被释放。

## RK\_IVS\_MD\_Exit

### 【描述】

移动侦测退出。

### 【语法】

RK\_S32 RK\_IVS\_MD\_Exit();

### 【参数】

无

### 【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

### 【引用】

头文件	ivs_md.h rk_md.h
库文件	libivs.a libive.a



## 【注意】

- 必须先调用 RK\_IVS\_MD\_Init 初始化才能调用此接口退出，否则返回错误。

## RK\_IVS\_MD\_CreateChn

## 【描述】

创建 MD 通道。会根据传入的参数申请一些必须的辅助内存。

## 【语法】

```
RK_S32 RK_IVS_MD_CreateChn(MD_CHN MdChn, MD_ATTR_S *pstMdAttr);
```

## 【参数】

参数名称	描述	输入/输出
MdChn	通道号（暂时没有用处，赋值 0 即可）	输入
pstMdAttr	通道信息指针。 不能为空	输入

## 【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

## 【引用】

头文件	ivs_md.h rk_md.h
库文件	libivs.a libive.a

## 【注意】

- 必须先调用 RK\_IVS\_MD\_Init 初始化才能调用此接口退出，否则返回错误。

## RK\_IVS\_MD\_DestroyChn

## 【描述】

销毁 MD 通道。释放 CreateChn 创建的一些必须的辅助内存。

## 【语法】

```
RK_S32 RK_IVS_MD_DestroyChn(MD_CHN MdChn);
```

## 【参数】

参数名称	描述	输入/输出
MdChn	通道号（暂时没有用处，赋值 0 即可）	输入

## 【返回值】

返回值	描述
0	成功。

非 0	失败，参见错误码。
-----	-----------

**【引用】**

头文件	ivs_md.h rk_md.h
库文件	libivs.a libive.a

**【注意】**

- 必须先调用 RK\_IVS\_MD\_Init 初始化才能调用此接口退出，否则返回错误。
- 必须和 RK\_IVS\_MD\_CreateChn 匹配调用，否则可能造成内存泄漏。

**RK\_IVS\_MD\_GetBg**

**【描述】**

获取移动侦测背景。

**【语法】**

RK\_S32 RK\_IVS\_MD\_GetBg(MD\_CHN MdChn, IVE\_DST\_IMAGE\_S \*pstBg);

**【参数】**

参数名称	描述	输入/输出
MdChn	通道号（暂时没有用处，赋值 0 即可）	输入
pstBg	背景图像指针。	输出

## 【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

## 【引用】

头文件	ivs_md.h rk_md.h
库文件	libivs.a libive.a

## 【注意】

- 必须先调用 RK\_IVS\_MD\_Init 初始化才能调用此接口退出，否则返回错误。
- 必须和 RK\_IVS\_MD\_CreateChn 匹配调用，否则可能造成内存泄漏。

## RK\_IVS\_MD\_Process

### 【描述】

移动侦测处理

### 【语法】

```
RK_S32 RK_IVS_MD_Process(MD_CHN MdChn, IVE_SRC_IMAGE_S *pstCur,  
IVE_SRC_IMAGE_S *pstRef, IVE_DST_IMAGE_S *pstSad,  
IVE_DST_MEM_INFO_S *pstBlob);
```

【参数】

参数名称	描述	输入/输出
MdChn	通道号（暂时没有用处，赋值 0 即可）	输入
pstCur	当前帧图像指针。  不能为空。	输入
pstRef	参考帧图像指针  用于第一帧输入背景或者更新背景用，可以为 NULL	输入
pstSad	Sad 指针（暂不支持）  根据 pstMdAttr->enSadOutCtrl，若需要输出则不能为空	输出
pstBlob	区域信息指针  不能为空	输出

参数名称	支持图像类型	地址对齐	分辨率
pstCur	U8C1	16 byte	64 x 64 ~ 1920 x 1080
pstRef	U8C1	16 byte	64 x 64 ~ 1920 x 1080
pstSad	U8C1	16 byte	根据 pstMdAttr->enSadMode，对应 4x4、8x8、16x16 分块模式，高、宽分别为 pstCur 的 1/4、1/8、1/16。
pstBlob	-	16 byte	-

【返回值】

返回值	描述
-----	----

0	成功。
非 0	失败，参见错误码。

**【引用】**

头文件	ivs_md.h rk_md.h
库文件	libivs.a libive.a

**【注意】**

- 必须先调用 RK\_IVS\_MD\_Init 初始化才能调用此接口退出，否则返回错误。
- 必须先调用 RK\_IVS\_MD\_CreateChn，process 需要辅助内存才能使用。
- 最多输出区域信息个数为 254，区域信息请参见“RKIVE API 参考第 3 章节数据类型中的 IVE\_CCBLOB\_S”。IVE\_CCBLOB\_S 的成员 u16CurAreaThr 是分块后的面积阈值信息。在这里输出的连通区域信息是连续储存。

### 3. MD 数据类型

---

#### MD\_ALG\_MODE\_E

**【描述】**

定义 MD 算法模式

**【定义】**

```
typedef enum hiMD_ALG_MODE_E
{
    MD_ALG_MODE_BG = 0x0,/*Base on background image*/
    MD_ALG_MODE_REF = 0x1,/*Base on reference image*/
    MD_ALG_MODE_BUTT
}MD_ALG_MODE_E;
```

**【成员】**

成员名称	描述
MD_ALOG_MODE_BG	背景法
MD_ALOG_MODE_REF	帧差法

#### MD\_ATTR\_S

**【描述】**

定义 MD 通道属性

**【定义】**

```
typedef struct rkMD_ATTR_S
{
    MD_ALG_MODE_E      enAlgMode;    /*Md algorithm mode*/
    IVE_SAD_MODE_E     enSadMode;    /*Sad mode*/
}
```

```

IVE_SAD_OUT_CTRL_S enSadOutCtrl    /*Sad Output ctrl*/
RK_U32              u32Width;        /*Image width*/
RK_U32              u32Height;       /*Image height*/
RK_U16              u16SadThr        /*Sad thresh*/
IVE_CCL_CTRL_S      stCclCtrl        /*Ccl ctrl*/
IVE_ADD_CTRL_S       stAddCtrl        /*Add ctrl*/
}

```

## 【成员】

成员名称	描述
enAlgMode	算法模式，参见 <a href="#">MD_ALG_MODE_E</a>
enSadMode	Sad 模式，（暂不支持）
enSadOutCtrl	Sad 输出控制（暂不支持）
u32Width	图像宽，必须为宏块宽的整数倍，范围：[64,1920]
u32Height	图像高，必须为宏块高的整数倍，范围：[64,1920]
u16SadThr	Sad 阈值。（暂不支持）
stCclCtrl	Ccl 控制参数，参见“RKIVE API 参考第 2 章数据类型中的 IVE_CCL_CTRL_S”。Ccl 控制参数成员信息都是针对分块后的图。
stAddCtrl	Add 控制参数，参见“RKIVE API 参考第 2 章数据类型中的 IVE_CCL_CTRL_S”。目前仅在帧差法内使用。



## 4. 参考示例

---

### 说明

具体 demo 代码位于 `ivs/tests` 目录，支持 `simulater` 使用 `opencv` 演示效果。

示例 demo 主要包括以下几个部分：

- 初始化：调用 `RK_IVS_MD_Init` 进行初始化。
- 内存分配：调用 `RK_IVS_MD_CreateChn` 设置 MD 通道属性，内部会自动申请必须的辅助内存（通过 MMZ）
- 运行：调用 `RK_IVS_MD_Process` 对当前输入的图像帧进行 MD 处理。并输出 Ccl Blob（连通区域标记）。需要注意的是，第一帧及更新背景的时候，参考帧不能为空。
- 后处理：对连通区域的处理。比如画框等。
- 结束：调用 `RK_IVS_MD_DestroyChn` 释放之前分配的辅助内存，调用 `RK_IVS_MD_Exit` 进行反初始化。