系统控制

文件标识：RK-SYS1-MPI-SYS

发布版本：V0.4.0

日　　期：2021.3

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。



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

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

# 前言

**概述**

系统控制模块实现RK MPI通用功能接口，提供系统相关功能、大块物理内存管理等功能。

**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 |
| --- | --- | --- | --- |
| V0.2.0 | 许丽明 | 2021-01-19 | 初始版本 |
| V0.3.0 | 许丽明 | 2021-02-07 | 增加接口和结构体索引，增加结构体描述 |
| V0.4.0 | 方兴文 | 2021-03-27 | 增加输入流模式配置接口 |
| V0.5.0 | 许丽明 | 2021-05-08 | 增加公共内存池操作接口 |

# 目录

# 功能描述

## 内存缓存池

内存缓存池主要向媒体业务提供内存管理功能，负责内存的分配和回收，充分发挥内存缓存池的作用，让内存资源在各个媒体处理模块中合理使用。

一组大小相同的内存缓存块组成一个内存缓存池。根据业务的不同，申请的缓存池的数量、缓存块的大小和数量不同。

## 系统绑定

RK MPI提供系统绑定接口（RK_MPI_SYS_Bind），即通过数据接收者绑定数据源来建立两者之间的关联关系（只允许数据接收者绑定数据源）。绑定后，数据源生成的数据将自动发送给接收者。

# API 参考

系统控制实现RK MPI系统绑定解绑、创建视频缓存池、提供系统时钟、内存管理等功能。
该功能模块为用户提供以下 API：

- **RK_MPI_SYS_Init**：初始化RK MPI系统。
- **RK_MPI_SYS_Exit**：反初始化RK MPI系统。
- **RK_MPI_SYS_Bind**：数据源到数据接收者绑定。
- **RK_MPI_SYS_UnBind**：数据源到数据接收者解绑定。
- **RK_MPI_SYS_MmzAlloc**：在用户态分配 MMZ 内存。
- **RK_MPI_SYS_MmzAlloc_Cached**：在用户态分配 MMZ 内存，该内存支持 cache 缓存。
- **RK_MPI_SYS_MmzAllocEx**：在用户态分配 MMZ 内存，可配置内存标志，如是否带cache/是否是物理连续内存。
- **RK_MPI_SYS_MmzFree**：在用户态释放 MMZ 内存。
- **RK_MPI_SYS_MmzFlushCache**：刷新 cache 里的内容到内存并且使 cache 里的内容无效。

- RK_MPI_SYS_Malloc：申请malloc内存。
- RK_MPI_SYS_Free：释放malloc申请的内存。
- RK_MPI_SYS_CreateMB：创建一个内存缓存快。
- RK_MPI_SYS_GetCurPTS：获取当前时间戳。
- RK_MPI_SYS_InitPTSBase：初始化 RK MPI 时间戳。
- RK_MPI_SYS_SyncPTS：同步 RK MPI 时间戳。
- RK_MPI_MB_CreatePool：创建一个内存缓存池。
- RK_MPI_MB_DestroyPool：销毁一个内存缓存池。
- RK_MPI_MB_GetMB：获取一个缓存块。
- RK_MPI_MB_ReleaseMB：释放一个已经获取的缓存块。
- RK_MPI_MB_Handle2PhysAddr：获取一个缓存块的物理地址。
- RK_MPI_MB_Handle2VirAddr：获取一个内存缓存池中的缓存块的用户态虚拟地址。
- RK_MPI_MB_Handle2Fd：用户态通过缓存块的句柄。
- RK_MPI_MB_Handle2PoolId：获取一个缓存块所在缓存池的 ID。
- RK_MPI_MB_GetSize：获取一个缓存块的大小。
- RK_MPI_MB_VirAddr2Handle：根据虚拟地址获取对应的缓存块。
- RK_MPI_MB_InquireUserCnt：查询缓存块使用计数信息。
- RK_MPI_MB_SetModPoolConfig：设置模块公共视频缓存池属性。
- RK_MPI_MB_GetModPoolConfig：获取模块公共视频缓存池属性。
- RK_MPI_MB_SetBufferStride：设置缓存块用于存储图像时该图像的虚宽高Stride。

## RK_MPI_SYS_Init

【描述】
初始化RK MPI系统。
【语法】

RK_S32 RK_MPI_SYS_Init(RK_VOID);

【参数】

无

【返回值】

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

【注意】

- 运行RK MPI系统前必须调用该接口进行初始化。

## RK_MPI_SYS_Exit

【描述】
反初始化RK MPI系统。
【语法】

RK_S32 RK_MPI_SYS_Exit(RK_VOID);

【参数】

无

【返回值】

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

【注意】

- 退出RK MPI系统前必须调用该函数。

## RK_MPI_SYS_Bind

【描述】
数据源到数据接收者绑定的接口。
【语法】

RK_S32 RK_MPI_SYS_Bind(const MPP_CHN_S *pstSrcChn, const MPP_CHN_S *pstDestChn);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pstSrcChn | 源通道指针。 | 输入 |
| pstDestChn | 目的通道指针。 | 输入 |

【返回值】

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

【注意】

- 同一个数据接收者只能绑定一个数据源。
- 绑定是指数据源和数据接收者建立关联关系。绑定后，数据源生成的数据将自动发送给接收者。
- VPSS 作为数据接收者时，是以设备（GROUP）为接收者，接收其他模块发来的数据，用户将通道号置为 0。

## RK_MPI_SYS_UnBind

【描述】
数据源到数据接收者解绑定。
【语法】

RK_S32 RK_MPI_SYS_UnBind(const MPP_CHN_S *pstSrcChn, const MPP_CHN_S *pstDestChn);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pstSrcChn | 源通道指针。 | 输入 |
| pstDestChn | 目的通道指针。 | 输入 |

【返回值】

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

【注意】

- 无

## RK_MPI_SYS_MmzAlloc

【描述】
在用户态分配 MMZ 内存。
【语法】

RK_S32 RK_MPI_SYS_MmzAlloc(MB_BLK *pBlk, const RK_CHAR *pstrMmb, const RK_CHAR *pstrZone, RK_U32 u32Len);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pBlk | 缓存块指针。 | 输出 |
| pstrMmb | Mmb 名称的字符串指针。填写RK_NULL。 | 输入 |
| pstrZone | MMZ zone 名称的字符串指针。填写RK_NULL。 | 输入 |
| u32Len | 缓存块大小。 | 输入 |

【返回值】

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

【注意】

- 无

## RK_MPI_SYS_MmzAlloc_Cached

【描述】
在用户态分配 MMZ 内存，该内存支持 cache 缓存。
【语法】

RK_S32 RK_MPI_SYS_MmzAlloc_Cached(MB_BLK *pBlk, const RK_CHAR *pstrMmb, const RK_CHAR *pstrZone, RK_U32 u32Len);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pBlk | 缓存块指针。 | 输出 |
| pstrMmb | Mmb 名称的字符串指针。填写RK_NULL。 | 输入 |
| pstrZone | MMZ zone 名称的字符串指针。填写RK_NULL。 | 输入 |
| u32Len | 缓存块大小。 | 输入 |

【返回值】

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

【注意】

- 无

## RK_MPI_SYS_MmzAllocEx

【描述】
在用户态分配 MMZ 内存，可配置内存标志，如是否带cache/是否是物理连续内存。
【语法】

RK_S32 RK_MPI_SYS_MmzAllocEx(MB_BLK *pBlk, const RK_CHAR *pstrMmb, const RK_CHAR *pstrZone, RK_U32 u32Len, RK_U32 u32HeapFlags);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pBlk | 缓存块指针。 | 输出 |
| pstrMmb | Mmb 名称的字符串指针。填写RK_NULL。 | 输入 |
| pstrZone | MMZ zone 名称的字符串指针。填写RK_NULL。 | 输入 |
| u32Len | 缓存块大小。 | 输入 |
| u32HeapFlags | 内存标志，可同时带上多个标记。当前支持如下标志：<br>MB_REMAP_MODE_NOCACHE：不支持Cache缓存<br>MB_REMAP_MODE_CACHED：支持Cache缓存<br>MB_DMA_TYPE_CMA：带上此标志表示申请物理连续内存，物理连续内存需要内核提前配置预留媒体专用内存，当前默认SDK是不支持。 | 输入 |

【返回值】

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

【注意】

- 无

## RK_MPI_SYS_MmzFree

【描述】
在用户态释放 MMZ 内存。
【语法】

RK_S32 RK_MPI_SYS_MmzFree(MB_BLK blk);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| blk | 缓存块ID。 | 输入 |

【返回值】

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

【注意】

- 无

## RK_MPI_SYS_MmzFlushCache

【描述】
刷新 cache 里的内容到内存并且使 cache 里的内容无效。
【语法】

RK_S32 RK_MPI_SYS_MmzFlushCache(MB_BLK blk, RK_BOOL bReadOnly);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| blk | 缓存块ID。 | 输入 |
| bReadOnly | True: 使 cache 里的内容无效。等同于DMA_BUF_SYNC_READ；<br>False: 将cache内容回写内存并使cache无效。<br>DMA_BUF_SYNC_RW | 输入 |

【返回值】

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

【注意】

- 无

## RK_MPI_SYS_Malloc

【描述】
申请malloc内存。
【语法】

RK_S32 RK_MPI_SYS_Malloc(MB_BLK *pBlk, RK_U32 u32Len);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pBlk | 缓存块指针。 | 输出 |
| u32Len | 缓存块大小。 | 输入 |

【返回值】

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

【注意】

- 无。

## RK_MPI_SYS_Free

【描述】
释放malloc申请的内存。
【语法】

RK_S32 RK_MPI_SYS_Free(MB_BLK blk);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| blk | 缓存块ID。 | 输入 |

【返回值】

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

【注意】

- 无

## RK_MPI_SYS_CreateMB

【描述】
创建一个内存缓存快。
【语法】

RK_S32 RK_MPI_SYS_CreateMB(MB_BLK *pBlk, MB_EXT_CONFIG_S *pstMbExtConfig);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pBlk | 缓存块指针。 | 输出 |
| pstMbExtConfig | 缓存块参数配置信息 | 输入 |

【返回值】

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

【注意】

- 无

## RK_MPI_SYS_GetCurPTS

【描述】
获取当前时间戳。
【语法】

RK_S32 RK_MPI_SYS_GetCurPTS(RK_U64 *pu64CurPTS);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pu64CurPTS | 当前时间戳指针。单位：微秒。 | 输出 |

【返回值】

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

【注意】

- 无

## RK_MPI_SYS_InitPTSBase

【描述】
初始化 RK MPI 时间戳。
【语法】

RK_S32 RK_MPI_SYS_InitPTSBase(RK_U64 u64PTSBase);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| u64PTSBase | 时间戳基准。单位：微秒。 | 输入 |

【返回值】

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

【注意】

- 初始化时间戳基准会将当前系统的时间戳强制置成 u64PTSBase，与系统原有时间戳没有任何约束。因此，建议在媒体业务没有启动时（例如操作系统刚启动），调用这个接口。

## RK_MPI_SYS_SyncPTS

【描述】
同步 RK MPI 时间戳。
【语法】

RK_S32 RK_MPI_SYS_SyncPTS(RK_U64 u64PTSBase);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| u64PTSBase | 时间戳基准。单位：微秒。 | 输入 |

【返回值】

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

【注意】

- 无

## RK_MPI_SYS_SetChnInputMode

【描述】
设置通道输入(接收)流模式。用户可根据需要配置通道输入(接收)流模式，可选择保留最新一帧，也可选择丢弃。当用户启用某模块通道时，没有绑定下级模块也没有消耗通道数据，将导致buffer无法正常轮转使用，此时可将此通道输入流模式配置为丢弃模式，保证通道buffer可正常轮转。
【语法】

RK_S32 RK_MPI_SYS_SetChnInputMode(const MPP_CHN_S *pstChn, CHN_INPUT_MODE_E mode);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pstChn | 通道指针 | 输入 |
| mode | 通道输入流模式 | 输入 |

【返回值】

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

【注意】

- 无

## RK_MPI_MB_CreatePool

【描述】
创建一个内存缓存池。
【语法】

MB_POOL RK_MPI_MB_CreatePool(MB_POOL_CONFIG_S *pstMbPoolCfg);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pstMbPoolCfg | 缓存池配置属性参数指针。 | 输入 |

【返回值】

| 返回值 | 描述 |
| --- | --- |
| 非 MB_INVALID_POOLID | 有效的缓存池 ID 号。 |
| MB_INVALID_POOLID | 创建缓存池失败，可能是参数非法或者保留内存不足。 |

【注意】

- 无

## RK_MPI_MB_DestroyPool

【描述】
销毁一个内存缓存池。
【语法】

RK_S32 RK_MPI_MB_DestroyPool(MB_POOL pool);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pool | 缓存池 ID 号。<br>取值范围：[0, MB_MAX_POOLS)。 | 输入 |

【返回值】

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

【注意】

- GROUP 必须已创建。

## RK_MPI_MB_GetMB

【描述】
获取一个缓存块。
【语法】

MB_BLK RK_MPI_MB_GetMB(MB_POOL pool, RK_U64 u64Size, RK_BOOL block = RK_FALSE);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pool | 缓存池ID号。<br>取值范围：[0, MB_MAX_POOLS)。 | 输入 |
| u64Size | 缓存块大小。<br>取值范围：数据类型全范围，以 byte 为单位。 | 输入 |
| block | 获取缓存块是否等待缓存块使用完毕返回缓存池。<br>RK_TRUE：阻塞。RK_FALSE：非阻塞。 默认非阻塞 | 输入 |

【返回值】

| 返回值 | 描述 |
| --- | --- |
| 非 MB_INVALID_HANDLE | 有效的缓存块句柄。 |
| MB_INVALID_HANDLE | 获取缓存块失败。 |

【注意】

- 无

## RK_MPI_MB_ReleaseMB

【描述】
释放一个已经获取的缓存块。
【语法】

RK_S32 RK_MPI_MB_ReleaseMB(MB_BLK mb);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| mb | 缓存块ID。 | 输入 |

【返回值】

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

【注意】

- 无

## RK_MPI_MB_Handle2PhysAddr

【描述】
获取一个缓存块的物理地址。
【语法】

RK_U64 RK_MPI_MB_Handle2PhysAddr(MB_BLK mb);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| mb | 缓存块ID。 | 输入 |

【返回值】

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

【注意】

- 暂不支持。

## RK_MPI_MB_Handle2VirAddr

【描述】
获取一个内存缓存池中的缓存块的用户态虚拟地址。
【语法】

RK_VOID *RK_MPI_MB_Handle2VirAddr(MB_BLK mb);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| mb | 缓存块ID。 | 输入 |

【返回值】

| 返回值 | 描述 |
| --- | --- |
| RK_NULL | 获取虚拟地址失败。 |
| 非RK_NULL | 有效的虚拟地址。 |

【注意】

- 无

## RK_MPI_MB_Handle2Fd

【描述】
用户态通过缓存块的句柄。
【语法】

RK_S32 RK_MPI_MB_Handle2Fd(MB_BLK mb);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| mb | 缓存块ID。 | 输入 |

【返回值】

| 返回值 | 描述 |
| --- | --- |
| 负数 | 获取缓存块句柄失败。 |
| 非负数 | 有效的缓存块句柄。 |

【注意】

- 缓存块句柄只有内存类型为DMA时才可获取有效的句柄。

## RK_MPI_MB_Handle2PoolId

【描述】
获取一个缓存块所在缓存池的 ID。
【语法】

MB_POOL RK_MPI_MB_Handle2PoolId(MB_BLK mb);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| mb | 缓存块ID。 | 输入 |

【返回值】

| 返回值 | 描述 |
| --- | --- |
| 非MB_INVALID_POOLID | 有效的缓存池 ID 号。 |
| MB_INVALID_POOLID | 无效的缓存池 ID 号。 |

【注意】

- 指定的缓存块应该是从RK MPI内存缓存池中获取的有效缓存块，否则无法获取到缓存池ID号。

## RK_MPI_MB_GetSize

【描述】
获取一个缓存块的大小。
【语法】

RK_U64 RK_MPI_MB_GetSize(MB_BLK mb);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| mb | 缓存块ID。 | 输入 |

【返回值】

| 返回值 | 描述 |
| --- | --- |
| 任意值 | 缓存块的大小。 |

【注意】

- 无

## RK_MPI_MB_VirAddr2Handle

【描述】
根据虚拟地址获取对应的缓存块。
【语法】

MB_BLK RK_MPI_MB_VirAddr2Handle(RK_VOID *pstVirAddr);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| pstVirAddr | 虚拟地址。 | 输入 |

【返回值】

| 返回值 | 描述 |
| --- | --- |
| MB_INVALID_HANDLE | 获取缓存块ID失败。 |
| 非MB_INVALID_HANDLE | 有效的缓存块ID。 |

【注意】

- 目前仅支持DMA内存从虚拟地址查找到缓存块。

## RK_MPI_MB_InquireUserCnt

【描述】
查询缓存块使用计数信息。
【语法】

RK_S32 RK_MPI_MB_InquireUserCnt(MB_BLK mb);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| mb | 缓存块ID。 | 输入 |

【返回值】

| 返回值 | 描述 |
| --- | --- |
| 非负数 | 获取缓存块的使用计数。 |
| 负数 | 无效的缓存块计数。 |

【注意】

- 无

## RK_MPI_MB_SetModPoolConfig

【描述】
设置模块公共视频缓存池属性。
【语法】

RK_S32 RK_MPI_MB_SetModPoolConfig(MB_UID_E enMbUid, const MB_CONFIG_S *pstMbConfig);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| enMbUid | 使用模块公共视频缓冲池的模块 ID。 | 输入 |
| pstMbConfig | 模块公共视频缓存池属性指针。 | 输入 |

【返回值】

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

【注意】

- 需要在调用RK_MPI_SYS_Init前设置该属性。

## RK_MPI_MB_GetModPoolConfig

【描述】
获取模块公共视频缓存池属性。
【语法】

RK_S32 RK_MPI_MB_GetModPoolConfig(MB_UID_E enMbUid, MB_CONFIG_S *pstMbConfig);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| enMbUid | 使用模块公共视频缓冲池的模块 ID。 | 输入 |
| pstMbConfig | 模块公共视频缓存池属性指针。 | 输出 |

【返回值】

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

【注意】

- 无

## RK_MPI_MB_SetBufferStride

【描述】
设置缓存块用于存储图像时该图像的虚宽高Stride。
【语法】

RK_S32 RK_MPI_MB_SetBufferStride(MB_BLK mb, RK_U32 u32HorStride, RK_U32 u32VerStride);

【参数】

| 参数名 | 描述 | 输入/输出 |
| --- | --- | --- |
| mb | 缓存块ID。 | 输入 |
| u32HorStride | 图像虚宽，以字节为单位。 | 输入 |
| u32VerStride | 图像虚高，以字节为单位。 | 输入 |

【返回值】

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

【注意】

- 无

# 数据类型

基本数据类型定义如下：

## 公共数据类型

```
typedef unsigned char           RK_UCHAR;
typedef uint8_t                 RK_U8;
typedef uint16_t                RK_U16;
typedef uint32_t                RK_U32;
typedef uint32_t                RK_UL;
typedef uintptr_t               RK_UINTPTR_T;

typedef char                    RK_CHAR;
typedef int8_t                  RK_S8;
typedef int16_t                 RK_S16;
typedef int32_t                 RK_S32;
typedef int32_t                 RK_SL;

typedef float                   RK_FLOAT;
typedef double                  RK_DOUBLE;

typedef uint64_t                RK_U64;
typedef int64_t                 RK_S64;

typedef uint32_t                RK_SIZE_T;
typedef uint32_t                RK_LENGTH_T;

typedef unsigned int            RK_HANDLE;

typedef enum {
    RK_FALSE = 0,
    RK_TRUE  = 1,
} RK_BOOL;

#ifndef NULL
    #define NULL                0L
#endif

#define RK_NULL                 0L
#define RK_SUCCESS              0
#define RK_FAILURE              (-1)

#define RK_VOID                 void
```

## MOD_ID_E

【说明】
定义模块 ID 枚举类型。
【定义】

```
typedef enum rkMOD_ID_E {
    RK_ID_CMPI    = 0,
```

```
    RK_ID_MB       = 1,
    RK_ID_SYS      = 2,
    RK_ID_RGN      = 3,
    RK_ID_VENC     = 4,
    RK_ID_VDEC     = 5,
    RK_ID_VPSS     = 6,
    RK_ID_VGS      = 7,
    RK_ID_VI       = 8,
    RK_ID_VO       = 9,
    RK_ID_AI       = 10,
    RK_ID_AO       = 11,
    RK_ID_AENC     = 12,
    RK_ID_ADEC     = 13,
    RK_ID_TDE      = 14,
    RK_ID_ISP      = 15,

    RK_ID_BUTT,
} MOD_ID_E;
```

【注意事项】
无

## MB_UID_E

【说明】
定义媒体内存缓冲池的模块 ID 枚举类型。
【定义】

```
typedef enum rkMB_UID_E {
    MB_UID_VI = 0,
    MB_UID_VO = 1,
    MB_UID_VGS = 2,
    MB_UID_VENC = 3,
    MB_UID_VDEC = 4,
    MB_UID_VPSS = 5,
    MB_UID_AI = 6,
    MB_UID_AENC = 7,
    MB_UID_ADEC = 8,
    MB_UID_BUTT = 9
} MB_UID_E;
```

【注意事项】
无

## RK_CODEC_ID_E

【说明】
定义音视频编码类型枚举。
【定义】

```
typedef enum rkCODEC_ID_E {
     RK_VIDEO_ID_Unused,             /**< Value when coding is N/A */
     RK_VIDEO_ID_AutoDetect,         /**< Autodetection of coding type */
     RK_VIDEO_ID_MPEG1VIDEO,
     RK_VIDEO_ID_MPEG2VIDEO,         /**< AKA: H.262 */
     RK_VIDEO_ID_H263,               /**< H.263 */
```

```
    RK_VIDEO_ID_MPEG4,              /**< MPEG-4 */
    RK_VIDEO_ID_WMV,                /**< Windows Media Video (WMV1,WMV2,WMV3)*/
    RK_VIDEO_ID_RV,                 /**< all versions of Real Video */
    RK_VIDEO_ID_AVC,                /**< H.264/AVC */
    RK_VIDEO_ID_MJPEG,              /**< Motion JPEG */
    RK_VIDEO_ID_VP8,                /**< VP8 */
    RK_VIDEO_ID_VP9,                /**< VP9 */
    RK_VIDEO_ID_HEVC,               /**< ITU H.265/HEVC */
    RK_VIDEO_ID_DolbyVision,        /**< Dolby Vision */
    RK_VIDEO_ID_ImageHEIC,          /**< HEIF image encoded with HEVC */
    RK_VIDEO_ID_VC1 = 0x01000000,   /**< Windows Media Video (WMV1,WMV2,WMV3)*/
    RK_VIDEO_ID_FLV1,               /**< Sorenson H.263 */
    RK_VIDEO_ID_DIVX3,              /**< DIVX3 */
    RK_VIDEO_ID_VP6,
    RK_VIDEO_ID_AVSPLUS,            /**< AVS+ profile=0x48 */
    RK_VIDEO_ID_AVS,                /**< AVS  profile=0x20 */
    /* *< Reserved region for introducing Khronos Standard Extensions */
    RK_VIDEO_ID_KhronosExtensions = 0x2F000000,
    /* *< Reserved region for introducing Vendor Extensions */
    RK_VIDEO_ID_VendorStartUnused = 0x3F000000,
    RK_VIDEO_ID_Max = 0x3FFFFFFF,

    RK_AUDIO_ID_Unused = 0x40000000,  /**< Placeholder value when coding is N/A
*/
    RK_AUDIO_ID_AutoDetect,  /**< auto detection of audio format */
    RK_AUDIO_ID_PCM_ALAW,    /** <g711a> */
    RK_AUDIO_ID_PCM_MULAW,   /** <g711u> */
    RK_AUDIO_ID_PCM_S16LE,   /**< Any variant of PCM_S16LE coding */
    RK_AUDIO_ID_PCM_S24LE,   /**< Any variant of PCM_S24LE coding */
    RK_AUDIO_ID_PCM_S32LE,   /**< Any variant of PCM_S32LE coding */
    RK_AUDIO_ID_ADPCM_G722,         /**< Any variant of ADPCM_G722 encoded data
*/
    RK_AUDIO_ID_ADPCM_G726,         /**< Any variant of ADPCM_G726 encoded data
*/
    RK_AUDIO_ID_ADPCM_IMA_QT,       /**< Any variant of ADPCM_IMA encoded data
*/
    RK_AUDIO_ID_AMR_NB,      /**< Any variant of AMR_NB encoded data */
    RK_AUDIO_ID_AMR_WB,      /**< Any variant of AMR_WB encoded data */
    RK_AUDIO_ID_GSMFR,       /**< Any variant of GSM fullrate (i.e. GSM610) */
    RK_AUDIO_ID_GSMEFR,      /**< Any variant of GSM Enhanced Fullrate encoded
data*/
    RK_AUDIO_ID_GSMHR,       /**< Any variant of GSM Halfrate encoded data */
    RK_AUDIO_ID_PDCFR,       /**< Any variant of PDC Fullrate encoded data */
    RK_AUDIO_ID_PDCEFR,      /**< Any variant of PDC Enhanced Fullrate encoded
data */
    RK_AUDIO_ID_PDCHR,       /**< Any variant of PDC Halfrate encoded data */
    RK_AUDIO_ID_TDMAFR,      /**< Any variant of TDMA Fullrate encoded data
(TIA/EIA-136-420) */
    RK_AUDIO_ID_TDMAEFR,     /**< Any variant of TDMA Enhanced Fullrate encoded
data (TIA/EIA-136-410) */
    RK_AUDIO_ID_QCELP8,      /**< Any variant of QCELP 8kbps encoded data */
    RK_AUDIO_ID_QCELP13,     /**< Any variant of QCELP 13kbps encoded data */
    RK_AUDIO_ID_EVRC,        /**< Any variant of EVRC encoded data */
    RK_AUDIO_ID_SMV,         /**< Any variant of SMV encoded data */
    RK_AUDIO_ID_G729,        /**< Any variant of G.729 encoded data */
    RK_AUDIO_ID_AAC,         /**< Any variant of AAC encoded data */
    RK_AUDIO_ID_MP3,         /**< Any variant of MP3 encoded data */
    RK_AUDIO_ID_SBC,         /**< Any variant of SBC encoded data */
```

```
    RK_AUDIO_ID_VORBIS,      /**< Any variant of VORBIS encoded data */
    RK_AUDIO_ID_WMA,         /**< Any variant of WMA encoded data */
    RK_AUDIO_ID_RA,          /**< Any variant of RA encoded data */
    RK_AUDIO_ID_MIDI,        /**< Any variant of MIDI encoded data */
    RK_AUDIO_ID_FLAC,        /**< Any variant of FLAC encoded data */
    RK_AUDIO_ID_APE = 0x50000000,
    /**< Reserved region for introducing Khronos Standard Extensions */
    RK_AUDIO_CodingKhronosExtensions = 0x6F000000,
    /**< Reserved region for introducing Vendor Extensions */
    RK_AUDIO_CodingVendorStartUnused = 0x7F000000,
    RK_AUDIO_ID_WMAV1,
    RK_AUDIO_ID_WMAV2,
    RK_AUDIO_ID_WMAPRO,
    RK_AUDIO_ID_WMALOSSLESS,
    RK_AUDIO_ID_MP1,
    RK_AUDIO_ID_MP2,
    /**< add audio bitstream Codec ID define for RT> */
    RK_AUDIO_ID_DTS,
    RK_AUDIO_ID_AC3,
    RK_AUDIO_ID_EAC3,
    RK_AUDIO_ID_DOLBY_TRUEHD,
    RK_AUDIO_ID_MLP,
    RK_AUDIO_ID_DTS_HD,
    RK_AUDIO_CodingMax = 0x7FFFFFFF,

    /* subtitle codecs */
    RK_SUB_ID_Unused = 0x17000,          ///< A dummy ID pointing at the start
of subtitle codecs.
    RK_SUB_ID_DVD,
    RK_SUB_ID_DVB,
    RK_SUB_ID_TEXT,  ///< raw UTF-8 text
    RK_SUB_ID_XSUB,
    RK_SUB_ID_SSA,
    RK_SUB_ID_MOV_TEXT,
    RK_SUB_ID_HDMV_PGS,
    RK_SUB_ID_DVB_TELETEXT,
    RK_SUB_ID_SRT,

    RK_SUB_ID_MICRODVD   = 0x17800,
    RK_SUB_ID_EIA_608,
    RK_SUB_ID_JACOSUB,
    RK_SUB_ID_SAMI,
    RK_SUB_ID_REALTEXT,
    RK_SUB_ID_STL,
    RK_SUB_ID_SUBVIEWER1,
    RK_SUB_ID_SUBVIEWER,
    RK_SUB_ID_SUBRIP,
    RK_SUB_ID_WEBVTT,
    RK_SUB_ID_MPL2,
    RK_SUB_ID_VPLAYER,
    RK_SUB_ID_PJS,
    RK_SUB_ID_ASS,
    RK_SUB_ID_HDMV_TEXT,
    RK_SUB_CodingMax
} RK_CODEC_ID_E;
```

【注意事项】
无

## ROTATION_E

【说明】
定义旋转角度枚举。
【定义】

```
typedef enum rkROTATION_E {
    ROTATION_0          = 0,
    ROTATION_90         = 1,
    ROTATION_180        = 2,
    ROTATION_270        = 3,
    ROTATION_BUTT
} ROTATION_E;
```

【注意事项】
无

## POINT_S

【说明】
定义坐标信息结构体。
【定义】

```
typedef struct rkPOINT_S {
    RK_S32 s32X;
    RK_S32 s32Y;
} POINT_S;
```

【注意事项】
无

## RECT_S

【说明】
定义矩形区域信息结构体。
【定义】

```
typedef struct rkRECT_S {
    RK_S32 s32X;
    RK_S32 s32Y;
    RK_U32 u32Width;
    RK_U32 u32Height;
} RECT_S;
```

【注意事项】
无

## CROP_INFO_S

【说明】
定义 CROP 属性结构体
【定义】

```
typedef struct rkCROP_INFO_S {
    RK_BOOL bEnable;
    RECT_S  stRect;
} CROP_INFO_S;
```

【注意事项】
无

## ROTATION_EX_S

【说明】
定义任意角度旋转属性。
【定义】

```
typedef struct rkROTATION_EX_S {
    /* RW;Range: [0, 2];Rotation mode*/
    ROTATION_VIEW_TYPE_E enViewType;
    /* RW;Range: [0,360];Rotation Angle:[0,360]*/
    RK_U32               u32Angle;
    /*
     * RW;Range: [-511, 511];Horizontal offset of the image
     * distortion center relative to image center.
     */
    RK_S32               s32CenterXOffset;
    /*
     * RW;Range: [-511, 511];Vertical offset of the image
     * distortion center relative to image center.
     */
    RK_S32               s32CenterYOffset;
    /* RW;Dest size of any angle rotation*/
    SIZE_S               stDestSize;
} ROTATION_EX_S;
```

【注意事项】
无

## MPP_CHN_S

【说明】
定义模块设备通道结构体。
【定义】

```
typedef struct rkMPP_CHN_S {
    MOD_ID_E    enModId;
    RK_S32      s32DevId;
    RK_S32      s32ChnId;
} MPP_CHN_S;
```

【成员】

| 成员名称 | 描述 |
| --- | --- |
| enModId | 模块号。 |
| s32DevId | 设备号。 |
| s32ChnId | 通道号。 |

【注意事项】
无

## CHN_INPUT_MODE_E

【说明】
定义通道输入流模式。
【定义】

```
typedef enum rkCHN_INPUT_MODE_E {
    CHN_INPUT_MODE_NORMAL,          /* CHN receive all packet */
    CHN_INPUT_MODE_REMAIN_NEWEST,   /* CHN remain newest packet */
    CHN_INPUT_MODE_DROP_ALWAYS,     /* CHN drop all packet */
    CHN_INPUT_MODE_BUTT
} CHN_INPUT_MODE_E;
```

【注意事项】
无

## MB_POOL

【说明】
定义MB内存池的句柄。
【定义】

```
typedef RK_U32 MB_POOL;
```

【注意事项】
无

## MB_BLK

【说明】
定义MB内存的句柄。
【定义】

```
typedef void * MB_BLK;
```

【注意事项】
无

## MB_MAX_POOLS

【说明】
定义MB内存池的最大个数。
【定义】

```
#define MB_MAX_POOLS            512
```

【注意事项】
无

## MB_POOL_CONFIG_S

【说明】
定义内存缓存池属性结构体。
【定义】

```
typedef struct rkMB_POOL_CONFIG_S {
    RK_U64 u64MBSize;
    RK_U32 u32MBCnt;
    MB_REMAP_MODE_E enRemapMode;
    MB_ALLOC_TYPE_E enAllocType;
    MB_DMA_TYPE_E enDmaType;
    RK_BOOL bPreAlloc;
} MB_POOL_CONFIG_S;
```

【成员】

| 成员名称 | 描述 |
| --- | --- |
| u64MBSize | 缓存块大小，以 Byte 位单位。 |
| u32MBCnt | 每个缓存池的缓存块个数。取值范围：(0, 10240] |
| enRemapMode | 内核态虚拟地址映射模式。 |
| enAllocType | 申请的内存类型。 |
| enDmaType | 申请的DMA物理内存类型，如配置是否物理连续 |
| bPreAlloc | 是否在缓存池创建时申请好缓存块。 |

【注意事项】
无

## MB_REMAP_MODE_E

【说明】
定义 MB 内核态虚拟地址映射模式。
【定义】

```
typedef enum rkMB_REMAP_MODE_E {
    MB_REMAP_MODE_NONE = 0, /* no remap */
    MB_REMAP_MODE_NOCACHE = 1 << 8, /* no cache remap */
    MB_REMAP_MODE_CACHED = 1 << 9, /* cache remap, if you use this mode, you
should flush cache by yourself */
    MB_REMAP_MODE_BUTT
} MB_REMAP_MODE_E;
```

【成员】

| 成员名称 | 描述 |
| --- | --- |
| MB_REMAP_MODE_NOCACHE | MB 映射 nocache 属性的内核态虚拟地址。 |
| MB_REMAP_MODE_CACHED | MB 映射 cached 属性的内核态虚拟地址。 |

【注意事项】
无

## MB_DMA_TYPE_E

【说明】
定义申请的物理内存类型。
【定义】

```
typedef enum rkMB_DMA_TYPE_E {
    MB_DMA_TYPE_NONE = 0, /* Physically Non-Continuous memory default */
    MB_DMA_TYPE_CMA = 1 << 12, /* Physically Continuous memory */
    MB_DMA_TYPE_BUTT
} MB_DMA_TYPE_E;
```

【成员】

| 成员名称 | 描述 |
| --- | --- |
| MB_DMA_TYPE_NONE | 申请的物理内存为非连续物理地址内存。 |
| MB_DMA_TYPE_CMA | 申请的物理内存为连续物理地址内存。 |

【注意事项】
无

## MB_ALLOC_TYPE_E

【说明】
定义MB申请的内存类型。
【定义】

```
typedef enum rkMB_ALLOC_TYPE {
    MB_ALLOC_TYPE_UNUSED = -1,
    MB_ALLOC_TYPE_DMA = 0,
    MB_ALLOC_TYPE_MALLOC,
    MB_ALLOC_TYPE_MAX,
} MB_ALLOC_TYPE_E;
```

【成员】

| 成员名称 | 描述 |
| --- | --- |
| MB_ALLOC_TYPE_DMA | 申请内核态可用的DMA内存。 |
| MB_ALLOC_TYPE_MALLOC | 申请malloc内存。 |

【注意事项】
无

## MB_EXT_CONFIG_S

【说明】
定义申请外部MB内存的信息。
【定义】

```
typedef struct _rkMB_EXT_CONFIG_S {
    RK_U8               *pu8VirAddr;
    RK_U64               u64Size;
    RK_MPI_MB_FREE_CB    pFreeCB;
    RK_VOID             *pOpaque;
} MB_EXT_CONFIG_S;
```

【成员】

| 成员名称 | 描述 |
| --- | --- |
| pu8VirAddr | 外部申请的虚拟内存。 |
| u64Size | 外部申请的内存大小。 |
| pFreeCB | 申请内存的释放回调方法。 |
| pOpaque | 申请内存的释放回调所需的上下文。 |

【注意事项】
无

## MB_CONFIG_S

【说明】
定义媒体内存缓存池属性结构体。
【定义】

```
typedef struct rkMB_CONFIG_S {
    RK_U32 u32MaxPoolCnt;
    MB_POOL_CONFIG_S astCommPool[MB_MAX_COMM_POOLS];
} MB_CONFIG_S;
```

【成员】

| 成员名称 | 描述 |
| --- | --- |
| u32MaxPoolCnt | 整个系统中可容纳的缓存池个数。 |
| astCommPool | 公共缓存池属性结构体。 |

【注意事项】
无

# 错误码

## 系统控制错误码

系统控制 API 错误码如下所示：

| 错误代码 | 宏定义 | 描述 |
| --- | --- | --- |
| 0xA0028006 | RK_ERR_SYS_NULL_PTR | 空指针错误 |
| 0xA0028010 | RK_ERR_SYS_NOTREADY | 系统控制属性未配置 |
| 0xA0028009 | RK_ERR_SYS_NOT_PERM | 操作不允许 |
| 0xA002800C | RK_ERR_SYS_NOMEM | 分配内存失败，如系统内存不足 |
| 0xA0028003 | RK_ERR_SYS_ILLEGAL_PARAM | 参数设置无效 |
| 0xA0028012 | RK_ERR_SYS_BUSY | 系统忙 |
| 0xA0028008 | RK_ERR_SYS_NOT_SUPPORT | 不支持的功能 |

## 内存缓存池错误码

| 错误代码 | 宏定义 | 描述 |
| --- | --- | --- |
| 0xA0028003 | RK_ERR_MB_ILLEGAL_PARAM | 参数设置无效 |
| 0xA0028005 | RK_ERR_MB_UNEXIST | 内存缓存池不存在 |
| 0xA0028006 | RK_ERR_MB_NULL_PTR | 参数空指针错误 |
| 0xA0028009 | RK_ERR_MB_NOT_PERM | 操作不允许 |
| 0xA002800C | RK_ERR_MB_NOMEM | 分配内存失败 |
| 0xA002800D | RK_ERR_MB_NOBUF | 分配缓存失败 |
| 0xA0028010 | RK_ERR_MB_NOTREADY | 系统控制属性未配置 |
| 0xA0028012 | RK_ERR_MB_BUSY | 系统忙 |
| 0xA0028013 | RK_ERR_MB_SIZE_NOT_ENOUGH | MB 块大小不够 |
| 0xA0028040 | RK_ERR_MB_2MPOOLS | 创建缓存池太多 |