 旷明智能 - QuaMing Intelligence -	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0




旷明 QuaMM 多媒体接口使用说明

部门	
文档编号	
版本号	V1.1.0
作者	旷明多媒体团队

版权所有

旷明智能科技（无锡）有限公司


本资料及其包含的所有内容为旷明智能科技（无锡）有限公司所有,受中国法律及适用之国际公约中有关著作权法律的保护。未经旷明智能科技（无锡）有限公司书面授权,任何人不得以任何形式复制、传播、散布、改动或以其它方式使用本资料的部分或全部内容,违者将被依法追究责任。

 旷明智能 - Quaming Intelligence -	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

更新记录 (Update History)

日期	更新人	版本	备注
2024/7/31		V1.0.0	初版完成
2024/12/21		V1.1.0	新增对QM10XV芯片支持

Quaming Confidential

 旷明智能 - QuaMing Intelligence -	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024. 08. 01
		版 本 Rev.	1. 1. 0


目录

1	概述	5
2	总体描述	6
2.1	基本概念	6
2.2	基本定义	6
2.3	基本流程	7
2.4	目录介绍	7
3	接口描述	8
3.1	通用接口	8
3.1.1	数据结构	8
3.1.2	函数接口	11
3.2	系统接口	13
3.2.1	数据结构	13
3.2.2	函数接口	15
3.3	视频模块接口	25
3.3.1	数据结构	25
3.3.2	解码设备函数接口	49
3.3.3	解码通道函数接口	50
3.3.4	编码设备函数接口	53
3.3.5	编码通道函数接口	53
3.4	音频模块接口	64
3.4.1	数据结构	64
3.4.2	音频输入输出函数接口	69
3.4.3	音频解码函数接口	72
3.4.4	音频编码函数接口	74
3.5	显示模块接口	77
3.5.1	数据结构	77
3.5.2	VO 设备函数接口	88
3.5.3	VO 通道函数接口	91
3.5.4	FB 设备函数接口	96
3.6	Camera 模块接口	101

 旷明智能 - Quaming Intelligence -	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版 本 Rev.	1.1.0

3.6.1	数据结构	101
3.6.2	函数接口	103
3.7	Filter 模块接口	105
3.7.1	数据结构	105
3.7.2	VGS 设备函数接口	126
3.7.3	TDE 设备函数接口	130
3.7.4	VPPU 设备函数接口	135
3.7.5	VPU 通道函数接口	138
3.8	AV 扩展模块接口	141
3.8.1	数据结构	141
3.8.2	媒体封装函数接口	149
3.8.3	媒体解封装函数接口	151
3.8.4	媒体解码函数接口	153
3.8.5	媒体编码函数接口	154

Quaming Confidential


 旷明智能 - Quaming Intelligence -	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

1 概述

本文档提供旷明软硬件平台解决方案的多媒体接口说明，方便客户开发查阅。

旷明多媒体接口（以下简称“QuaMM 接口”）采用 C 语言编写，屏蔽了旷明平台底层所采用的芯片和操作系统的差异，对上层应用程序提供统一的接口调用，方便客户开展应用程序的开发和跨平台移植。

Quaming Confidential

	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

2 总体描述

2.1 基本概念

平台 (Platform)：指旷明自研并向客户交付的软件和硬件解决方案，由芯片型号和操作系统组合来唯一标识。

模块 (Module)：指具有特定功能的多媒体子系统，如音频、视频、显示、Camera 等。

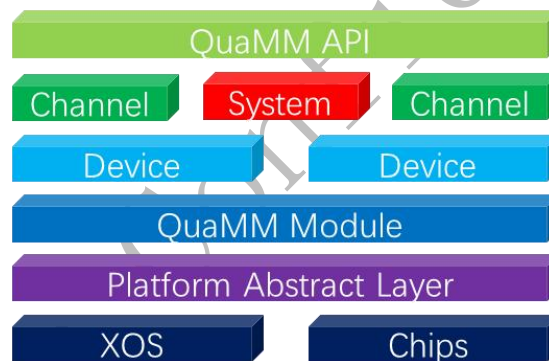
设备 (Device)：指多媒体模块下进一步区分功能的硬件或软件设备，比如视频模块的视频编码设备、视频解码设备，显示模块的 FB (帧缓冲) 设备、VO (视频输出) 设备等。

通道 (Channel)：指通过多媒体设备可以创建的输入和输出通道，一般会包含一路多媒体数据流，比如显示模块下的 VO 设备，可以通过它创建视频层通道来显示一路视频。不是所有的设备都支持创建通道。

系统 (System)：这里指旷明多媒体系统，属于系统模块，负责多媒体功能模块的运行管理和内存管理。

QuaMM 接口：应用程序可以直接调用从而能使用旷明多媒体框架提供的各种功能。

总体架构模型如下：



2.2 基本定义

平台、模块、设备、通道都通过 ID 来定义。

目前 QuaMM 接口里平台相关的定义包括：


- 支持的操作系统，包括 Android、鸿蒙等 Rich OS (简称“ROS”) 和 Linux、RT-Thread 等 Lite OS (简称“LOS”)，“ROS”和“LOS”统称“XOS”。

```
#define QUA_SYS_OS_ANDROID "android"
#define QUA_SYS_OS_HARMONY "harmony"
#define QUA_SYS_OS_LINUX "linux"
#define QUA_SYS_OS_RTT "rtt"
```

- 支持的芯片

```
#define QUA_SYS_CHIP_MC6870 "mc6870"
#define QUA_SYS_CHIP_MC331x "mc331x"
#define QUA_SYS_CHIP_QM10XH "qm10xh"
#define QUA_SYS_CHIP_QM10XV "qm10xv"
```

平台 ID 由芯片名与操作系统名组合而成，可通过 QUA_SYS_MAKE_PLATFORM 宏来生成具体的

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

平台 ID，比如：“mc6870_android”、“qm10xh_linux”、“mc331x_rtt”等。

目前 QuaMM 接口里定义的多媒体模块 ID 有：

```
#define QUA_MM_MODULE_VIDEO "quamm.video"
#define QUA_MM_MODULE_AUDIO "quamm.audio"
#define QUA_MM_MODULE_DISPLAY "quamm.display"
#define QUA_MM_MODULE_CAMERA "quamm.camera"
#define QUA_MM_MODULE_FILTER "quamm.filter"
#define QUA_MM_MODULE_AVEXT "quamm.avext"
```

设备 ID 的定义参见所属模块的头文件。

通道 ID 由整型数字来标识。

2.3 基本流程

QuaMM API 包含有通用接口、系统接口和各功能模块的接口，应用程序使用的流程一般如下：


1. 根据所使用的平台，首先调用通用接口里的 `qua_mm_init` 函数进行初始化，得到系统接口的函数操作指针 `qua_mm_system_ops_t`；
2. 根据步骤 1 里传入的参数是否是 `primary` 应用程序，如果是，必须调用系统接口的 `sys_init` 函数，其它系统接口函数根据需要在合适的时候调用，比如对硬件模块的绑定、视频缓存池创建、MMZ 内存操作等；
3. 根据应用程序所需要的功能来按序加载对应的模块，可以调用通用接口里的 `qua_mm_load_module` 函数，得到该模块的通用模块指针，类型为 `qua_mm_module_t`；
4. 根据应用程序所使用的功能模块的具体设备，通过通用模块指针调用 `open_device` 函数打开设备，得到该设备的通用设备指针，类型为 `qua_mm_device_t`，可以将通用设备指针转换为具体设备指针（比如显示模块的 VO 设备，指针类型为 `qua_mm_vo_device_t`），调用具体设备指针的操作函数对该设备进行属性设置或初始化、使能等（比如 VO 设备，可以调用其 `set_public_attr` 函数来设置该 VO 设备的基本属性并 `enable`）；
5. 根据应用程序的功能需求和以上具体设备的操作要求，通过通用设备指针调用 `create_channel` 函数创建该设备的一个通道，得到该通道的通用通道指针，类型为 `qua_mm_channel_t`，可以将通用通道指针转换为具体通道指针（比如 VO 设备的通道指针类型为 `qua_mm_vo_channel_t`），调用具体通道指针的操作函数对该通道进行属性设置或初始化、使能等；
6. 如果应用程序使用了多个模块，每个模块都按照上述步骤 3~5 进行操作，其中步骤 5 根据具体设备是否有通道定义做为可选步骤；
7. 应用程序在得到所需具体模块、具体设备、具体通道的操作接口函数后，可以在合适的时机进行调用来实现具体的功能。

2.4 目录介绍

对于旷明 LOS 工程，QuaMM 接口目录位于 `core/package/libnative/qlibquammapi` 下，其内部子目录说明如下：

- `include`: 存放 QuaMM 接口头文件；
- `sample`: QuaMM 接口的示例程序目录，可执行目录外面的 `build_sample.sh` 脚本一键编译；
- `platform`: 存放 QuaMM 编译出来的库文件。

对于旷明 ROS 工程，如 Android、鸿蒙系统，QuaMM 接口目录的位置另外指定。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

3 接口描述

3.1 通用接口

通用接口提供基于 QuaMM 开发的常用或必要接口。所在的头文件路径：`include/common`。

3.1.1 数据结构

qua_mm_module_t

【说明】

多媒体模块的通用表示。

【定义】

```
typedef struct qua_mm_module_t {
    QUA_U16 api_version;
    QUA_CONST_CHAR *id;
    QUA_VOID_PTR dso;
    QUA_S32 (*open_device)(const struct qua_mm_module_t* module, QUA_CONST_CHAR* dev_id,
    QUA_S32 dev_num, struct qua_mm_device_t** device);
} qua_mm_module_t;
```

【描述】

成员名称	描述
api_version	该模块的当前 API 版本
id	模块 ID
dso	API 内部使用，应用程序无需关心
open_device	打开该模块下的某个设备

所涉及的函数指针描述如下：

open_device:			
参数名称	输入/输出	描述	返回值
module	输入	设备所属的模块	成功: QUA_SUCCESS; 失败: 其它值
dev_id	输入	设备 ID	
dev_num	输入	设备号，见各模块下的设备定义	
device	输出	返回的设备通用实例	


qua_mm_device_t

【说明】

多媒体设备的通用表示。

【定义】

```
typedef struct qua_mm_device_t {
    QUA_CONST_CHAR *id;
    struct qua_mm_module_t* module;
```


 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024. 08. 01
	版本 Rev.	1. 1. 0

```

QUA_S32 (*close)(struct qua_mm_device_t* device);
QUA_S32 (*create_channel)(const struct qua_mm_device_t* device, QUA_S32* chn_id,
    QUA_VOID_PTR chn_attr, struct qua_mm_channel_t** channel);
} qua_mm_device_t;

```

【描述】

成员名称	描述
id	设备 id
module	设备所属的模块
close	关闭该设备
create_channel	创建该设备的一个通道

所涉及的函数指针描述如下：

close:			
参数名称	输入/输出	描述	返回值
device	输入	该设备的通用实例	成功: QUA_SUCCESS; 失败: 其它值

create_channel:			
参数名称	输入/输出	描述	返回值
device	输入	该设备的通用实例	成功: QUA_SUCCESS; 失败: 其它值
chn_id	输入或输出	通道 ID, 见具体设备的通道定义	
chn_attr	输入	通道属性参数, 见具体设备的通道定义	
channel	输出	返回的通道实例	

qua_mm_channel_t

【说明】

多媒体通道的通用表示。

【定义】

```

typedef struct qua_mm_channel_t {
    QUA_S32 id;
    struct qua_mm_device_t* device;
    QUA_S32 (*release)(struct qua_mm_channel_t* channel);
} qua_mm_channel_t;


```

【描述】

成员名称	描述
id	通道 id, 见具体设备的通道定义
device	通道所在的设备
release	释放该通道

所涉及的函数指针描述如下：

release:			
参数名称	输入/输出	描述	返回值
channel	输入	该通道的实例	成功: QUA_SUCCESS; 失败: 其它值

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_mm_system_t

【说明】

QuaMM 系统模块的通用表示。

【定义】

```
typedef struct qua_mm_system_t {
    struct qua_mm_module_t mod_info;
    QUA_CONST_CHAR *sdk_version;
} qua_mm_system_t;
```

【描述】

成员名称	描述
mod_info	系统模块信息，应用程序无需关心
sdk_version	QuaMM SDK 版本信息

qua_video_frame_info_t

【说明】

视频帧信息属性定义。

【定义】

```
typedef struct qua_video_frame_info_t {
    qua_video_frame_t video_frame;
    QUA_U32 pool_id;
} qua_video_frame_info_t;
```

【描述】

成员名称	描述
video_frame	视频帧属性
pool_id	当前视频帧 buffer 所属 buffer 池 id

qua_video_frame_t

【说明】

视频帧属性定义。

【定义】

```
typedef struct qua_video_frame_t {
    QUA_U32 width;
    QUA_U32 height;
    qua_video_field_t field;
    qua_pixel_fmt_t pixel_fmt;

    qua_video_format_t video_fmt;
    qua_compress_mode_t compress_mode;
```

	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

```

QUA_U64 phy_addr[3];
QUA_U64 vir_addr[3];
QUA_U32 stride[3];
QUA_U64 header_phy_addr[3];
QUA_U64 header_vir_addr[3];
QUA_U32 header_stride[3];

```

```

QUA_S16 offset_top;
QUA_S16 offset_bottom;
QUA_S16 offset_left;
QUA_S16 offset_right;

```

```

QUA_U64 pts;
QUA_U32 time_ref;
QUA_U32 priv_data;
qua_video_supplement_t supplement;
QUA_U32 src_expand_info;


```

```
} qua_video_frame_t;
```

【描述】

成员名称	描述
width	视频帧宽度
height	视频帧长度
field	视频帧场模式
pixel_fmt	视频帧像素格式
video_fmt	视频帧格式
compress_mode	压缩模式
phy_addr	物理地址
vir_addr	虚拟地址
stride	水平步长
header_phy_addr	头信息物理地址
header_vir_addr	头信息虚拟地址
header_stride	头信息步长
offset_top	上边界偏移
offset_bottom	下边界偏移
offset_left	左边界偏移
offset_right	右边界偏移
pts	时间戳
time_ref	参考时间
priv_data	私有数据
supplement	视频数据增强
src_expand_info	扩展信息

3.1.2 函数接口

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_mm_init

【说明】

QuaMM 初始化接口，应用程序必须调用，一个应用进程只需要调用一次。

【定义】

QUA_S32 qua_mm_init(QUA_BOOL primary_user, QUA_CONST_CHAR *platform, qua_mm_system_t **system)

【描述】

参数名称	输入/输出	描述	返回值
primary_user	输入	是否为首要调用者，一般开机启动后第一个调用该函数的应用进程为首要调用者	成功：QUA_SUCCESS； 失败：其它值
platform	输入	平台 ID	
system	输出	系统模块实例	

qua_mm_load_module

【说明】

加载一个多媒体模块实例。

【定义】

QUA_S32 qua_mm_load_module(QUA_CONST_CHAR *id, const qua_mm_module_t **module)

【描述】

参数名称	输入/输出	描述	返回值
id	输入	模块 ID	成功：QUA_SUCCESS； 失败：其它值
module	输出	返回的模块实例	

qua_mm_get_platform

【说明】


获取平台的 ID。

【定义】

QUA_CONST_CHAR *qua_mm_get_platform()

【描述】

参数名称	输入/输出	描述	返回值
无	无	无	成功：平台 ID 字符串； 失败：NULL

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

3.2 系统接口

系统模块主要提供系统初始化管理、模块之间的数据绑定管理、系统内存操作、视频缓存池管理等系统性接口。所在的头文件路径：`include/system`。

3.2.1 数据结构

qua_vb_config_t

【说明】

定义视频缓存池属性。

【定义】

```
typedef struct qua_vb_config {
    QUA_U32 max_pool_cnt;
    struct {
        QUA_U32 block_size;
        QUA_U32 block_cnt;
        QUA_CHAR pool_name[QUA_VB_MAX_POOL_NAME_LEN];
    } common_pools[QUA_VB_MAX_COMMON_POOLS];
} qua_vb_config_t;
```

【描述】

成员名称	描述
max_pool_cnt	系统中可容纳的缓存池个数，目前应用可赋固定值 <code>QUA_VB_MAX_POOLS</code>
common_pools	定义一组公共视频缓存池属性，其成员： block_size: 缓存池里的缓存快大小（Byte）； block_cnt: 缓存池里的缓存快数量； pool_name: 缓存池的名称

qua_mm_system_ops_t


【说明】

定义系统模块所涉及的函数接口。

【定义】

```
typedef struct qua_mm_system_ops_t {
    struct qua_mm_system_t system;

    QUA_S32 (*sys_init)(qua_vb_config_t *vb_config);
    QUA_S32 (*sys_bind)(qua_mpp_chn_t *src_chn, qua_mpp_chn_t *dst_chn);
    QUA_S32 (*sys_unbind)(qua_mpp_chn_t *src_chn, qua_mpp_chn_t *dst_chn);
    QUA_S32 (*sys_unbind_by_src)(qua_mpp_chn_t *src_chn);
    QUA_S32 (*sys_unbind_by_dst)(qua_mpp_chn_t *dst_chn);
};
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024. 08. 01
	版本 Rev.	1. 1. 0

```

QUA_S32 (*sys_get_bind_by_dst)(qua_mpp_chn_t *dst_chn, qua_mpp_chn_t *src_chn);
QUA_U32 (*sys_get_chipid)();
QUA_S32 (*sys_set_reg)(QUA_ULONG phy_addr, QUA_U32 value);
QUA_S32 (*sys_get_reg)(QUA_ULONG phy_addr, QUA_U32 *value);
QUA_VOID_PTR (*sys_mmap)(QUA_ULONG phy_addr, QUA_U32 size);
QUA_VOID_PTR (*sys_mmap_cache)(QUA_ULONG phy_addr, QUA_U32 size);
QUA_S32 (*sys_munmap)(QUA_VOID_PTR virt_addr, QUA_U32 size);
QUA_S32 (*sys_mflush_cache)(QUA_ULONG phy_addr, QUA_VOID_PTR virt_addr, QUA_U32
size);
QUA_S32 (*sys_mmz_alloc)(const QUA_CHAR *mmb_name, const QUA_CHAR *mmz_zone,
QUA_U32 size, QUA_ULONG *phy_addr, QUA_VOID_PTR *virt_addr);
QUA_S32 (*sys_mmz_alloc_cached)(const QUA_CHAR *mmb_name, const QUA_CHAR
*mmz_zone, QUA_U32 size, QUA_ULONG *phy_addr, QUA_VOID_PTR *virt_addr);
QUA_S32 (*sys_mmz_flush_cache)(QUA_ULONG phy_addr, QUA_VOID_PTR virt_addr,
QUA_U32 size);
QUA_S32 (*sys_mmz_free)(QUA_ULONG phy_addr, QUA_VOID_PTR virt_addr);
QUA_VB_POOL (*vb_create_pool)(QUA_U32 blk_size, QUA_U32 blk_cnt, const QUA_CHAR
*mmz_zone);
QUA_S32 (*vb_destroy_pool)(QUA_VB_POOL pool);
QUA_VB_BLK (*vb_get_block)(QUA_VB_POOL pool, QUA_U32 blk_size, const QUA_CHAR
*mmz_zone);
QUA_S32 (*vb_release_block)(QUA_VB_BLK block);
QUA_U64 (*vb_handle_to_phyaddr)(QUA_VB_BLK block);
QUA_VB_POOL (*vb_handle_to_poolid)(QUA_VB_BLK block);
QUA_S32 (*vb_inquire_user_cnt)(QUA_VB_BLK block);
QUA_S32 (*vb_mmap_pool)(QUA_VB_POOL pool);
QUA_S32 (*vb_munmap_pool)(QUA_VB_POOL pool);
QUA_S32 (*vb_get_block_virtaddr)(QUA_VB_POOL pool, QUA_U64 phy_addr,
QUA_VOID_PTR *virt_addr);
QUA_VB_POOL (*vb_create_virt_pool)(QUA_U32 blk_cnt, const QUA_CHAR *mmz_zone);
QUA_VB_BLK (*vb_add_block)(QUA_VB_POOL pool, QUA_U64 phy_addr, QUA_U32 size);
QUA_S32 (*clk_enable)(QUA_CHAR *clk_name);
QUA_S32 (*clk_disable)(QUA_CHAR *clk_name);
QUA_S32 (*clk_set_rate)(QUA_CHAR *clk_name, QUA_ULONG freq);
QUA_S32 (*clk_get_rate)(QUA_CHAR *clk_name, QUA_ULONG *freq);
} qua_mm_system_ops_t;

```

【描述】

成员名称	描述
system	系统模块的通用表示
sys_init	系统初始化
sys_bind	模块之间的通道绑定，绑定后能传输数据
sys_unbind	模块之间的通道去绑定
sys_unbind_by_src	解除数据源通道的所有绑定关系（存在多个接收者时也会全部解除）
sys_unbind_by_dst	解除数据接收通道的所有绑定关系



sys_get_bind_by_dst	根据数据接收通道获取数据源通道的信息
sys_get_chipid	获取芯片标识
sys_set_reg	设置寄存器/内存地址
sys_get_reg	获取寄存器/内存地址
sys_mmap	对一块物理内存映射出用户态虚拟地址
sys_mmap_cache	对一块物理内存映射出用户态虚拟地址，带 cache 属性
sys_munmap	解映射一块用户态虚拟内存
sys_mflush_cache	刷新 cache 里的内容到内存并且使 cache 里的内容无效
sys_mmz_alloc	申请 MMZ 内存，内存属性为 non-cacheable
sys_mmz_alloc_cached	申请 MMZ 内存，内存属性为 cacheable
sys_mmz_flush_cache	刷新 cache 里的内容到 MMZ 内存并且使 cache 里的内容无效
sys_mmz_free	释放 MMZ 内存
vb_create_pool	创建一个视频缓存池
vb_destroy_pool	销毁一个视频缓存池
vb_get_block	从视频缓存池获取一个缓存块
vb_release_block	释放一个缓存块到缓存池
vb_handle_to_phyaddr	获取缓存块的物理地址
vb_handle_to_poolid	获取缓存块所在缓存池的 ID
vb_inquire_user_cnt	查询缓存块的使用计数信息
vb_mmap_pool	为一个视频缓存池映射用户态虚拟地址
vb_munmap_pool	为一个视频缓存池解除用户态映射
vb_get_block_virtaddr	获取缓存池中缓存块的用户态虚拟地址
vb_create_virt_pool	创建虚拟缓存池，池子里没有物理缓存块
vb_add_block	往虚拟缓存池里增加缓存块
clk_enable	使能时钟
clk_disable	禁用时钟
clk_set_rate	设置时钟频率
clk_get_rate	获取时钟频率

以上表格里的函数指针成员放在下一节“函数接口”里做具体说明。

3. 2. 2 函数接口

sys_init


【说明】

系统初始化。

QUA_S32 (*sys_init)(qua_vb_config_t *vb_config)			
参数名称	输入/输出	描述	返回值
vb_config	输入	视频缓存池配置属性	成功: QUA_SUCCESS; 失败: 其它值

【注意】

- 公共视频缓存池的属性需根据应用所使用到的硬件模块要求和应用参数（如分辨率、视频格式）来设置。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

- 公共视频缓存池主要用来存放 Camera 模块的捕获图像，因此，对公共缓存池的不当操作（如占用过多的缓存块）会影响 QuaMM 的正常运行。

sys_bind

【说明】

模块之间的通道绑定，绑定后能传输数据。

QUA_S32 (*sys_bind)(qua_mpp_chn_t *src_chn, qua_mpp_chn_t *dst_chn)			
参数名称	输入/输出	描述	返回值
src_chn	输入	数据源通道	成功: QUA_SUCCESS; 失败: 其它值
dst_chn	输入	数据接收通道	

sys_unbind

【说明】

模块之间的通道去绑定。

QUA_S32 (*sys_unbind)(qua_mpp_chn_t *src_chn, qua_mpp_chn_t *dst_chn)			
参数名称	输入/输出	描述	返回值
src_chn	输入	数据源通道	成功: QUA_SUCCESS; 失败: 其它值
dst_chn	输入	数据接收通道	

sys_unbind_by_src

【说明】

解除数据源通道的所有绑定关系（存在多个接收者时也会全部解除）。

QUA_S32 (*sys_unbind_by_src)(qua_mpp_chn_t *src_chn)			
参数名称	输入/输出	描述	返回值
src_chn	输入	数据源通道	成功: QUA_SUCCESS; 失败: 其它值

【注意】

"mc6870"和"mc331x"芯片平台不支持该接口。


sys_unbind_by_dst

【说明】

解除数据接收通道的所有绑定关系。

QUA_S32 sys_unbind_by_dst(qua_mpp_chn_t *dst_chn)			
参数名称	输入/输出	描述	返回值
dst_chn	输入	数据接收通道	成功: QUA_SUCCESS; 失败: 其它值

【注意】

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

"mc6870"和"mc331x"芯片平台不支持该接口。

sys_get_bind_by_dst

【说明】

根据数据接收通道获取数据源通道的信息。

QUA_S32 sys_get_bind_by_dst(qua_mpp_chn_t *dst_chn, qua_mpp_chn_t *src_chn)			
参数名称	输入/输出	描述	返回值
dst_chn	输入	数据接收通道	成功: QUA_SUCCESS;
src_chn	输出	数据源通道	失败: 其它值

【注意】

"mc6870"和"mc331x"芯片平台不支持该接口。

sys_set_reg

【说明】

设置寄存器/内存地址。

QUA_S32 (*sys_set_reg)(QUA_ULONG phy_addr, QUA_U32 value)			
参数名称	输入/输出	描述	返回值
phy_addr	输入	寄存器的物理地址	成功: QUA_SUCCESS;
value	输入	要设置的寄存器值	失败: 其它值

【注意】

"mc6870"和"mc331x"芯片平台不支持该接口。

sys_get_reg

【说明】

获取寄存器/内存地址。

QUA_S32 (*sys_get_reg)(QUA_ULONG phy_addr, QUA_U32 *value)			
参数名称	输入/输出	描述	返回值
phy_addr	输入	寄存器的物理地址	成功: QUA_SUCCESS;
value	输出	获取的寄存器值	失败: 其它值

【注意】

"mc6870"和"mc331x"芯片平台不支持该接口。

sys_mmap

【说明】

对一块物理内存映射出用户态虚拟地址。

QUA_VOID_PTR (*sys_mmap)(QUA_ULONG phy_addr, QUA_U32 size)			
参数名称	输入/输出	描述	返回值



phy_addr	输入	内存块的起始物理地址	成功：用户态虚拟地址； 失败：NULL
size	输入	要映射的内存长度（Byte）	

sys_mmap_cache

【说明】

对一块物理内存映射出用户态虚拟地址，带 cache 属性。

QUA_VOID_PTR (*sys_mmap_cache)(QUA_ULONG phy_addr, QUA_U32 size)			
参数名称	输入/输出	描述	返回值
phy_addr	输入	内存块的起始物理地址	成功：用户态虚拟地址； 失败：NULL
size	输入	要映射的内存长度（Byte）	

sys_munmap

【说明】

解映射一块用户态虚拟内存。

QUA_S32 (*sys_munmap)(QUA_VOID_PTR virt_addr, QUA_U32 size)			
参数名称	输入/输出	描述	返回值
virt_addr	输入	用户态虚拟地址	成功：QUA_SUCCESS； 失败：其它值
size	输入	内存块长度（Byte）	

sys_mflush_cache

【说明】

刷新 cache 里的内容到内存并且使 cache 里的内容无效。

QUA_S32 (*sys_mflush_cache)(QUA_ULONG phy_addr, QUA_VOID_PTR virt_addr, QUA_U32 size)			
参数名称	输入/输出	描述	返回值
phy_addr	输入	内存块的起始物理地址	成功：QUA_SUCCESS； 失败：其它值
virt_addr	输入	用户态虚拟地址	
size	输入	内存块长度（Byte）	

sys_mmz_alloc

【说明】

申请 MMZ 内存，内存属性为 non-cacheable。

QUA_S32 (*sys_mmz_alloc)(const QUA_CHAR *mmb_name, const QUA_CHAR *mmz_zone, QUA_U32 size, QUA_ULONG *phy_addr, QUA_VOID_PTR *virt_addr)			
参数名称	输入/输出	描述	返回值
mmb_name	输入	MMZ 内存块名称	成功：QUA_SUCCESS； 失败：其它值
mmz_zone	输入	MMZ 内存区域名称	



size	输入	内存块长度 (Byte)	
phy_addr	输出	内存块的起始物理地址	
virt_addr	输出	用户态虚拟地址	

【注意】

如果 MMZ 中有名为 anonymous 的区域，mmz_zone 可设为 NULL。如果 mmb_name 设为 NULL，创建的内存分块名为 “<null>”。

sys_mmz_alloc_cached

【说明】

申请 MMZ 内存，内存属性为 cacheable。

QUA_S32 (*sys_mmz_alloc_cached)(const QUA_CHAR *mmb_name, const QUA_CHAR *mmz_zone, QUA_U32 size, QUA_ULONG *phy_addr, QUA_VOID_PTR *virt_addr)

参数名称	输入/输出	描述	返回值
mmb_name	输入	MMZ 内存块名称	成功: QUA_SUCCESS; 失败: 其它值
mmz_zone	输入	MMZ 内存区域名称	
size	输入	内存块长度 (Byte)	
phy_addr	输出	内存块的起始物理地址	
virt_addr	输出	用户态虚拟地址	

sys_mmz_flush_cache

【说明】

刷新 cache 里的内容到 MMZ 内存并且使 cache 里的内容无效。

QUA_S32 (*sys_mmz_flush_cache)(QUA_ULONG phy_addr, QUA_VOID_PTR virt_addr, QUA_U32 size)

参数名称	输入/输出	描述	返回值
phy_addr	输入	内存块的起始物理地址	成功: QUA_SUCCESS; 失败: 其它值
virt_addr	输入	用户态虚拟地址	
size	输入	内存块长度 (Byte)	

【注意】

- 当 cache 里的数据为最新数据时，为了保证不能直接访问 cache 的硬件在访问内存时能够得到正确的数据，此时需要先调用此接口将 cache 里的内容更新到内存，这样，当硬件访问内存时，保证了数据的一致性和正确性。
- 此接口应与 sys_mmz_alloc_cached 接口配套使用。
- 若将 size 设为 0，则表示操作整个 cache 区域。

sys_mmz_free

【说明】

释放 MMZ 内存。

QUA_S32 (*sys_mmz_free)(QUA_ULONG phy_addr, QUA_VOID_PTR virt_addr)



参数名称	输入/输出	描述	返回值
phy_addr	输入	内存块的起始物理地址	成功: QUA_SUCCESS;
virt_addr	输入	用户态虚拟地址	失败: 其它值

vb_create_pool

【说明】

创建一个视频缓存池。

QUA_VB_POOL (*vb_create_pool)(QUA_U32 blk_size, QUA_U32 blk_cnt, const QUA_CHAR *mmz_zone)

参数名称	输入/输出	描述	返回值
blk_size	输入	视频缓存池的缓存块长度 (Byte)	成功: 视频缓存池 ID;
blk_cnt	输入	视频缓存池的缓存块数量	失 败 :
mmz_zone	输入	视频缓存池所在的 MMZ 内存区域名称	QUA_VB_INVALID_POOLID 或其它值

【注意】

- 该缓存池是从空闲 MMZ 内存中分配，一个缓存池包含若干个大小相同的缓存块，如果该缓存池的大小超过了预留内存中的空闲空间，则创建缓存池会失败。
- 如果 mmz_zone 设为 NULL，则表示从 anonymous 的区域创建缓存池。

vb_destroy_pool

【说明】

销毁一个视频缓存池。

QUA_S32 (*vb_destroy_pool)(QUA_VB_POOL pool)

参数名称	输入/输出	描述	返回值
pool	输入	视频缓存池 ID	成功: QUA_SUCCESS;
			失败: 其它值

【注意】

- 销毁缓存池前，要确保该缓存池的所有已经被获取的缓存块已经被释放掉，否则会销毁失败。
- 传入的缓存池 ID 应当在[0, QUA_VB_MAX_POOLS]范围内，且是通过 vb_create_pool 创建的未销毁的缓存池 ID。

vb_get_block

【说明】

从视频缓存池获取一个缓存块。

QUA_VB_BLK (*vb_get_block)(QUA_VB_POOL pool, QUA_U32 blk_size, const QUA_CHAR *mmz_zone)

参数名称	输入/输出	描述	返回值
pool	输入	视频缓存池 ID	成功: 缓存块 ID;
blk_size	输入	视频缓存池中每个缓存块的大小	失 败 :



		(Byte)	QUA_VB_INVALID_HANDLE 或其它值
mmz_zone	输入	视频缓存池所在的 MMZ 区域名称	

【注意】

- 用户可以在创建一个缓存池之后，调用本接口从该缓存池中获取一个缓存块（此时参数 blk_size 可以指定为 0）。
- 如果用户需要从任意一个公共缓存池中获取一块指定大小的缓存块，则可以将参数 pool 设置为无效 ID 号 (QUA_VB_INVALID_POOLID)，将参数 blk_size 设置为需要的缓存块大小，如果将 mmz_zone 设置为 NULL，则表示在 anonymous 的 MMZ 区域上的公共缓存池获取缓存块。

vb_release_block**【说明】**

释放一个缓存块到缓存池。

QUA_S32 (*vb_release_block)(QUA_VB_BLK block)			
参数名称	输入/输出	描述	返回值
block	输入	缓存块 ID	成功: QUA_SUCCESS; 失败: 其它值

【注意】

获取的缓存块使用完后，应该调用此接口释放缓存块。

vb_handle_to_phyaddr**【说明】**

获取缓存块的物理地址。


QUA_U64 (*vb_handle_to_phyaddr)(QUA_VB_BLK block)			
参数名称	输入/输出	描述	返回值
block	输入	缓存块 ID	成功: 缓存块的物理地址; 失败: 其它负数

vb_handle_to_poolid**【说明】**

获取缓存块所在缓存池的 ID。

QUA_VB_POOL (*vb_handle_to_poolid)(QUA_VB_BLK block)			
参数名称	输入/输出	描述	返回值
block	输入	缓存块 ID	成功: 缓存块所在的缓存池 ID; 失败: 其它负数

vb_inquire_user_cnt**【说明】**

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

查询缓存块的使用计数信息。

QUA_S32 (*vb_inquire_user_cnt)(QUA_VB_BLK block)			
参数名称	输入/输出	描述	返回值
block	输入	缓存块 ID	成功：缓存块使用计数值； 失败：其它负数

vb_mmap_pool

【说明】

为一个视频缓存池映射用户态虚拟地址。

QUA_S32 (*vb_mmap_pool)(QUA_VB_POOL pool)			
参数名称	输入/输出	描述	返回值
pool	输入	视频缓存池 ID	成功：QUA_SUCCESS； 失败：其它值

【注意】

重复映射视为成功。

vb_munmap_pool

【说明】

为一个视频缓存池解除用户态映射。

QUA_S32 (*vb_munmap_pool)(QUA_VB_POOL pool)			
参数名称	输入/输出	描述	返回值
pool	输入	视频缓存池 ID	成功：QUA_SUCCESS； 失败：其它值

【注意】

确保视频缓存池中的缓存块没有被应用程序占用，如果被占用，视为正在使用映射的用户态虚拟地址，本调用会返回失败。

vb_get_block_virtaddr


【说明】

获取缓存池中缓存块的用户态虚拟地址。

QUA_S32 (*vb_get_block_virtaddr)(QUA_VB_POOL pool, QUA_U64 phy_addr, QUA_VOID_PTR *virt_addr)			
参数名称	输入/输出	描述	返回值
pool	输入	视频缓存池 ID	成功：QUA_SUCCESS； 失败：其它值
phy_addr	输入	缓存池中的缓存快物理地址	
virt_addr	输出	缓存块的用户态虚拟地址	

【注意】

必须已调用了 vb_mmap_pool 对该缓存池做了用户态虚拟地址映射。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

clk_enable

【说明】

使能时钟。

QUA_S32 (*clk_enable)(QUA_CHAR *clk_name)			
参数名称	输入/输出	描述	返回值
clk_name	输入	时钟名称	成功: QUA_SUCCESS; 失败: 其它值

【注意】

"mc6870"和"mc331x"芯片平台不支持该接口。

clk_disable

【说明】

禁用时钟。

QUA_S32 (*clk_disable)(QUA_CHAR *clk_name)			
参数名称	输入/输出	描述	返回值
clk_name	输入	时钟名称	成功: QUA_SUCCESS; 失败: 其它值

【注意】

"mc6870"和"mc331x"芯片平台不支持该接口。

clk_set_rate

【说明】

设置时钟频率

QUA_S32 (*clk_set_rate)(QUA_CHAR *clk_name, QUA_ULONG freq)			
参数名称	输入/输出	描述	返回值
clk_name	输入	时钟名称	成功: QUA_SUCCESS; 失败: 其它值
freq	输入	要设置的时钟频率	

【注意】

"mc6870"和"mc331x"芯片平台不支持该接口。

clk_get_rate

【说明】

获取时钟频率。

QUA_S32 (*clk_get_rate)(QUA_CHAR *clk_name, QUA_ULONG *freq)			
参数名称	输入/输出	描述	返回值
clk_name	输入	时钟名称	成功: QUA_SUCCESS;



QuaMM 多媒体接口使用说明

生效日期
Effective Date

2024.08.01

版本
Rev.


1.1.0

freq	输出	返回的时钟频率	失败：其它值
------	----	---------	--------

【注意】

"mc6870"和"mc331x"芯片平台不支持该接口。

Quaming Confidential

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

3.3 视频模块接口

视频模块主要提供包括 H.264/H.265/MJPEG/JPEG 等编解码接口，所在的头文件路径：`include/video`。

3.3.1 数据结构

qua_mm_video_module_t

【说明】

视频模块定义

【定义】

```
typedef struct qua_mm_video_module_t {
    struct qua_mm_module_t parent;
} qua_mm_video_module_t;
```

【描述】

成员名称	描述
parent	详情见通用数据结构定义

qua_mm_vdec_device_t

【说明】

解码设备定义。

【定义】

```
typedef struct qua_mm_vdec_device_t {
    struct qua_mm_device_t parent;

    QUA_S32 (*init)();
    QUA_S32 (*exit)();
} qua_mm_vdec_device_t;
```

【描述】

成员名称	描述
parent	详情见通用数据结构定义
init	初始化解码设备
exit	去初始化解码设备

以上表格里的函数指针成员放在 3.3.2 节“解码设备函数接口”里做具体说明。

qua_mm_vdec_channel_t

【说明】

解码通道定义。

【定义】

```
typedef struct qua_mm_vdec_channel_t {
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024. 08. 01
	版本 Rev.	1. 1. 0

```
struct qua_mm_channel_t parent;
```

```

QUA_S32 (*start_chn)(QUA_S32 chn_id);
QUA_S32 (*stop_chn)(QUA_S32 chn_id);
QUA_S32 (*reset_chn)(QUA_S32 chn_id);
QUA_S32 (*get_chn_fd)(QUA_S32 chn_id);
QUA_S32 (*send_stream)(QUA_S32 chn_id, const qua_vdec_stream_t*packet,QUA_S32 timeout_millis);
QUA_S32 (*get_frame)(QUA_S32chn_id,qua_video_frame_info_t*frame_info,QUA_S32 timeout_millis);
QUA_S32 (*release_frame)(QUA_S32 chn_id, qua_video_frame_info_t *frame_info);
QUA_S32 (*get_chn_attr)(QUA_S32 chn_id, qua_vdec_chn_attr_t *chn_attr);
QUA_S32 (*query_chn)(QUA_S32 chn_id, qua_vdec_chn_stat_t *chn_stat);
QUA_S32 (*get_chn_param)(QUA_S32 chn_id, qua_vdec_chn_param_t* chn_param);
QUA_S32 (*set_chn_param)(QUA_S32 chn_id, qua_vdec_chn_param_t* chn_param);

```

```
} qua_mm_vdec_channel_t;
```

【描述】

成员名称	描述
parent	详情见通用数据结构定义
start_chn	解码器开始接收用户发送的码流
stop_chn	解码器停止接收用户发送的码流
reset_chn	复位视频解码通道
get_chn_fd	获取视频解码通道对应的文件描述符
send_stream	向视频解码通道发送码流数据
get_frame	获取视频解码通道的解码图像
release_frame	释放视频解码通道的图像
get_chn_attr	获取视频解码通道属性
query_chn	查询解码通道状态
get_chn_param	获取解码通道参数
set_chn_param	设置解码通道参数

以上表格里的函数指针成员放在 3.3.3 节“解码通道函数接口”里做具体说明。

qua_mm_venc_device_t

【说明】

编码设备定义。

【定义】

```

typedef struct qua_mm_venc_device_t {
    struct qua_mm_device_t parent;

    QUA_S32 (*init)();
    QUA_S32 (*deinit)();
} qua_mm_venc_device_t;

```

【描述】

成员名称	描述
parent	详情见通用数据结构定义

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024. 08. 01
	版本 Rev.	1. 1. 0

init	初始化编码设备
deinit	去初始化编码设备

以上表格里的函数指针成员放在 3.3.4 “编码设备函数接口” 里做具体说明。

qua_mm_venc_channel_t

【说明】


编码通道定义。

【定义】

```
typedef struct qua_mm_venc_channel_t {
    struct qua_mm_channel_t parent;

    QUA_S32 (*reset_chn)(QUA_S32 chn_id);
    QUA_S32 (*start_recv_pic)(QUA_S32 chn_id);
    QUA_S32 (*stop_recv_pic)(QUA_S32 chn_id);
    QUA_S32 (*query_chn)(QUA_S32 chn_id, qua_venc_chn_stat_t *chn_stat);
    QUA_S32 (*set_chn_attr)(QUA_S32 chn_id, const qua_venc_chn_attr_t *chn_attr);
    QUA_S32 (*get_chn_attr)(QUA_S32 chn_id, qua_venc_chn_attr_t *chn_attr);
    QUA_S32 (*get_stream)(QUA_S32 chn_id, qua_venc_stream_t *packet, QUA_S32 timeout_millis);
    QUA_S32 (*release_stream)(QUA_S32 chn_id, qua_venc_stream_t *packet);
    QUA_S32(*send_frame)(QUA_S32chn_id,qua_video_frame_info_t*frame_info,QUA_S32
    timeout_millis);
    QUA_S32 (*set_rotate)(QUA_S32 chn_id, const qua_rotate_t rotate_ops);
    QUA_S32 (*get_rotate)(QUA_S32 chn_id, qua_rotate_t *rotate_ops);
    QUA_S32(*check_in_addr)(QUA_S32chn_id,QUA_U64in_addr,QUA_U32size,QUA_S32 timeout_millis,
    QUA_U32 *is_idle);
    QUA_S32 (*set_roi_map)(QUA_S32 chn_id, const qua_roi_map_t *roi_map);
    QUA_S32 (*set_roi_cfg)(QUA_S32 chn_id, const qua_roi_t *roi_cfg);
    QUA_S32 (*clear_roi)(QUA_S32 chn_id);
    QUA_S32 (*set_rc_attr)(QUA_S32 chn_id, const qua_venc_rc_attr_t *rc_attr);
    QUA_S32 (*get_rc_attr)(QUA_S32 chn_id, qua_venc_rc_attr_t *rc_attr);
    QUA_S32 (*set_rc_change_param)(QUA_S32 chn_id, const qua_venc_rc_change_param_t *rc_params);
    QUA_S32(*set_enc_param)(QUA_S32chn_id,qua_enc_param_cmd_t cmd, const qua_enc_param_union_t
    *param);
    QUA_S32(*get_enc_param)(QUA_S32chn_id,qua_enc_param_cmd_tcmd, const qua_enc_param_union_t
    *param);
    QUA_S32 (*set_h264_entropy)(QUA_S32 chn_id, const qua_h264_entropy_t *entropy);
    QUA_S32 (*get_h264_entropy)(QUA_S32 chn_id, qua_h264_entropy_t *entropy);
    QUA_S32 (*set_h264_dblk)(QUA_S32 chn_id, const qua_h264_dblk_t *dblk);
    QUA_S32 (*get_h264_dblk)(QUA_S32 chn_id, qua_h264_dblk_t *dblk);
    QUA_S32 (*set_h265_entropy)(QUA_S32 chn_id, const qua_h265_entropy_t entropy);
    QUA_S32 (*get_h265_entropy)(QUA_S32 chn_id, qua_h265_entropy_t *entropy);
    QUA_S32 (*set_h265_dblk)(QUA_S32 chn_id, const qua_h265_dblk_t *dblk);
    QUA_S32 (*get_h265_dblk)(QUA_S32 chn_id, qua_h265_dblk_t *dblk);

```

	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024. 08. 01
		版本 Rev.	1. 1. 0

```


QUA_S32 (*set_debreath_effect)(QUA_S32 chn_id, const qua_debreath_t debreath);
QUA_S32 (*get_debreath_effect)(QUA_S32 chn_id, qua_debreath_t debreath);
QUA_S32 (*clear_yuv_queue)(QUA_S32 chn_id);
QUA_S32 (*request_idr)(QUA_S32 chn_id);
} qua_mm_venc_channel_t;

```

【描述】

成员名称	描述
reset_chn	复位视频编码通道
start_recv_pic	通道开启接收图像数据
stop_recv_pic	通道停止接收图像数据
query_chn	查询通道状态信息
set_chn_attr	设置通道属性参数
get_chn_attr	获取通道属性参数
get_stream	获取指定编码通道的码流
release_stream	释放码流
send_frame	向编码通道提交图像数据进行编码
set_rotate	设置编码旋转属性
get_rotate	获取编码旋转属性
check_in_addr	检查 VENC 输入缓存地址是否被使用
set_roi_map	设置 ROI Map
set_roi_cfg	设置 ROI 属性
clear_roi	清除之前设置的 ROI 配置
set_rc_attr	设置码率控制参数
get_rc_attr	获取码率控制参数
set_rc_change_param	实时调整部分码控参数, 通过该接口设置时会保留之前的码率情况, 不会重置码控和强制 I 帧
set_enc_param	配置编码参数
get_enc_param	获取编码参数
set_h264_entropy	设置 H.264 熵编码类型
get_h264_entropy	获取 H.264 熵编码类型
set_h264_dblk	设置 H.264 Deblocking 参数
get_h264_dblk	获取 H.264 Deblocking 参数
set_h265_entropy	设置 H.265 熵编码类型
get_h265_entropy	获取 H.265 熵编码类型
set_h265_dblk	设置 H.265 Deblocking 参数
get_h265_dblk	获取 H.265 Deblocking 参数
set_debreath_effect	设置去呼吸效应
get_debreath_effect	获取去呼吸效应使能状态
clear_yuv_queue	清空编码 yuv 队列 & 输出码流队列
request_idr	申请强制 I 帧

以上表格里的函数指针成员放在 3.3.5 节“编码通道函数接口”里做具体说明。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_mm_jpege_channel_t

【说明】

编码 jpeg 通道定义。

【定义】

```
typedef struct qua_mm_jpege_channel_t {
    struct qua_mm_channel_t parent;

    QUA_S32 (*reset_chn)(QUA_S32 chn_id);
    QUA_S32 (*start_recv_pic)(QUA_S32 chn_id);
    QUA_S32 (*stop_recv_pic)(QUA_S32 chn_id);
    QUA_S32 (*query_chn)(QUA_S32 chn_id, qua_venc_chn_stat_t *chn_stat);
    QUA_S32 (*set_chn_attr)(QUA_S32 chn_id, const qua_venc_chn_attr_t *chn_attr);
    QUA_S32 (*get_chn_attr)(QUA_S32 chn_id, qua_venc_chn_attr_t *chn_attr);
    QUA_S32 (*get_stream)(QUA_S32 chn_id, qua_venc_stream_t *packet, QUA_S32 timeout_millis);
    QUA_S32 (*release_stream)(QUA_S32 chn_id, qua_venc_stream_t *packet);
    QUA_S32 (*send_frame)(QUA_S32 chn_id, qua_video_frame_info_t *frame_info, QUA_S32
timeout_millis);
    QUA_S32 (*set_max_stream_cnt)(QUA_S32 chn_id, QUA_U32 max_strm_cnt);
    QUA_S32 (*get_max_stream_cnt)(QUA_S32 chn_id, QUA_U32 *max_strm_cnt);
    QUA_S32 (*get_jpeg_param)(QUA_S32 chn_id, qua_venc_param_jpeg_t *jpeg_param);
    QUA_S32 (*set_jpeg_param)(QUA_S32 chn_id, qua_venc_param_jpeg_t *jpeg_param);
    QUA_S32 (*set_crop)(QUA_S32 chn_id, const qua_venc_crop_cfg_t *crop_cfg);
    QUA_S32 (*get_crop)(QUA_S32 chn_id, qua_venc_crop_cfg_t *crop_cfg);
    QUA_S32 (*set_rotate)(QUA_S32 chn_id, const qua_rotate_t rotate_ops);
    QUA_S32 (*get_rotate)(QUA_S32 chn_id, qua_rotate_t *rotate_ops);
    QUA_S32 (*get_chn_fd)(QUA_S32 chn_id);
} qua_mm_jpege_channel_t;
```

【描述】

成员名称	描述
reset_chn	复位视频编码通道
start_recv_pic	通道开启接收图像数据
stop_recv_pic	通道停止接收图像数据
query_chn	查询通道状态信息
set_chn_attr	设置通道属性参数
get_chn_attr	获取通道属性参数
get_stream	获取指定编码通道的码流
release_stream	释放码流
send_frame	向编码通道提交图像数据进行编码
set_max_stream_cnt	设置 JPEG 编码通道最多编码的缓存帧数
get_max_stream_cnt	获得 JPEG 编码通道最多编码的缓存帧数
get_jpeg_param	获取 JPEG 编码通道 QP 值等
set_jpeg_param	设置 JPEG 编码通道 QP 值等

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

set_crop	设置 JPEG 编码通道 crop 参数
get_crop	获取 JPEG 编码通道 crop 参数
set_rotate	设置 JPEG 编码通道 rotate 参数
get_rotate	获取 JPEG 编码通道 rotate 参数
get_chn_fd	获得 JPEG 编码通道的文件句柄

以上表格里的函数指针成员放在 3.3.5 节“编码通道函数接口”里做具体说明

qua_video_channel_t

【说明】

通道类型定义。

【定义】

```
typedef enum qua_video_channel_t {
    QUA_MM_VIDEO_VENC_CHN,
    QUA_MM_VIDEO_JPEGE_CHN
} qua_video_channel_t;
```

【描述】

成员名称	描述
QUA_MM_VIDEO_VENC_CHN	创建类型格式为 H.264/H265
QUA_MM_VIDEO_JPEGE_CHN	创建类型格式为 JPEG

qua_mm_vdec_chn_attr_t

【说明】

解码属性定义。

【定义】

```
typedef struct qua_mm_vdec_chn_attr_t {
    qua_vdec_chn_attr_t chn_attr;

    QUA_S32 vb_cnt;
} qua_mm_vdec_chn_attr_t;
```

【描述】

成员名称	描述
chn_attr	通道属性
vb_cnt	vb 个数（解码 buffer），至少需要参考帧+1


qua_vdec_chn_attr_t

【说明】

解码通道属性定义。

【定义】

```
typedef struct qua_vdec_chn_attr_t {
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```

qua_coding_type_t coding_type;
QUA_U32 in_stream_buf_size;
QUA_U32 priority;
QUA_U32 pic_width;
QUA_U32 pic_height;

```

```

union {
    qua_vdec_attr_jpeg_t vdec_jpeg_attr;
    qua_vdec_attr_video_t vdec_video_attr;
};

```

```
} qua_vdec_chn_attr_t;
```

【描述】

成员名称	描述
coding_type	解码协议类型
in_stream_buf_size	码流 Buffer 大小，一般配置 w*h*3/4
priority	通道优先级（不支持）
pic_width	通道宽
pic_height	通道高
vdec_jpeg_attr	jpeg 或 mjpeg 通道的相关属性
vdec_video_attr	h264 或 h265 通道的相关属性

qua_vdec_attr_jpeg_t

【说明】

解码 jpeg/mjpeg 通道属性定义。

【定义】

```

typedef struct qua_vdec_attr_jpeg_t {
    qua_video_mode_t mode;
    qua_vdec_jpeg_color_fmt_t jpeg_fmt;
} qua_vdec_attr_jpeg_t;

```

【描述】

成员名称	描述
mode	视频流模式，只支持 frame
jpeg_fmt	jpeg 格式

qua_vdec_attr_video_t

【说明】


解码 h264/h265 通道属性定义。

【定义】

```

typedef struct qua_vdec_attr_video_t {
    QUA_U32 ref_frame_num;
    qua_video_mode_t mode;
}

```

	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

```

QUA_BOOL temporal_mvp_enable;
} qua_vdec_attr_video_t;

```

【描述】

成员名称	描述
ref_frame_num	参考帧个数（无效）
mode	视频流模式，只支持 frame
temporal_mvp_enable	指定时间运动矢量是否用于帧间预测（不支持）

qua_vdec_stream_t

【说明】

解码输入码流属性定义。

【定义】

```

typedef struct qua_vdec_stream_t {
    QUA_U64 in_data;
    QUA_U32 in_size;
    QUA_U64 pts;
    QUA_BOOL end_of_frame;
    QUA_BOOL end_of_stream;
} qua_vdec_stream_t;

```

【描述】

成员名称	描述
in_data	输入码流 buffer
in_size	输入码流大小
pts	输入时间戳
end_of_frame	是否为一帧结束，只支持送完整的一帧数据
end_of_stream	是否为流结束

qua_vdec_chn_stat_t

【说明】


解码通道状态属性定义。

【定义】

```

typedef struct qua_vdec_chn_stat_t {
    qua_coding_type_t coding_type;
    QUA_U32 left_stream_bytes;
    QUA_U32 left_stream_frames;
    QUA_U32 left_pics;
    QUA_BOOL start_rcv_stream;
    QUA_U32 rcv_stream_frames;
    QUA_U32 decode_stream_frames;
    qua_decoder_error_t dec_err;
} qua_vdec_chn_stat_t;

```


 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

【描述】

成员名称	描述
coding_type	解码协议
left_stream_bytes	待解码码流字节数
left_stream_frames	待解码帧数
left_pics	待获取的图像数
start_recv_stream	是否允许接收码流
recv_stream_frames	已接收的码流帧数
decode_stream_frames	已解码的码流帧数
dec_err	解码错误信息

qua_vdec_chn_param_t

【说明】

解码通道参数属性定义。

【定义】

```
typedef struct qua_vdec_chn_param_t {
    QUA_S32 display_frame_num;
    QUA_S32 chan_err_thr;
    QUA_S32 chan_strm_of_thr;
    QUA_S32 dec_mode;
    QUA_S32 dec_order_output;
    qua_video_format_t video_fmt;
    qua_compress_mode_t compress_mode;
} qua_vdec_chn_param_t;
```

【描述】

成员名称	描述
display_frame_num	显示帧数
chan_err_thr	通道错误率阈值（QM10XH 不支持）
chan_strm_of_thr	解码前丢帧阈值（不支持）
dec_mode	解码模式（QM10XH 固定 IP 模式）
dec_order_output	解码图像输出顺序（QM10XH 固定解码序）
video_fmt	解码图像数据格式（无效 固定线性）
compress_mode	解码图像压缩模式（无效）


qua_mm_venc_chn_attr_t

【说明】

编码通道参数属性定义。

【定义】

```
typedef struct qua_mm_venc_chn_attr_t {
    union {
        qua_venc_chn_attr_t chn_attr;
    };
};
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```

qua_venc_chn_cap_t chn_cap;
};
qua_video_channel_t video_chn;
} qua_mm_venc_chn_attr_t;

```

【描述】

成员名称	描述
chn_attr	通道属性，配置通道属性时使用
chn_cap	通道能力属性，创建通道时使用
video_chn	通道类型

qua_venc_chn_cap_t

【说明】

编码通道能力属性定义

【定义】

```

typedef struct qua_venc_chn_cap_t {
    qua_venc_type_t support_type;
    qua_size_t max_size;
    QUA_U32 bframe_num;
} qua_venc_chn_cap_t;

```

【描述】

成员名称	描述
support_type	编码类型,可以同时支持多种类型
max_size	最大编码幅面
bframe_num	对 B 帧的支持（不支持）

【注意】

仅 QM10XH 芯片平台需要配置

qua_venc_chn_attr_t

【说明】

编码通道属性定义

【定义】


```

typedef struct qua_venc_chn_attr_t {
    qua_venc_attr_t venc_attr;
    qua_venc_rc_attr_t venc_rc_attr;
} qua_venc_chn_attr_t;

```

【描述】

成员名称	描述
venc_attr	编码属性
venc_rc_attr	编码码率属性

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_venc_attr_t

【说明】

编码属性定义

【定义】

```
typedef struct qua_venc_attr_t {
    qua_venc_type_t enc_type;

    union {
        qua_venc_attr_h264_t attr_h264;
        qua_venc_attr_s264_t attr_s264;
        qua_venc_attr_mjpeg_t attr_mjpeg;
        qua_venc_attr_jpeg_t attr_jpeg;
        qua_venc_attr_h265_t attr_h265;
        qua_venc_attr_s265_t attr_s265;
    };

    QUA_BOOL enable_smart;
    QUA_U32 skip_mode;
} qua_venc_attr_t;
```

【描述】

成员名称	描述
enc_type	编码类型
attr_h264	普通 h264 编码属性
attr_s264	智能 h264 编码属性
attr_mjpeg	MJPEG 编码属性
attr_jpeg	JPEG 编码属性(无效)
attr_h265	普通 h265 编码属性
attr_s265	智能 h265 编码属性
enable_smart	使能智能编码(QM10XH 无效)
skip_mode	跳帧模式(QM10XH 无效)

qua_venc_attr_h264_t


【说明】

普通 h264 编码属性定义

【定义】

```
typedef struct qua_venc_attr_h264_t {
    QUA_U32 max_pic_width;
    QUA_U32 max_pic_height;

    QUA_U32 stream_buf_size;
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```
qua_h264_profile_t profile;
QUA_BOOL by_frame;
```

```
QUA_U32 pic_width;
QUA_U32 pic_height;
```

```
QUA_U32 i_frame_interval;
} qua_venc_attr_h264_t;
```

【描述】

成员名称	描述
max_pic_width	最大图像宽度(QM10XH 无效)
max_pic_height	最大图像高度(QM10XH 无效)
stream_buf_size	码流 buffer 大小(QM10XH 无效)
profile	编码 profile
by_frame	B 帧不支持
pic_width	图像宽度
pic_height	图像高度
i_frame_interval	I 帧间隔, GOP 长度

qua_venc_attr_s264_t

【说明】

智能 h264 编码属性定义

【定义】

```
typedef struct qua_venc_attr_s264_t {
    qua_h264_profile_t profile;
    QUA_U32 refresh_frame_interval;
    qua_size_t size;

    QUA_BOOL smart_en;
    QUA_BOOL texture_en;
    QUA_BOOL bgm_en;
    QUA_BOOL fresh_ltr_en;
    qua_gop_th_t gop_th;
    QUA_U32 bgm_chn;
} qua_venc_attr_s264_t;
```

【描述】

成员名称	描述
profile	编码 profile
refresh_frame_interval	刷新帧间隔
size	图像幅面
smart_en	智能编码使能
texture_en	纹理映射结果使能
bgm_en	背景建模结果使能



fresh_ltrf_en	使用刷新帧作为长期参考帧
gop_th	设置静止帧门限，决定 GOP 长度
bgm_chn	获取 BGM 信息的 BGM 通道号

qua_venc_attr_h265_t

【说明】

普通 h265 编码属性定义

【定义】

```
typedef struct qua_venc_attr_h265_t {
    QUA_U32 max_pic_width;
    QUA_U32 max_pic_height;

    QUA_U32 stream_buf_size;
    qua_h264_profile_t profile;
    QUA_BOOL by_frame;

    QUA_U32 pic_width;
    QUA_U32 pic_height;

    QUA_U32 i_frame_interval;
} qua_venc_attr_h265_t;
```

【描述】

成员名称	描述
max_pic_width	最大图像宽度(QM10XH 无效)
max_pic_height	最大图像高度(QM10XH 无效)
stream_buf_size	码流 buffer 大小(QM10XH 无效)
profile	编码 profile
by_frame	B 帧不支持
pic_width	图像宽度
pic_height	图像高度
i_frame_interval	I 帧间隔，GOP 长度


qua_venc_attr_s265_t

【说明】

智能 h265 编码属性定义

【定义】

```
typedef struct qua_venc_attr_s264_t {
    qua_h264_profile_t profile;
    QUA_U32 refresh_frame_interval;
    qua_size_t size;
}
```

	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

```

QUA_BOOL smart_en;
QUA_BOOL texture_en;
QUA_BOOL bgm_en;
QUA_BOOL fresh_ltrf_en;
qua_gop_th_t gop_th;
QUA_U32 bgm_chn;
} qua_venc_attr_s265_t;

```

【描述】

成员名称	描述
profile	编码 profile
refresh_frame_interval	刷新帧间隔
size	图像幅面
smart_en	智能编码使能
texture_en	纹理映射结果使能
bgm_en	背景建模结果使能
fresh_ltrf_en	使用刷新帧作为长期参考帧
gop_th	设置静止帧门限，决定 GOP 长度
bgm_chn	获取 BGM 信息的 BGM 通道号

qua_venc_rc_attr_t

【说明】

码控属性定义

【定义】

```


typedef struct qua_venc_rc_attr_t {
    qua_venc_rc_mode_t rc_mode;

    union {
        qua_venc_attr_h264_cbr_t attr_h264_cbr;
        qua_venc_attr_h264_vbr_t attr_h264_vbr;
        qua_venc_attr_h264_avbr_t attr_h264_avbr;
        qua_venc_attr_h264_fixqp_t attr_h264_fixqp;
        qua_venc_attr_h264_qvbr_t attr_h264_qvbr;
        qua_venc_attr_h264_qpmap_t attr_h264_qpMap;

        qua_venc_attr_mjpeg_cbr_t attr_mjpeg_cbr;
        qua_venc_attr_mjpeg_vbr_t attr_mjpeg_vbr;
        qua_venc_attr_mjpeg_fixqp_t attr_mjpeg_fixqp;

        qua_venc_attr_h265_cbr_t attr_h265_cbr;
        qua_venc_attr_h265_vbr_t attr_h265_vbr;
        qua_venc_attr_h265_avbr_t attr_h265_avbr;
        qua_venc_attr_h265_fixqp_t attr_h265_fixqp;
        qua_venc_attr_h265_qvbr_t attr_h265_qvbr;
    };
};

```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```
qua_venc_attr_h265_qpmap_t attr_h265_qpmap;
```

```
};
```

```
QUA_VOID_PTR rc_attr;
```

```
} qua_venc_rc_attr_t;
```

【描述】

成员名称	描述
mode	码控模式
attr_h264_cbr	H264 定码率
attr_h264_vbr	H264 变码率
attr_h264_avbr	H264 自适应变码率(QM10XH 不支持)
attr_h264_fixqp	H264 定QP
attr_h264_qvbr	H264 QVBR(QM10XH 不支持)
attr_h264_qpMap	H264 QPMAP(QM10XH 不支持)
attr_mjpeg_cbr	MJPEG 定码率
attr_mjpeg_vbr	MJPEG 变码率
attr_mjpeg_fixqp	MJPEG 定QP
attr_h265_cbr	H265 定码率
attr_h265_vbr	H265 变码率
attr_h265_avbr	H265 自适应变码率(QM10XH 不支持)
attr_h265_fixqp	H265 定QP
attr_h265_qvbr	H265 QVBR(QM10XH 不支持)
attr_h265_qpmap	H265 QPMAP(QM10XH 不支持)
rc_attr	扩展参数(QM10XH 不支持)

qua_venc_attr_h264_cbr_t

【说明】

H264 定码率码控属性定义

【定义】

```
typedef struct qua_venc_attr_h264_cbr_t {
```

```
    QUA_U32 gop;
```

```
    QUA_U32 stat_time;
```

```
    QUA_U32 src_frmrate;
```

```
    QUA_U32 dst_frmrate;
```

```
    QUA_U32 bitRate;
```

```
    QUA_U32 fluctuate_level;
```

```
    QUA_U32 init_qp;
```

```
    QUA_U32 max_rate_percent;
```


```
    QUA_U32 i_frm_max_bits;
```

```
    QUA_S32 ip_qp_delta;
```

```
    QUA_S32 i_bit_prop;
```

```
    QUA_S32 p_bit_prop;
```

```
} qua_venc_attr_h264_cbr_t;
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

【描述】

成员名称	描述
gop	gop 长度(QM10XH 无效)
stat_time	记录时间(QM10XH 无效)
src_frmrate	输入参考帧率，没有帧率控制的作用
dst_frmrate	输出参考帧率，没有帧率控制的作用
bitRate	目标码率，单位 bit
fluctuate_level	码率波动级别，从小到大，为从质量平稳优先到码率平稳优先。推荐值 0。取值范围: [0-6]
init_qp	初始化 QP。取值范围: [0-51]
max_rate_percent	最大码率，为目标码率的百分比。推荐值 200。取值范围: [120-800]
i_frm_max_bits	允许的最大 I 帧大小，单位为 bit。设为 0 为不限制 I 帧大小。
ip_qp_delta	P 帧最小 QP 与当前 I 帧 QP 的限制， $pqp \geq iqp + IP_QPDelta$ 。可以用来抑制呼吸效应等现象。推荐值 3。
i_bit_prop	I 帧的码率分配比例，推荐值 5
p_bit_prop	P 帧的码率分配比例，推荐值 1

qua_venc_attr_h264_vbr_t

【说明】

H264 变码率码控属性定义

【定义】

```
typedef struct qua_venc_attr_h264_vbr_t {
    QUA_U32 gop;
    QUA_U32 stat_time;
    QUA_U32 src_frmrate;
    QUA_U32 dst_frmrate;
    QUA_U32 max_bitRate;
    QUA_U32 max_qp;
    QUA_U32 min_qp;
    QUA_U32 min_iqp;

    QUA_U32 max_iqp;
    QUA_U32 init_qp;
    QUA_U32 max_rate_percent;
    QUA_U32 i_frm_max_bits;
    QUA_S32 ip_qp_delta;
    QUA_S32 i_bit_prop;
    QUA_S32 p_bit_prop;
    QUA_U32 fluctuate_level;
} qua_venc_attr_h264_vbr_t;
```

【描述】



成员名称	描述
gop	gop 长度(QM10XH 无效)
stat_time	记录时间(QM10XH 无效)
src_frmrate	输入参考帧率, 没有帧率控制的作用
dst_frmrate	输出参考帧率, 没有帧率控制的作用
max_bitRate	目标码率, 单位 bit
max_qp	P 帧最大 QP。取值范围: [0-51]
min_qp	P 帧最小 QP。取值范围: [0-51]
min_iqp	I 帧最小 QP。取值范围: [0-51]
max_iqp	I 帧最大 QP。取值范围: [0-51]
init_qp	初始化 QP。取值范围: [0-51]
max_rate_percent	最大码率, 为目标码率的百分比。推荐值 200。取值范围: [120-800]
i_frm_max_bits	允许的最大 I 帧大小, 单位为 bit。设为 0 为不限制 I 帧大小
ip_qp_delta	P 帧最小 QP 与当前 I 帧 QP 的限制, $pqp \geq iqp + IP_QPDelta$ 。可以用来抑制呼吸效应等现象。推荐值 3。
i_bit_prop	I 帧的码率分配比例, 推荐值 5
p_bit_prop	I 帧的码率分配比例, 推荐值 1
fluctuate_level	码率波动级别, 从小到大, 为从质量平稳优先到码率平稳优先。推荐值 0。取值范围: [0-6]

qua_venc_attr_h264_fixqp_t

【说明】


H264 定 QP 码控属性定义

【定义】

```
typedef struct qua_venc_attr_h264_fixqp_t {
    QUA_U32 gop;
    QUA_U32 src_frmrate;
    QUA_U32 dst_frmrate;
    QUA_U32 i_qp;
    QUA_U32 p_qp;
    QUA_U32 b_qp;
} qua_venc_attr_h264_fixqp_t;
```

【描述】

成员名称	描述
gop	gop 长度(QM10XH 无效)
src_frmrate	输入参考帧率, 没有帧率控制的作用
dst_frmrate	输出参考帧率, 没有帧率控制的作用
i_qp	I 帧 QP。取值范围: [0-51]
p_qp	P/B 帧 QP。取值范围: [0-51]
b_qp	不支持

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_venc_attr_h265_cbr_t

【说明】

H265 定码率码控属性定义

【定义】

```
typedef struct qua_venc_attr_h265_cbr_t {
    QUA_U32 gop;
    QUA_U32 stat_time;
    QUA_U32 src_frmrate;
    QUA_U32 dst_frmrate;
    QUA_U32 bitRate;
    QUA_U32 fluctuate_level;


    QUA_U32 init_qp;
    QUA_U32 max_rate_percent;
    QUA_U32 i_frm_max_bits;
    QUA_S32 ip_qp_delta;
    QUA_S32 i_bit_prop;
    QUA_S32 p_bit_prop;
} qua_venc_attr_h265_cbr_t;
```

【描述】

成员名称	描述
gop	gop 长度(QM10XH 无效)
stat_time	记录时间(QM10XH 无效)
src_frmrate	输入参考帧率，没有帧率控制的作用
dst_frmrate	输出参考帧率，没有帧率控制的作用
bitRate	目标码率，单位 bit
fluctuate_level	码率波动级别，从小到大，为从质量平稳优先到码率平稳优先。推荐值 0。取值范围: [0-6]
init_qp	初始化 QP。取值范围: [0-51]
max_rate_percent	最大码率，为目标码率的百分比。推荐值 200。取值范围: [120-800]
i_frm_max_bits	允许的最大 I 帧大小，单位为 bit。设为 0 为不限制 I 帧大小。
ip_qp_delta	P 帧最小 QP 与当前 I 帧 QP 的限制， $pqp \geq iqp + IP_QPDelta$ 。可以用来抑制呼吸效应等现象。推荐值 3。
i_bit_prop	I 帧的码率分配比例，推荐值 5
p_bit_prop	P 帧的码率分配比例，推荐值 1

qua_venc_attr_h265_vbr_t

【说明】

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

H265 变码率码控属性定义


【定义】

```
typedef struct qua_venc_attr_h265_vbr_t {
    QUA_U32 gop;
    QUA_U32 stat_time;
    QUA_U32 src_frmrate;
    QUA_U32 dst_frmrate;
    QUA_U32 max_bitRate;
    QUA_U32 max_qp;
    QUA_U32 min_qp;
    QUA_U32 min_iqp;

    QUA_U32 max_iqp;
    QUA_U32 init_qp;
    QUA_U32 max_rate_percent;
    QUA_U32 i_frm_max_bits;
    QUA_S32 ip_qp_delta;
    QUA_S32 i_bit_prop;
    QUA_S32 p_bit_prop;
    QUA_U32 fluctuate_level;
} qua_venc_attr_h265_vbr_t;
```

【描述】

成员名称	描述
gop	gop 长度(QM10XH 无效)
stat_time	记录时间(QM10XH 无效)
src_frmrate	输入参考帧率，没有帧率控制的作用
dst_frmrate	输出参考帧率，没有帧率控制的作用
max_bitRate	目标码率，单位 bit
max_qp	P 帧最大 QP。取值范围: [0-51]
min_qp	P 帧最小 QP。取值范围: [0-51]
min_iqp	I 帧最小 QP。取值范围: [0-51]
max_iqp	I 帧最大 QP。取值范围: [0-51]
init_qp	初始化 QP。取值范围: [0-51]
max_rate_percent	最大码率，为目标码率的百分比。推荐值 200。取值范围: [120-800]
i_frm_max_bits	允许的最大 I 帧大小，单位为 bit。设为 0 为不限制 I 帧大小
ip_qp_delta	P 帧最小 QP 与当前 I 帧 QP 的限制， $pqp \geq iqp + IP_QPDelta$ 。可以用来抑制呼吸效应等现象。推荐值 3。
i_bit_prop	I 帧的码率分配比例，推荐值 5
p_bit_prop	I 帧的码率分配比例，推荐值 1
fluctuate_level	码率波动级别，从小到大，为从质量平稳优先到码率平稳优先。推荐值 0。取值范围: [0-6]

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_venc_attr_h265_fixqp_t

【说明】

H265 定 QP 码控属性定义

【定义】

```
typedef struct qua_venc_attr_h265_fixqp_t {
    QUA_U32 gop;
    QUA_U32 src_frmrate;
    QUA_U32 dst_frmrate;
    QUA_U32 i_qp;
    QUA_U32 p_qp;
    QUA_U32 b_qp;
} qua_venc_attr_h265_fixqp_t;
```

【描述】

成员名称	描述
gop	gop 长度(QM10XH 无效)
src_frmrate	输入参考帧率，没有帧率控制的作用
dst_frmrate	输出参考帧率，没有帧率控制的作用
i_qp	I 帧 QP。取值范围: [0-51]
p_qp	P/B 帧 QP。取值范围: [0-51]
b_qp	不支持

qua_venc_chn_stat_t

【说明】

编码通道状态属性定义

【定义】

```
typedef struct qua_venc_chn_stat_t {
    QUA_U32 left_pics;
    QUA_U32 left_stream_bytes;
    QUA_U32 left_stream_frames;
    QUA_U32 cur_packs;
    QUA_U32 left_recv_pics;
    QUA_U32 left_enc_pics;

    QUA_U32 enc_pics;
    QUA_U32 lost_pics;
    QUA_U32 last_qp;
    QUA_U32 last_iqp;
    QUA_U32 bps;
} qua_venc_chn_stat_t;
```

【描述】



成员名称	描述
left_pics	(QM10XH 无效)
left_stream_bytes	(QM10XH 无效)
left_stream_frames	输出队列中的帧数
cur_packs	(QM10XH 无效)
left_recv_pics	(QM10XH 无效)
left_enc_pics	待编码帧数
enc_pics	已编码帧数
lost_pics	通道的累计丢帧数
last_qp	上帧 QP
last_iqp	上一 I 帧 QP
bps	实际码率

qua_venc_stream_t

【说明】

编解码流数据结构定义

【定义】


```
typedef struct qua_venc_stream_t {
    qua_stream_type_t stream_type;
    QUA_S32 chn_id;

    union {
        qua_h264_stream_t h264_stream;
        qua_h265_stream_t h265_stream;
        qua_jpeg_stream_t jpeg_stream;
    };

    union {
        qua_venc_stream_advance_info_h264_t advance_h264_info;
        qua_venc_stream_advance_info_jpeg_t advance_jpeg_info;
        qua_venc_stream_advance_info_h265_t advance_h265_info;
    };
} qua_venc_stream_t;
```

【描述】

成员名称	描述
stream_type	码流类型
chn_id	通道号
h264_stream	H264 码流
h265_stream	H265 码流
advance_h264_info	H264 码流统计信息
advance_h265_info	H265 码流统计信息

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_h264_stream_t

【说明】

H264 码流数据结构定义

【定义】

```
typedef struct qua_h264_stream_t {
    QUA_U8 *start;
    qua_enc_frame_type_t frame_type;
    QUA_U32 length;
    QUA_U64 time_stamp;
    QUA_U32 nalu_cnt;
    qua_enc_stream_nalu_t nalu[QUA_MAX_NALU_CNT];
    qua_enc_ref_type_t ref_type;
} qua_h264_stream_t;
```

【描述】

成员名称	描述
start	码流地址
frame_type	帧类型
length	码流长度
time_stamp	时间戳
nalu_cnt	NALU 数量
nalu	NALU 属性
ref_type	帧参考类型

qua_h265_stream_t

【说明】

H265 码流数据结构定义

【定义】

```
typedef struct qua_h265_stream_t {
    QUA_U8 *start;
    qua_enc_frame_type_t frame_type;
    QUA_U32 length;
    QUA_U64 time_stamp;
    QUA_U32 nalu_cnt;
    qua_enc_stream_nalu_t nalu[QUA_MAX_NALU_CNT];
    qua_enc_ref_type_t ref_type;
} qua_h265_stream_t;
```

【描述】

成员名称	描述
start	码流地址
frame_type	帧类型



length	码流长度
time_stamp	时间戳
nal_cnt	NALU 数量
nal	NALU 属性
ref_type	帧参考类型

qua_venc_stream_advance_info_h264_t

【说明】

H264 码流统计信息数据结构定义

【定义】

```
typedef struct qua_venc_stream_advance_info_h264_t {
    QUA_U32 mean_qp;
    QUA_U32 residual_bit_num;
    QUA_U32 head_bit_num;
    QUA_U32 madi_val;
    QUA_U32 madp_val;
    QUA_U64 psnr_val;
    QUA_U64 distortion;
    QUA_U32 mseLcu_cnt;
    QUA_U32 mse_sum;
    qua_venc_sse_info_t sst_info[8];
    QUA_U32 qp_hstgrm[52];
} qua_venc_stream_advance_info_h264_t;
```


【描述】

成员名称	描述
mean_qp	编码当前帧平均 QP
residual_bit_num	编码当前帧残差 bit 数
head_bit_num	编码当前帧头信息 bit 数
madi_val	编码当前帧空域纹理复杂度 Madi 值
madp_val	编码当前帧时域运动复杂度 Madp 值
psnr_val	psnr, 因为驱动定点化原因有一些精度损失
distortion	失真度, 用户可以用此值得到准确 psnr, $psnr = 10 \log_{10}(w * h * 255 * 255 / distortion)$
mseLcu_cnt	编码当前帧中的 LCU 个数
mse_sum	编码当前帧中的 MSE(均方差)
sst_info	(QM10XH 无效)
qp_hstgrm	(QM10XH 无效)

qua_venc_stream_advance_info_h265_t

【说明】

H265 码流统计信息数据结构定义

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

【定义】

```
typedef struct qua_venc_stream_advance_info_h265_t {
    QUA_U32 mean_qp;
    QUA_U32 residual_bit_num;
    QUA_U32 head_bit_num;
    QUA_U32 madi_val;
    QUA_U32 madp_val;
    QUA_U64 psnr_val;
    QUA_U64 distortion;
    QUA_U32 mseLcu_cnt;
    QUA_U32 mse_sum;
    qua_venc_sse_info_t sst_info[8];
    QUA_U32 qp_hstgrm[52];
} qua_venc_stream_advance_info_h265_t;
```

【描述】

成员名称	描述
mean_qp	编码当前帧平均 QP
residual_bit_num	编码当前帧残差 bit 数
head_bit_num	编码当前帧头信息 bit 数
madi_val	编码当前帧空域纹理复杂度 Madi 值
madp_val	编码当前帧时域运动复杂度 Madp 值
psnr_val	psnr, 因为驱动定点化原因有一些精度损失
distortion	失真度, 用户可以用此值得到准确 psnr, $psnr = 10\log_{10}(w*h*255*255/distortion)$
mseLcu_cnt	编码当前帧中的 LCU 个数
mse_sum	编码当前帧中的 MSE(均方差)
sst_info	(QM10XH 无效)
qp_hstgrm	(QM10XH 无效)

qua_roi_map_t

【说明】

ROI Map 结构定义

【定义】

```
typedef struct qua_roi_map_t {
    QUA_S32 qp[7];
    QUA_BOOL is_delta_qp[7];
    QUA_U32 size;
    QUA_U64 roi_addr;
} qua_roi_map_t;
```

【描述】

成员名称	描述
qp	对应 level-1 使用的 QP。取值范围: [绝对 QP:0-51, 相对 QP:-51-51]



is_delta_qp	对应 level-1 的设置，0:绝对 QP,使用用户 QP 1:相对 QP,在 RC QP 基础上加减 Delta 值。取值范围: [0-1]
size	map 长度, w/64 * h/64 * 16, w/h 为 64 对齐后的宽高
roi_addr	map 基地址

qua_roi_t

【说明】

ROI 属性定义

【定义】

```
typedef struct qua_roi_t {
    QUA_U32 enable;
    QUA_S32 qp;
    qua_rect_t rect;
    QUA_U32 level;
    QUA_BOOL is_delta_qp;
} qua_roi_t; 【描述】
```

成员名称	描述
enable	ROI 使能。取值范围: [0-1]
qp	对应 level 使用的 QP。取值范围: [绝对 QP:0-51, 相对 QP:-51-51]
rect	ROI 区域
level	ROI level,0 代表不使用 ROI。取值范围: [0-7]
is_delta_qp	0:绝对 QP,使用用户 QP 1:相对 QP,在 RC QP 基础上加减 Delta 值。取值范围: [0-1]

3.3.2 解码设备函数接口

init

【说明】

解码设备初始化。

QUA_S32 (*init)();			
参数名称	输入/输出	描述	返回值
			成功: QUA_SUCCESS; 失败: 其它值

exit

【说明】

解码设备反初始化。

QUA_S32 (*exit)();



参数名称	输入/输出	描述	返回值
			成功: QUA_SUCCESS; 失败: 其它值

3.3.3 解码通道函数接口

start_chn

【说明】

解码器开始接收用户发送的码流。

QUA_S32 (*start_chn)(QUA_S32 chn_id)			
参数名称	输入/输出	描述	返回值
chn_id	输入	解码通道号	成功: QUA_SUCCESS; 失败: 其它值

【注意】

- 启动接收码流前必须保证通道已创建。
- 启动接收码流之后，才能发送码流数据。

stop_chn

【说明】

解码器停止接收用户发送的码流。

QUA_S32 (*stop_chn)(QUA_S32 chn_id)			
参数名称	输入/输出	描述	返回值
chn_id	输入	解码通道号	成功: QUA_SUCCESS; 失败: 其它值

【注意】

停止接收码流前必须保证通道已创建。

reset_chn


【说明】

复位视频解码通道。

QUA_S32 (*reset_chn)(QUA_S32 chn_id)			
参数名称	输入/输出	描述	返回值
chn_id	输入	解码通道号	成功: QUA_SUCCESS; 失败: 其它值

【注意】

复位前必须保证通道已创建。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

get_chn_fd

【说明】

获取视频解码通道对应的文件描述符。

QUA_S32 (*get_chn_fd)(QUA_S32 chn_id)			
参数名称	输入/输出	描述	返回值
chn_id	输入	解码通道号	成功: QUA_SUCCESS; 失败: 其它值

【注意】

获取得到通道 fd,可使用 select 接口监听解码通道

send_stream

【说明】

向视频解码通道发送码流数据。

QUA_S32 (*send_stream)(QUA_S32 chn_id, const qua_vdec_stream_t *packet, QUA_S32 timeout_millis)			
参数名称	输入/输出	描述	返回值
chn_id	输入	解码通道号	成功: QUA_SUCCESS; 失败: 其它值
packet	输入	码流数据结构指针	
timeout_millis	输入	超时时间	

【注意】

- 超时时间设置-1 为阻塞模式
- “QM10XH” 芯片平台不能发送 end_of_stream 为 0 的空码流包（码流长度为 0 或者码流地址为空）

get_frame

【说明】

获取视频解码通道的解码图像。


QUA_S32 (*release_frame)(QUA_S32 chn_id, qua_video_frame_info_t *frame_info)			
参数名称	输入/输出	描述	返回值
chn_id	输入	解码通道号	成功: QUA_SUCCESS; 失败: 其它值
frame_info	输出	视频帧信息数据结构指针	

【注意】

- 超时时间设置-1 为阻塞模式
- “QM10XH” 芯片平台不能发送 end_of_stream 为 0 的空码流包（码流长度为 0 或者码流地址为空）

release_frame

【说明】

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

释放视频解码通道的图像。

QUA_S32 (*release_frame)(QUA_S32 chn_id, qua_video_frame_info_t *frame_info)			
参数名称	输入/输出	描述	返回值
chn_id	输入	解码通道号	成功: QUA_SUCCESS;
frame_info	输入	视频帧信息数据结构指针	失败: 其它值

【注意】

与 get_frame 接口配对使用，视频帧数据结构不可修改

get_chn_attr

【说明】

获取视频解码通道属性。

QUA_S32 (*get_chn_attr)(QUA_S32 chn_id, qua_vdec_chn_attr_t *chn_attr)			
参数名称	输入/输出	描述	返回值
chn_id	输入	解码通道号	成功: QUA_SUCCESS;
chn_attr	输出	解码通道属性数据结构指针	失败: 其它值

query_chn

【说明】

查询解码通道状态。

QUA_S32 (*query_chn)(QUA_S32 chn_id, qua_vdec_chn_stat_t *chn_stat);			
参数名称	输入/输出	描述	返回值
chn_id	输入	解码通道号	成功: QUA_SUCCESS;
chn_stat	输出	解码通道状态数据结构指针	失败: 其它值

get_chn_param

【说明】

获取解码通道参数。

QUA_S32 (*get_chn_param)(QUA_S32 chn_id, qua_vdec_chn_param_t *chn_param);			
参数名称	输入/输出	描述	返回值
chn_id	输入	解码通道号	成功: QUA_SUCCESS;
chn_param	输出	解码通道参数数据结构指针	失败: 其它值

【注意】

JPEG/MJPEG 不支持该接口

set_chn_param

【说明】

设置解码通道参数。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

QUA_S32 (*set_chn_param)(QUA_S32 chn_id, qua_vdec_chn_param_t* chn_param);			
参数名称	输入/输出	描述	返回值
chn_id	输入	解码通道号	成功: QUA_SUCCESS; 失败: 其它值
chn_param	输入	解码通道参数数据结构指针	

【注意】

JPEG/MJPEG 不支持该接口

3.3.4 编码设备函数接口

init

【说明】

编码设备初始化。

QUA_S32 (*init)();			
参数名称	输入/输出	描述	返回值
			成功: QUA_SUCCESS; 失败: 其它值

deinit

【说明】

编码设备反初始化。

QUA_S32 (*deinit)();			
参数名称	输入/输出	描述	返回值
			成功: QUA_SUCCESS; 失败: 其它值

3.3.5 编码通道函数接口

reset_chn

【说明】

复位视频编码通道。

QUA_S32 (*reset_chn)(QUA_S32 chn_id);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值

start_recv_pic

【说明】

编码器开始接收用户发送的码流。



QUA_S32 (*start_recv_pic)(QUA_S32 chn_id);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值

stop_recv_pic

【说明】

编码器停止接收用户发送的码流。

QUA_S32 (*stop_recv_pic)(QUA_S32 chn_id);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值

【注意】

停止接收码流前必须保证通道已创建。

query_chn

【说明】

查询编码通道状态。

QUA_S32 (*query_chn)(QUA_S32 chn_id, qua_venc_chn_stat_t *chn_stat);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值
chn_stat	输出	编码通道状态数据结构指针	

set_chn_attr

【说明】

设置编码通道属性。

QUA_S32 (*set_chn_attr)(QUA_S32 chn_id, const qua_venc_chn_attr_t *chn_attr);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值
chn_attr	输入	编码通道属性指针	

get_chn_attr

【说明】

获取编码通道属性。

QUA_S32 (*get_chn_attr)(QUA_S32 chn_id, const qua_venc_chn_attr_t *chn_attr);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS;



chn_attr	输出	编码通道属性指针	失败：其它值
----------	----	----------	--------

get_stream

【说明】

获取编码码流。

QUA_S32(*get_stream)(QUA_S32chn_id,qua_venc_stream_t*packet,QUA_S32 timeout_millis);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
packet	输出	编码码流指针	
timeout_millis	输入	超时时间	

【注意】

需要成对调用 release_stream

release_stream

【说明】

释放编码码流。

QUA_S32(*release_stream)(QUA_S32chn_id,qua_venc_stream_t*packet);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
packet	输入	编码码流指针	

send_frame

【说明】

向编码通道提交图像数据进行编码。


QUA_S32(*send_frame)(QUA_S32chn_id,qua_video_frame_info_t*frame_info,QUA_S32timeout_millis);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
frame_info	输入	编码码流指针	
timeout_millis	输入	超时时间	

set_rotate

【说明】

设置编码旋转属性。

QUA_S32(*set_rotate)(QUA_S32chn_id,const qua_rotate_t rotate_ops);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
rotate_ops	输入	旋转角度	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

get_rotate

【说明】

获取编码旋转属性。

QUA_S32 (*get_rotate)(QUA_S32 chn_id, qua_rotate_t *rotate_ops);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
rotate_ops	输入	旋转角度指针	

check_in_addr

【说明】

检查 VENC 输入缓存地址是否被使用。

QUA_S32 (*check_in_addr)(QUA_S32 chn_id, QUA_U64 in_addr, QUA_U32 size, QUA_S32 timeout_millis, QUA_U32 *is_idle);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
in_addr	输入	输入物理地址	
size	输入	大小	
timeout_millis	输入	超时	
is_idle	输出	空闲状态	

set_roi_map

【说明】

设置 ROI Map。

QUA_S32 (*set_roi_map)(QUA_S32 chn_id, const qua_roi_map_t *roi_map);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
roi_map	输入	roi map 结构体指针	

【注意】

- JPEG、MJPEG 通道不支持设置
- ROI 通过位图形式设置，因此不支持获取 ROI 配置

set_roi_cfg

【说明】

设置 ROI 属性。

QUA_S32 (*set_roi_cfg)(QUA_S32 chn_id, const qua_roi_t *roi_cfg);			
参数名称	输入/输出	描述	返回值



chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
roi_cfg	输出	roi_cfg 属性指针	

【注意】

- JPEG、MJPEG 通道不支持设置
- ROI 通过位图形式设置，因此不支持获取 ROI 配置

clear_roi**【说明】**

清除之前设置的 ROI 配置。

QUA_S32 (*clear_roi)(QUA_S32 chn_id);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值

【注意】

JPEG、MJPEG 通道不支持设置

set_rc_attr**【说明】**

设置码率控制参数。

QUA_S32 (*set_rc_attr)(QUA_S32 chn_id, const qua_venc_rc_attr_t *rc_attr);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
rc_attr	输入	码率控制参数指针	

【注意】

推荐用户使用该接口来更新码率参数，和 set_chn_attr 相比可以减少重新绑定，设置通道属性等参数。

get_rc_attr**【说明】**

获取码率控制参数。

QUA_S32 (*get_rc_attr)(QUA_S32 chn_id, qua_enc_param_cmd_t cmd, const qua_enc_param_union_t *param);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
cmd	输入	参数 id	
param	输出	码率控制参数指针	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

set_enc_param

【说明】

配置一些编码参数。

QUA_S32 (*get_enc_param)(QUA_S32 chn_id, qua_enc_param_cmd_t cmd, const qua_enc_param_union_t *param);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
cmd	输入	参数 id	
param	输入	编码参数参数指针	

get_enc_param

【说明】

获取一些编码参数。

QUA_S32 (*get_enc_param)(QUA_S32 chn_id, const qua_venc_rc_attr_t *rc_attr);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
rc_attr	输出	编码参数参数指针	

set_h264_entropy

【说明】

设置熵编码类型。

QUA_S32 (*set_h264_entropy)(QUA_S32 chn_id, const qua_h264_entropy_t *entropy);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
entropy	输入	熵编码类型指针	

【注意】

仅 H264 编码支持该接口

get_h264_entropy


【说明】

获取熵编码类型。

QUA_S32 (*get_h264_entropy)(QUA_S32 chn_id, const qua_h264_entropy_t *entropy);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功：QUA_SUCCESS; 失败：其它值
entropy	输出	熵编码类型指针	

【注意】

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

仅 H264 编码支持该接口

set_h264_dblk

【说明】

设置 Deblocking 参数。

QUA_S32 (*set_h264_dblk)(QUA_S32 chn_id, const qua_h264_dblk_t *dblk);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS;
dblk	输入	deblock 参数指针	失败: 其它值

【注意】

仅 H264 编码支持该接口

get_h264_dblk

【说明】

获取 Deblocking 参数。

QUA_S32 (*get_h264_dblk)(QUA_S32 chn_id, const qua_h264_dblk_t *dblk);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS;
dblk	输出	deblock 参数指针	失败: 其它值

【注意】

仅 H264 编码支持该接口

set_h265_entropy

【说明】

设置熵编码类型。

QUA_S32 (*set_h265_entropy)(QUA_S32 chn_id, const qua_h265_entropy_t *entropy);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS;
entropy	输入	熵编码类型指针	失败: 其它值

【注意】

仅 H265 编码支持该接口

get_h265_entropy

【说明】

获取熵编码类型。

QUA_S32 (*get_h265_entropy)(QUA_S32 chn_id, const qua_h265_entropy_t *entropy);			
参数名称	输入/输出	描述	返回值

	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值
entropy	输出	熵编码类型指针	

【注意】

仅 H265 编码支持该接口

set_h265_dblk

【说明】

设置 Deblocking 参数。

QUA_S32 (*set_h265_dblk)(QUA_S32 chn_id, const qua_h265_dblk_t *dblk);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值
dblk	输入	deblock 参数指针	

【注意】

仅 H265 编码支持该接口

get_h265_dblk

【说明】

获取 Deblocking 参数。

QUA_S32 (*get_h265_dblk)(QUA_S32 chn_id, const qua_h265_dblk_t *dblk);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值
dblk	输出	deblock 参数指针	

【注意】

仅 H265 编码支持该接口

set_debreath_effect

【说明】

设置去呼吸效应。

QUA_S32 (*set_debreath_effect)(QUA_S32 chn_id, const qua_debreath_t *debreath);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值
debreath	输入	debreath 参数指针	

get_debreath_effect

【说明】

获取去呼吸效应使能状态。

QUA_S32 (*get_debreath_effect)(QUA_S32 chn_id, qua_debreath_t *debreath);			
---	--	--	--

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值
debreath	输出	debreath 参数指针	

clear_yuv_queue

【说明】

清空编码 yuv 队列 & 输出码流队列。

QUA_S32 (*clear_yuv_queue)(QUA_S32 chn_id);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值

request_idr

【说明】

请求 IDR 帧。

QUA_S32 (*request_idr)(QUA_S32 chn_id);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值

set_max_stream_cnt

【说明】

设置 JPEG 编码通道最多编码的缓存帧数。


QUA_S32 (*set_max_stream_cnt)(QUA_S32 chn_id, QUA_U32 max_strm_cnt);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值

get_max_stream_cnt

【说明】

获取 JPEG 编码通道最多编码的缓存帧数。

QUA_S32 (*get_max_stream_cnt)(QUA_S32 chn_id, QUA_U32 max_strm_cnt);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

get_jpeg_param

【说明】

获取 JPEG 编码通的 QP 值等。

QUA_S32 (*get_jpeg_param)(QUA_S32 chn_id, qua_venc_param_jpeg_t *jpeg_param);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值
jpeg_param	输出	jpeg 参数指针	

set_jpeg_param

【说明】

设置 JPEG 编码通的 QP 值等。

QUA_S32 (*set_jpeg_param)(QUA_S32 chn_id, qua_venc_param_jpeg_t *jpeg_param);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值
jpeg_param	输入	jpeg 参数指针	

set_crop

【说明】

设置 JPEG 编码通 crop 参数。

QUA_S32 (*set_crop)(QUA_S32 chn_id, const qua_venc_crop_cfg_t *crop_cfg);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值
crop_cfg	输入	crop 参数指针	

get_crop

【说明】

获取 JPEG 编码通 crop 参数。

QUA_S32 (*get_crop)(QUA_S32 chn_id, qua_venc_crop_cfg_t *crop_cfg);			
参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值
crop_cfg	输出	crop 参数指针	

get_chn_fd

【说明】

获取 JPEG 编码通道文件句柄



QuaMM 多媒体接口使用说明

生效日期
Effective Date

2024.08.01


版本
Rev.

1.1.0

QUA_S32 (*get_chn_fd)(QUA_S32 chn_id);

参数名称	输入/输出	描述	返回值
chn_id	输入	编码通道号	成功: QUA_SUCCESS; 失败: 其它值

Quaming Confidential

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

3.4 音频模块接口

音频模块接口提供基于 QuaMM 开发的音频接口，实现音频输入输出、编解码功能。所在的头文件路径：`include/video`。

3.4.1 数据结构

qua_mm_audio_device_t

【说明】


音频设备的通用表示。

【定义】

```
typedef struct qua_mm_audio_device_t {
    struct qua_mm_device_t parent;
    QUA_S32 dev_num;
    QUA_S32 (*play_init)(QUA_U32 id, qua_audio_cfg_arg* param);
    QUA_S32 (*play_deinit)(QUA_U32 id);
    QUA_S32 (*play_start)(QUA_U32 id);
    QUA_S32 (*play_stop)(QUA_U32 id);
    QUA_S32 (*play_volume)(QUA_U32 id, QUA_U32 vol_type, QUA_U32 vol);
    QUA_S32 (*write_frame)(QUA_U32 id, QUA_VOID_PTR buf, QUA_U32 size);
    QUA_S32 (*capture_init)(QUA_U32 id, qua_audio_cfg_arg* param);
    QUA_S32 (*capture_deinit)(QUA_U32 id);
    QUA_S32 (*capture_start)(QUA_U32 id);
    QUA_S32 (*capture_stop)(QUA_U32 id);
    QUA_S32 (*capture_volume)(QUA_U32 id, QUA_U32 vol_type, QUA_U32 vol);
    QUA_S32 (*read_frame)(QUA_U32 id, QUA_VOID_PTR buf, QUA_U32 size);
    QUA_S32 (*exit)(QUA_U32 id);
} qua_mm_audio_device_t;
```

【描述】

成员名称	描述
parent	音频模块的模块指针，应用层无需关注
dev_num	设备号，默认配置为 0
play_init	播放资源初始化
play_deinit	播放资源释放
play_start	启动播放
play_stop	停止播放
play_volume	设置播放音量
write_frame	写入播放 buf 数据
capture_init	录音资源初始化
capture_deinit	录音资源释放
capture_start	启动录音
capture_stop	停止录音

	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

capture_volume	设置录音音量
read_frame	读取录音 buf 数据
exit	退出音频播放/录音

以上表格里的函数指针成员放在 3.4.2 节“音频输入输出函数接口”里做具体说明。

qua_audio_cfg_arg

【说明】

音频配置参数。

【定义】

```
typedef struct _qua_audio_cfg_arg {
    int enc_type;
    int io_type;
    int frame_size;
    int frame_cnt;
    int channels;
    int channel_bit;
    int rate;
} qua_audio_cfg_arg;
```

【描述】

成员名称	描述
enc_type	音频编码方式
io_type	IO 类型， 播放可配置成 QUA_LINE_HPOUT 录音可配置成 QUA_MIC_IN
frame_size	分配每个缓存 buf 大小 计算公式： frame_size = 采样率*时长*位深度*声道数 以采样率 48000，时长 20ms，位深度 2，声道数 2 为例 frame_size = 48000 * 20 / 1000 * 2 * 2 = 3840
frame_cnt	分配缓存 buf 数量
channels	声道数量
channel_bit	采样精度，默认应配置为 16 6870 如支持 HDMI 须配置为 32
rate	采样率

qua_mm_adec_device_t

【说明】

音频解码设备的通用表示。

【定义】

```
typedef struct qua_mm_adec_device {
```

	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

```

struct qua_mm_device_t parent;
QUA_S32 (*bind_init)(QUA_U32 mod_id);
QUA_S32 (*bind_exit)(QUA_U32 mod_id);
} qua_mm_adec_device_t;

```

【描述】

成员名称	描述
parent	音频解码设备的模块指针，应用层无需关注
bind_init	音频解码通道绑定初始化
bind_exit	音频解码通道绑定去初始化

以上表格里的函数指针成员放在 3.4.3 节“音频解码函数接口”里做具体说明。

qua_mm_adec_channel_t

【说明】

音频解码通道的通用表示。

【定义】

```

typedef struct qua_mm_adec_channel {
    struct qua_mm_channel_t parent;
    QUA_S32 (*send_stream)(QUA_ADEC_CHN chn_id, qua_audio_stream_t *stream, QUA_S32 timeout);
    QUA_S32 (*clear_chn_buf)(QUA_ADEC_CHN chn_id);
    QUA_S32 (*get_frame)(QUA_ADEC_CHN chn_id, qua_audio_frame_t *frame, QUA_S32 timeout);
    QUA_S32 (*release_frame)(QUA_ADEC_CHN chn_id, qua_audio_frame_t *frame);
    QUA_S32 (*ext_api)(QUA_ADEC_CHN chn_id, QUA_VOID_PTR attr);
} qua_mm_adec_channel_t;

```

【描述】

成员名称	描述
parent	音频解码通道的模块指针，应用层无需关注
send_stream	发送音频解码前数据到解码器
clear_chn_buf	清除音频解码通道缓存数据
get_frame	获取音频解码后数据
release_frame	释放音频解码通道数据
ext_api	音频解码模块扩展 API

以上表格里的函数指针成员放在 3.4.3 节“音频解码函数接口”里做具体说明。

qua_mm_aenc_device_t

【说明】


音频编码设备的通用表示。

【定义】

```

typedef struct qua_mm_aenc_device {
    struct qua_mm_device_t parent;
    QUA_S32 (*bind_init)(QUA_U32 mod_id);
    QUA_S32 (*bind_exit)(QUA_U32 mod_id);
} qua_mm_aenc_device_t;

```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```
} qua_mm_aenc_device_t;
```

【描述】

成员名称	描述
parent	音频编码设备的模块指针，应用层无需关注
bind_init	音频编码通道绑定初始化
bind_exit	音频编码通道绑定去初始化

以上表格里的函数指针成员放在 3.4.4 节“音频编码函数接口”里做具体说明

qua_mm_aenc_channel_t

【说明】

音频编码通道的通用表示。

【定义】

```
typedef struct qua_mm_aenc_channel {
    struct qua_mm_channel_t parent;
    QUA_S32 (*get_stream)(QUA_AENC_CHN chn_id, qua_audio_stream_t *stream, QUA_S32 timeout);
    QUA_S32 (*clear_chn_buf)(QUA_AENC_CHN chn_id);
    QUA_S32 (*send_frame)(QUA_AENC_CHN chn_id, qua_audio_frame_t *frame, QUA_S32 timeout);
    QUA_S32 (*release_frame)(QUA_AENC_CHN chn_id, qua_audio_frame_t *frame);
    QUA_S32 (*ext_api)(QUA_AENC_CHN chn_id, QUA_VOID_PTR attr);
} qua_mm_aenc_channel_t;
```

【描述】

成员名称	描述
parent	音频编码通道的模块指针，应用层无需关注
get_stream	获取编码通道编码后码流数据
clear_chn_buf	清除音频编码通道缓存数据
send_frame	发送音频编码前数据给编码器
release_frame	释放音频编码通道数据
ext_api	音频编码器扩展 API

以上表格里的函数指针成员放在 3.4.4 节“音频编码函数接口”里做具体说明


qua_audio_stream_t

【说明】

音频流的通用表示。

【定义】

```
typedef struct qua_audio_stream {
    QUA_U8* stream;
    QUA_U32 phy_addr;
    QUA_U32 len;
    QUA_U64 pts;
    QUA_U32 seq;
    qua_sample_rate_t sample_rate;
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```

qua_bit_width_t bit_width;
qua_audio_channel_t channel;
} qua_audio_stream_t;

```

【描述】

成员名称	描述
stream	stream 指针地址
phy_addr	stream 物理地址
len	stream 长度
pts	时间戳
seq	帧序号
sample_rate	采样率
bit_width	采样精度
channel	声道数量

qua_audio_frame_t

【说明】

音频帧的通用表示。

【定义】


```

typedef struct qua_audio_frame {
    QUA_VOID_PTR vir_addr;
    QUA_U32 phy_addr;
    QUA_U32 seq;
    QUA_U32 len;
    QUA_U32 timeout;
    QUA_U64 pts;
    qua_sample_rate_t sample_rate;
    qua_bit_width_t bit_width;
    qua_audio_channel_t channel;
    QUA_U32 exp_id;
} qua_audio_frame_t;

```

【描述】

成员名称	描述
vir_addr	帧虚拟地址
phy_addr	帧物理地址
seq	帧序号
len	帧长度
timeout	超时时间
pts	时间戳
sample_rate	采样率
bit_width	采样精度
channel	声道数量
exp_id	扩展参数

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

3.4.2 音频输入输出函数接口

play_init

【说明】

播放资源初始化。

QUA_S32 (*play_init)(QUA_U32 id, qua_audio_cfg_arg* param)			
参数名称	输入/输出	描述	返回值
id	输入	音频输出设备 ID	成功: QUA_SUCCESS;
param	输入	音频配置参数	失败: 其它值

play_deinit

【说明】

播放资源释放。

QUA_S32 (*play_deinit)(QUA_U32 id)			
参数名称	输入/输出	描述	返回值
id	输入	音频输出设备 ID	成功: QUA_SUCCESS;
			失败: 其它值

play_start

【说明】

启动播放。


QUA_S32 (*play_start)(QUA_U32 id)			
参数名称	输入/输出	描述	返回值
id	输入	音频输出设备 ID	成功: QUA_SUCCESS;
			失败: 其它值

play_stop

【说明】

停止播放。

QUA_S32 (*play_stop)(QUA_U32 id)			
参数名称	输入/输出	描述	返回值
id	输入	音频输出设备 ID	成功: QUA_SUCCESS;
			失败: 其它值

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

play_volume

【说明】

设置播放音量。

QUA_S32 (*play_volume)(QUA_U32 id, QUA_U32 vol_type, QUA_U32 vol)			
参数名称	输入/输出	描述	返回值
id	输入	音频输出设备 ID	成功: QUA_SUCCESS; 失败: 其它值
vol_type	输入	音量类型	
vol	输入	音量大小	

【注意】

- “mc6870”芯片平台不支持音量设置
- “qm10xh”和”mc331x”支持 QUA_VOLUME_TYPE_ANA, QUA_VOLUME_TYPE_DIG 音量设置
其中:

“qm10xh” QUA_VOLUME_TYPE_ANA 音量范围【播放】 $0 \leq \text{value} \leq 31$

“qm10xh” QUA_VOLUME_TYPE_DIG 音量范围【播放】 $0x1B \leq \text{value} \leq 0x7F$

“mc331x” QUA_VOLUME_TYPE_ANA 音量范围【播放】 $0 \leq \text{value} \leq 31$

“mc331x” QUA_VOLUME_TYPE_DIG 音量范围【播放】 $0 \leq \text{value} \leq 0x80$

write_frame

【说明】

写入播放 buf 数据。


QUA_S32 (*write_frame)(QUA_U32 id, QUA_VOID_PTR buf, QUA_U32 size)			
参数名称	输入/输出	描述	返回值
id	输入	音频输出设备 ID	成功: QUA_SUCCESS; 失败: 其它值
buf	输入	播放 buf 数据	
size	输入	数据大小	

capture_init

【说明】

录音资源初始化。

QUA_S32 (*capture_init)(QUA_U32 id, qua_audio_cfg_arg* param)			
参数名称	输入/输出	描述	返回值
id	输入	音频输入设备 ID	成功: QUA_SUCCESS; 失败: 其它值
param	输入	音频配置参数	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

capture_deinit

【说明】

录音资源释放。

QUA_S32 (*capture_deinit)(QUA_U32 id)			
参数名称	输入/输出	描述	返回值
id	输入	音频输入设备 ID	成功: QUA_SUCCESS; 失败: 其它值

capture_start

【说明】

启动录音。

QUA_S32 (*capture_start)(QUA_U32 id)			
参数名称	输入/输出	描述	返回值
id	输入	音频输入设备 ID	成功: QUA_SUCCESS; 失败: 其它值

capture_stop

【说明】

停止录音。

QUA_S32 (*capture_stop)(QUA_U32 id)			
参数名称	输入/输出	描述	返回值
id	输入	音频输入设备 ID	成功: QUA_SUCCESS; 失败: 其它值

capture_volume


【说明】

设置录音音量。

QUA_S32 (*capture_volume) (QUA_U32 id, QUA_U32 vol_type, QUA_U32 vol)			
参数名称	输入/输出	描述	返回值
id	输入	音频输入设备 ID	成功: QUA_SUCCESS; 失败: 其它值
vol_type	输入	音量类型	
vol	输入	音量大小	

【注意】

- “mc6870”芯片平台不支持音量设置
- “qm10xh”和“mc331x”支持 QUA_VOLUME_TYPE_ANA, QUA_VOLUME_TYPE_DIG 音量设置

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

其中：

“qm10xh” QUA_VOLUME_TYPE_ANA 音量范围【录音】 $0 \leq \text{value} \leq 31$

“qm10xh” QUA_VOLUME_TYPE_DIG 音量范围【录音】 $0x1B \leq \text{value} \leq 0x7F$

“mc331x” QUA_VOLUME_TYPE_ANA 音量范围【录音】 $0 \leq \text{value} \leq 3$

“mc331x” QUA_VOLUME_TYPE_DIG 音量范围【录音】 $0 \leq \text{value} \leq 31$

read_frame

【说明】

读取录音 buf 数据。

QUA_S32 (*write_frame)(QUA_U32 id, QUA_VOID_PTR buf, QUA_U32 size)			
参数名称	输入/输出	描述	返回值
id	输入	音频输入设备 ID	成功: QUA_SUCCESS; 失败: 其它值
buf	输入/输出	录音 buf 数据	
size	输入	数据大小	

exit

【说明】

退出音频播放/录音。

QUA_S32 (*exit)(QUA_U32 id)			
参数名称	输入/输出	描述	返回值
id	输入	音频输入/输出设备 ID	成功: QUA_SUCCESS; 失败: 其它值

【注意】

对于同一个设备 ID 播放录制共存的情况，需确保播放录制都结束后调用

3.4.3 音频解码函数接口

bind_init


【说明】

音频解码通道绑定初始化。

QUA_S32 (*bind_init)(QUA_U32 mod_id)			
参数名称	输入/输出	描述	返回值
mod_id	输入	模块 ID	成功: QUA_SUCCESS; 失败: 其它值

bind_exit

【说明】

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

音频解码通道绑定去初始化。

QUA_S32 (*bind_exit) (QUA_U32 mod_id)			
参数名称	输入/输出	描述	返回值
mod_id	输入	模块 ID	成功: QUA_SUCCESS; 失败: 其它值

send_stream

【说明】

发送音频解码前数据到解码器。

QUA_S32 (*send_stream)(QUA_ADEC_CHN chn_id, qua_audio_stream_t *stream, QUA_S32 timeout)			
参数名称	输入/输出	描述	返回值
chn_id	输入	音频解码通道号	成功: QUA_SUCCESS; 失败: 其它值
stream	输入	码流结构指针	
timeout	输入	超时时间, -1 阻塞/0 立即返回	

clear_chn_buf

【说明】

清除音频解码通道缓存数据。

QUA_S32 (*clear_chn_buf)(QUA_ADEC_CHN chn_id)			
参数名称	输入/输出	描述	返回值
chn_id	输入	音频解码通道号	成功: QUA_SUCCESS; 失败: 其它值

get_frame

【说明】

获取音频解码后数据。

QUA_S32 (*get_frame)(QUA_ADEC_CHN chn_id, qua_audio_frame_t *frame, QUA_S32 timeout)			
参数名称	输入/输出	描述	返回值
chn_id	输入	音频解码通道号	成功: QUA_SUCCESS; 失败: 其它值
frame	输入	获取的音频帧信息	
timeout	输入	超时时间, -1 阻塞/0 立即返回	

release_frame

【说明】

释放音频解码通道数据。



QUA_S32 (*release_frame)(QUA_ADEC_CHN chn_id, qua_audio_frame_t *frame)

参数名称	输入/输出	描述	返回值
chn_id	输入	音频解码通道号	成功: QUA_SUCCESS;
frame	输入	音频帧信息	失败: 其它值

ext_api

【说明】

音频解码模块扩展 API。

QUA_S32 (*ext_api)(QUA_ADEC_CHN chn_id, QUA_VOID_PTR attr);

参数名称	输入/输出	描述	返回值
chn_id	输入	音频解码通道号	成功: QUA_SUCCESS;
attr	输入	参数指针	失败: 其它值

【注意】

仅 qm10xh 芯片平台支持音频编解码功能

3.4.4 音频编码函数接口

bind_init

【说明】

音频编码通道绑定初始化。

QUA_S32 (*bind_init)(QUA_U32 mod_id)

参数名称	输入/输出	描述	返回值
mod_id	输入	模块 ID	成功: QUA_SUCCESS; 失败: 其它值

bind_exit

【说明】

音频编码通道绑定去初始化。

QUA_S32 (*bind_exit)(QUA_U32 mod_id)

参数名称	输入/输出	描述	返回值
mod_id	输入	模块 ID	成功: QUA_SUCCESS; 失败: 其它值

get_stream

【说明】

获取编码通道编码后码流数据。

QUA_S32 (*get_stream)(QUA_AENC_CHN chn_id, qua_audio_stream_t *stream, QUA_S32



参数名称	输入/输出	描述	返回值
chn_id	输入	视频编码通道号	成功: QUA_SUCCESS; 失败: 其它值
stream	输入	码流结构指针	
timeout	输入	超时时间, -1 阻塞/0 立即返回	

clear_chn_buf

【说明】

清除音频编码通道缓存数据。

QUA_S32 (*clear_chn_buf)(QUA_AENC_CHN chn_id)

参数名称	输入/输出	描述	返回值
chn_id	输入	音频编码通道号	成功: QUA_SUCCESS; 失败: 其它值

send_frame

【说明】

发送音频编码前数据给编码器。

QUA_S32 (*send_frame)(QUA_AENC_CHN chn_id, qua_audio_frame_t *frame, QUA_S32 timeout)

参数名称	输入/输出	描述	返回值
chn_id	输入	音频编码通道号	成功: QUA_SUCCESS; 失败: 其它值
frame	输入	发送的音频帧信息	
timeout	输入	超时时间, -1 阻塞/0 立即返回	

release_frame

【说明】

释放音频编码通道数据。

QUA_S32 (*release_frame)(QUA_AENC_CHN chn_id, qua_audio_frame_t *frame)

参数名称	输入/输出	描述	返回值
chn_id	输入	音频编码通道号	成功: QUA_SUCCESS; 失败: 其它值
frame	输入	音频帧信息	

ext_api

【说明】

音频编码模块扩展 API。

QUA_S32 (*ext_api)(QUA_AENC_CHN chn_id, QUA_VOID_PTR attr)

参数名称	输入/输出	描述	返回值
------	-------	----	-----



QuaMM 多媒体接口使用说明

生效日期
Effective Date

2024.08.01

版本
Rev.


1.1.0

chn_id	输入	音频编码通道号	成功: QUA_SUCCESS; 失败: 其它值
attr	输入	参数指针	

【注意】

仅”qm10xh”芯片平台支持音频编解码功能

Quaming Confidential

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

3.5 显示模块接口

显示模块主要提供视频层、图形层和 VO 设备层等显示接口。所在的头文件路径：`include/display`。

3.5.1 数据结构

qua_vo_pub_attr_t

【说明】

定义设备层公共属性。

【定义】

```
typedef struct qua_vo_pub_attr {
    QUA_U32                bg_color;
    QUA_VO_INTF_TYPE      intf_type;
    qua_vo_intf_sync_t    intf_sync;
    qua_vo_sync_info_t    sync_info;
} qua_vo_pub_attr_t;
```

【描述】

成员名称	描述
bg_color	设备背景色，表示方法 RGB888。
intf_type	接口类型典型配置，原型定义： <pre>typedef QUA_S32 QUA_VO_INTF_TYPE; #define QUA_VO_INTF_CVBS (0x01L<<0) #define QUA_VO_INTF_YPBPR (0x01L<<1) #define QUA_VO_INTF_VGA (0x01L<<2) #define QUA_VO_INTF_BT656 (0x01L<<3) #define QUA_VO_INTF_BT1120 (0x01L<<4) #define QUA_VO_INTF_HDMI (0x01L<<5) #define QUA_VO_INTF_LCD (0x01L<<6) #define QUA_VO_INTF_BT656_H (0x01L<<7) #define QUA_VO_INTF_BT656_L (0x01L<<8) #define QUA_VO_INTF_AHD (0x01L<<9)</pre>
intf_sync	接口时序典型配置，原型定义： <pre>typedef enum qua_vo_intf_sync { QUA_VO_OUTPUT_PAL = 0, QUA_VO_OUTPUT_NTSC, QUA_VO_OUTPUT_960H_PAL, QUA_VO_OUTPUT_960H_NTSC, QUA_VO_OUTPUT_1080P24, QUA_VO_OUTPUT_1080P25, QUA_VO_OUTPUT_1080P30,</pre>



QUA_VO_OUTPUT_720P50,
QUA_VO_OUTPUT_720P60,
QUA_VO_OUTPUT_1080I50,
QUA_VO_OUTPUT_1080I60,
QUA_VO_OUTPUT_1080P50,
QUA_VO_OUTPUT_1080P60,

QUA_VO_OUTPUT_576P50,
QUA_VO_OUTPUT_480P60,

QUA_VO_OUTPUT_640x480_60,
QUA_VO_OUTPUT_800x480_60,
QUA_VO_OUTPUT_800x600_60,
QUA_VO_OUTPUT_1024x768_60,
QUA_VO_OUTPUT_1024x600_60,
QUA_VO_OUTPUT_1280x1024_60,
QUA_VO_OUTPUT_1366x768_60,
QUA_VO_OUTPUT_1440x900_60,
QUA_VO_OUTPUT_1280x800_60,
QUA_VO_OUTPUT_1680x1050_60,
QUA_VO_OUTPUT_1920x2160_30,
QUA_VO_OUTPUT_1600x1200_60,
QUA_VO_OUTPUT_1920x1200_60,
QUA_VO_OUTPUT_2560x1440_30,
QUA_VO_OUTPUT_2560x1440_60,
QUA_VO_OUTPUT_2560x1600_60,
QUA_VO_OUTPUT_3840x2160_25,
QUA_VO_OUTPUT_3840x2160_30,
QUA_VO_OUTPUT_3840x2160_50,
QUA_VO_OUTPUT_3840x2160_60,
QUA_VO_OUTPUT_4096x2160_30,
QUA_VO_OUTPUT_7680x4320_30_FRL_8G,
QUA_VO_OUTPUT_USER,

QUA_VO_OUTPUT_800x1280_60,
QUA_VO_OUTPUT_240x320_60,
QUA_VO_OUTPUT_480x854_60,
QUA_VO_OUTPUT_1200x576_50,
QUA_VO_OUTPUT_3840x2160_15,
QUA_VO_OUTPUT_1280x960_60,
QUA_VO_OUTPUT_720x540_60,
QUA_VO_OUTPUT_960x1280_60,
QUA_VO_OUTPUT_720x575_50,
QUA_VO_OUTPUT_720x1280_60,
QUA_VO_OUTPUT_720P25,

	<p>QUA_VO_OUTPUT_720P30, QUA_VO_OUTPUT_1080x1920_60,</p> <p>QUA_VO_OUTPUT_BUTT }qua_vo_intf_sync_t;</p>
sync_info	<p>接口时序结构体，原型定义： typedef struct qua_vo_sync_info { QUA_BOOL synm; QUA_BOOL iop; QUA_U8 intfb; QUA_U16 vact; QUA_U16 vbb; QUA_U16 vfb; QUA_U16 hact; QUA_U16 hbb; QUA_U16 hfb; QUA_U16 hmid; QUA_U16 bvact; QUA_U16 bvbb; QUA_U16 bvfb; QUA_U16 hpw; QUA_U16 vpw; QUA_BOOL idv; QUA_BOOL ihs; QUA_BOOL ivs; QUA_U32 clk; } qua_vo_sync_info_t;</p>


qua_vo_video_layer_attr_t

【说明】

定义视频层属性。

【定义】

```
typedef struct qua_vo_video_layer_attr {
    qua_rect_t    disp_rect;
    qua_size_t    image_size;
    QUA_U32      disp_fmrt;
    qua_pixel_fmt_t pix_format;
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```

QUA_BOOL      double_frame;
QUA_BOOL      cluster_mode;

```

```

} qua_vo_video_layer_attr_t;

```

【描述】

成员名称	描述
disp_rect	视频显示区域矩形结构体
image_size	图像分辨率结构体，即合成画面尺寸
disp_frmrt	视频显示帧率
pix_format	视频层输入像素格式
double_frame	视频层倍帧开关
cluster_mode	视频层内存聚集使能开关

qua_vo_compress_attr_t

【说明】

定义视频层属性。

【定义】

```

typedef struct qua_vo_compress_attr {
    QUA_BOOL support_compress;
} qua_vo_compress_attr_t;

```

【描述】

成员名称	描述
support_compress	视频层压缩开关

qua_vo_hdmi_fmt_t

【说明】

定义 HDMI 格式。

【定义】

```

typedef struct qua_vo_hdmi_fmt {
    qua_hdmi_format_t hdmi_fmt;
} qua_vo_hdmi_fmt_t;

```

【描述】

成员名称	描述
hdmi_fmt	HDMI 格式，原型定义： <pre> typedef enum qua_hdmi_format { QUA_HDMI_FORMAT_RGB888 = 0, QUA_HDMI_FORMAT_YUV444, QUA_HDMI_FORMAT_EXT_YUV422_8BIT, QUA_HDMI_FORMAT_EXT_YUV422_16BIT, QUA_HDMI_FORMAT_EXT_YUV444_24BIT, QUA_HDMI_FORMAT_BT_1120, QUA_HDMI_FORMAT_NUM, </pre>



} qua_hdmi_format_t;

qua_vo_chn_attr_t

【说明】

定义视频层通道属性。

【定义】

```
typedef struct qua_vo_chn_attr {
    QUA_U32      priority;
    qua_rect_t   rect;
    QUA_BOOL     deflicker;
} qua_vo_chn_attr_t;
```

【描述】

成员名称	描述
priority	视频通道优先级，数值越大优先级越高，优先级高的叠加在上层
rect	视频通道显示区域矩形结构体
deflicker	抗闪烁开关

qua_vo_chn_param_t

【说明】

定义视频层通道参数。

【定义】

```
typedef struct qua_vo_chn_param {
    qua_aspect_ratio_info_t aspect_ratio;
} qua_vo_chn_param_t;
```

【描述】

成员名称	描述
aspect_ratio	幅型比参数。对于没有视频显示的区域，通过设置相应颜色填充

qua_fb_colorkey_t

【说明】

定义图形层透明色。

【定义】

```
typedef struct qua_fb_colorkey {
    QUA_BOOL     key_enable;
    QUA_U32     key_value;
} qua_fb_colorkey_t;
```

【描述】



成员名称	描述
key_enable	是否使能透明色
key_value	透明色的值

qua_fb_alpha_t

【说明】

定义图形层 Alpha 参数。

【定义】

```
typedef struct qua_fb_alpha {
    QUA_BOOL    alpha_enable;
    QUA_BOOL    alpha_channel;
    QUA_U8      alpha0;
    QUA_U8      alpha1;
    QUA_U8      global_alpha;
    QUA_U8      reserved;
} qua_fb_alpha_t;
```

【描述】

成员名称	描述
alpha_enable	Alpha 叠加使能，默认为 1
alpha_channel	Alpha 通道使能，默认为 0
alpha0	Alpha0 值，范围 0~255，默认为 255。在 RGB1:5:5:5 格式下，当最高位为 0 时，选择该值作为 Alpha 叠加的 Alpha 值
alpha1	Alpha1 值，范围 0~255，默认为 255。在 RGB1:5:5:5 格式下，当最高位为 1 时，选择该值作为 Alpha 叠加的 Alpha 值
global_alpha	全局 Alpha 值，范围为 0~128，默认为 128。
reserved	保留项


qua_fb_layer_info_t

【说明】

定义图形层信息。

【定义】

```
typedef struct qua_fb_layer_info {
    qua_fb_buffer_mode_t buf_mode;
    qua_fb_antiflicker_level_t antiflicker_level;
    QUA_S32 origion_x;
    QUA_S32 origion_y;
    QUA_S32 canvas_hidth;
    QUA_S32 canvas_height;
    QUA_U32 display_width;
```

	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

```

QUA_U32 display_height;
QUA_U32 screen_width;
QUA_U32 screen_height;
QUA_BOOL pre_mul;
QUA_U32 mask;
QUA_U32 buf_len;
QUA_U32 buf_phys;
} qua_fb_layer_info_t;

```

【描述】

成员名称	描述
buf_mode	刷新模式，原型定义： <pre> typedef enum qua_fb_buffer_mode { FB_BUF_MODE_DOUBLE = 0, FB_BUF_MODE_ONE, FB_BUF_MODE_NONE, FB_BUF_MODE_DOUBLE_IMMEDIATE, FB_BUF_MODE_BUTT } qua_fb_buffer_mode_t; </pre>
antiflicker_level	图形层抗闪烁等级
origion_x	图形层在屏幕上的原点横坐标
origion_y	图形层在屏幕上的原点纵坐标
canvas_width	画布的宽
canvas_height	画布的高
display_width	显示分辨率的宽
display_height	显示分辨率的高
screen_width	屏幕分辨率的宽
screen_height	屏幕分辨率的高
pre_mul	是否为预乘数据
mask	修改掩码位
buf_len	Buffer 长度
buf_phys	Buffer 物理地址

qua_mm_display_module_t

【说明】

QuaMM 显示模块表示。

【定义】

```

typedef struct qua_mm_display_module {
    struct qua_mm_module_t parent;
} qua_mm_display_module_t;

```

【描述】

成员名称	描述
parent	父模块信息，包含打开设备的函数指针。



要打开 VO 设备层，open_device 函数的 dev_id 参数传 QUA_MM_DISP_DEV_VO，dev_num 参数传设备号；
要打开图形层，open_device 函数的 dev_id 参数传 QUA_MM_DISP_DEV_FB，dev_num 参数传如下枚举值：

```
typedef enum qua_fb_dev {
    QUA_FB_DEV0 = 0,
    QUA_FB_DEV1,
    QUA_FB_DEV2,
    QUA_FB_DEV3,
    QUA_FB_DEV4,
    QUA_FB_DEV5,
    QUA_FB_DEV6,
    QUA_FB_DEV7,
    QUA_FBDEV_BUTT
} qua_fb_dev_t;
```

【注意】

- 对于"mc6870"芯片平台，QUA_FB_DEV0、QUA_FB_DEV2、QUA_FB_DEV4 分别为 HD0 的 OSD0、HC 和 OSD1；QUA_FB_DEV1、QUA_FB_DEV3 分别为 SD 的 OSD 和 HC；QUA_FB_DEV5、QUA_FB_DEV6、QUA_FB_DEV7 分别为 HD1 的 OSD0，HC，OSD1。
- 对于"qm10xh"芯片平台，QUA_FB_DEV0、QUA_FB_DEV2 分别为 HD 的 OSD 和 HC。
- 对于"mc331x"芯片平台，如果是"linux"操作系统，则 QUA_FB_DEV0、QUA_FB_DEV2 分别为 HD 的 OSD 和 HC，QUA_FB_DEV1、QUA_FB_DEV3 分别为 SD 的 OSD 和 HC。如果是"rtt"操作系统，则需通过 bind 接口绑定设备层和图形层。
- HD、SD、HC 分别指高清设备、标清设备、硬件鼠标。

qua_mm_vo_device_t**【说明】**


定义 VO 设备层所涉及的函数接口。

【定义】

```
typedef struct qua_mm_vo_device {
    struct qua_mm_device_t parent;

    QUA_VO_DEV dev_num;

    QUA_S32 (*set_public_attr)(QUA_VO_DEV dev_num, const qua_vo_pub_attr_t *attr);
    QUA_S32 (*get_public_attr)(QUA_VO_DEV dev_num, qua_vo_pub_attr_t *attr);
    QUA_S32 (*enable)(QUA_VO_DEV dev_num);
    QUA_S32 (*disable)(QUA_VO_DEV dev_num);
    QUA_S32 (*set_video_layer_attr)(QUA_VO_LAYER layer, const qua_vo_video_layer_attr_t *layer_attr,
        const qua_vo_compress_attr_t *compress_attr);
    QUA_S32 (*get_video_layer_attr)(QUA_VO_LAYER layer, qua_vo_video_layer_attr_t *layer_attr,
        qua_vo_compress_attr_t *compress_attr);
    QUA_S32 (*enable_video_layer)(QUA_VO_LAYER layer);
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```

QUA_S32 (*disable_video_layer)(QUA_VO_LAYER layer);
QUA_S32 (*bind_video_layer)(QUA_VO_LAYER layer, QUA_VO_DEV dev_num);
QUA_S32 (*unbind_video_layer)(QUA_VO_LAYER layer, QUA_VO_DEV dev_num);
QUA_S32 (*set_video_layer_priority)(QUA_VO_LAYER layer, QUA_U32 priority);
QUA_S32 (*set_video_disp_buflen)(QUA_VO_LAYER layer, QUA_U32 buflen);
QUA_S32 (*set_hdmi_fmt)(QUA_VO_DEV dev_num, const qua_vo_hdmi_fmt_t *hdmi_fmt);
} qua_mm_vo_device_t;

```

【描述】

成员名称	描述
parent	父设备信息，包含关闭设备和创建通道的函数指针
dev_num	设备号
set_public_attr	设置设备层公共属性
get_public_attr	获取设备层公共属性
enable	使能设备层
disable	去使能设备层
set_video_layer_attr	设置视频层属性
get_video_layer_attr	获取视频层属性
enable_video_layer	使能视频层
disable_video_layer	去使能视频层
bind_video_layer	绑定视频层
unbind_video_layer	解绑视频层
set_video_layer_priority	设置视频层优先级
set_video_disp_buflen	设置视频层缓存个数
set_hdmi_fmt	设置 HDMI 格式

以上表格里的函数指针成员放在 3.5.2 节“VO 设备函数接口”里做具体说明。

qua_mm_vo_channel_t

【说明】

定义视频层通道所涉及的函数接口。

【定义】


```

typedef struct qua_mm_vo_channel {
    struct qua_mm_channel_t parent;

    QUA_VO_CHN chn_id;

    QUA_S32 (*set_chn_attr)(QUA_VO_LAYER layer, QUA_VO_CHN chn, const qua_vo_chn_attr_t
*chn_attr);
    QUA_S32 (*get_chn_attr)(QUA_VO_LAYER layer, QUA_VO_CHN chn, qua_vo_chn_attr_t *chn_attr);
    QUA_S32 (*enable_chn)(QUA_VO_LAYER layer, QUA_VO_CHN chn);
    QUA_S32 (*disable_chn)(QUA_VO_LAYER layer, QUA_VO_CHN chn);
    QUA_S32 (*set_chn_param)(QUA_VO_LAYER layer, QUA_VO_CHN chn, const qua_vo_chn_param_t
*param);
    QUA_S32 (*get_chn_param)(QUA_VO_LAYER layer, QUA_VO_CHN chn, qua_vo_chn_param_t

```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

*param);

QUA_S32 (*set_chn_disp_pos)(QUA_VO_LAYER layer, QUA_VO_CHN chn, const qua_point_t *pos);

QUA_S32 (*get_chn_disp_pos)(QUA_VO_LAYER layer, QUA_VO_CHN chn, qua_point_t *pos);

QUA_S32 (*set_chn_frame_rate)(QUA_VO_LAYER layer, QUA_VO_CHN chn, QUA_S32 frame_rate);

QUA_S32 (*get_chn_frame_rate)(QUA_VO_LAYER layer, QUA_VO_CHN chn, QUA_S32

*frame_rate);

QUA_S32 (*get_chn_frame)(QUA_VO_LAYER layer, QUA_VO_CHN chn, qua_video_frame_info_t

*frame, QUA_S32 m_sec);

QUA_S32 (*release_chn_frame)(QUA_VO_LAYER layer, QUA_VO_CHN chn, const qua_video_frame_info_t *frame);

QUA_S32 (*show_chn)(QUA_VO_LAYER layer, QUA_VO_CHN chn);

QUA_S32 (*hide_chn)(QUA_VO_LAYER layer, QUA_VO_CHN chn);

QUA_S32 (*get_chn_pts)(QUA_VO_LAYER layer, QUA_VO_CHN chn, QUA_U64 *chn_pts);

QUA_S32 (*send_frame)(QUA_VO_LAYER layer, QUA_VO_CHN chn, qua_video_frame_info_t

*frame, QUA_S32 m_sec);

QUA_S32 (*clear_chn_buffer)(QUA_VO_LAYER layer, QUA_VO_CHN chn, QUA_BOOL clear_all);


QUA_S32 (*fill_chn)(QUA_VO_LAYER layer, QUA_VO_CHN chn, QUA_U32 rgb);

} qua_mm_vo_channel_t;

【描述】

成员名称	描述
parent	父通道信息，包含销毁通道的函数指针
chn_id	通道号
set_chn_attr	设置视频通道属性
get_chn_attr	获取视频通道属性
enable_chn	使能视频通道
disable_chn	去使能视频通道
set_chn_param	设置视频通道参数
get_chn_param	获取视频通道参数
set_chn_disp_pos	设置视频通道显示位置
get_chn_disp_pos	获取视频通道显示位置
set_chn_frame_rate	设置视频通道帧率
get_chn_frame_rate	获取视频通道帧率
get_chn_frame	获取视频通道帧数据
release_chn_frame	释放视频通道帧数据
show_chn	显示视频通道
hide_chn	隐藏视频通道
get_chn_pts	获取视频通道时间戳
send_frame	发送视频通道帧数据
clear_chn_buffer	清除视频通道缓存
fill_chn	填充视频通道

以上表格里的函数指针成员放在 3.5.3 节“VO 通道函数接口”里做具体说明。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_mm_fb_device_t

【说明】

定义图形层所涉及的函数接口。

【定义】

```
typedef struct qua_mm_fb_device {
    struct qua_mm_device_t parent;

    qua_fb_dev_t dev_num;
    QUA_S32 fd;

    QUA_S32 (*show)(struct qua_mm_fb_device *dev, QUA_BOOL is_show);
    QUA_S32 (*compress)(struct qua_mm_fb_device *dev, QUA_BOOL is_compress);
    QUA_S32 (*get_origin)(struct qua_mm_fb_device *dev, qua_point_t *pos);
    QUA_S32 (*put_origin)(struct qua_mm_fb_device *dev, const qua_point_t *pos);
    QUA_S32 (*get_size)(struct qua_mm_fb_device *dev, qua_size_t *screen_size);
    QUA_S32 (*put_size)(struct qua_mm_fb_device *dev, const qua_size_t *screen_size);
    QUA_S32 (*get_colorkey)(struct qua_mm_fb_device *dev, qua_fb_colorkey_t *colorkey);
    QUA_S32 (*put_colorkey)(struct qua_mm_fb_device *dev, const qua_fb_colorkey_t *colorkey);
    QUA_S32 (*get_alpha)(struct qua_mm_fb_device *dev, qua_fb_alpha_t *alpha);
    QUA_S32 (*put_alpha)(struct qua_mm_fb_device *dev, const qua_fb_alpha_t *alpha);
    QUA_S32 (*get_var_info)(struct qua_mm_fb_device *dev, qua_fb_var_screeninfo_t *var_info);
    QUA_S32 (*put_var_info)(struct qua_mm_fb_device *dev, const qua_fb_var_screeninfo_t *var_info);
    QUA_S32 (*get_fix_info)(struct qua_mm_fb_device *dev, qua_fb_fix_screeninfo_t *fix_info);
    QUA_VOID_PTR (*map)(struct qua_mm_fb_device *dev, QUA_SIZE_T size, QUA_OFF_T offset);
    QUA_VOID (*unmap)(struct qua_mm_fb_device *dev, QUA_VOID_PTR vaddr, QUA_SIZE_T size);
    QUA_S32 (*render)(struct qua_mm_fb_device *dev, const qua_fb_var_screeninfo_t *var_info);
    QUA_S32 (*bind)(struct qua_mm_fb_device *dev, QUA_VO_DEV dev_num, QUA_VO_LAYER layer);
    QUA_S32 (*unbind)(struct qua_mm_fb_device *dev, QUA_VO_DEV dev_num, QUA_VO_LAYER layer);

    QUA_S32 (*get_layer_info)(struct qua_mm_fb_device *dev, qua_fb_layer_info_t *layer_info);
    QUA_S32 (*put_layer_info)(struct qua_mm_fb_device *dev, const qua_fb_layer_info_t *layer_info);
} qua_mm_fb_device_t;
```

【描述】

成员名称	描述
parent	父设备信息，包含关闭设备的函数指针
dev_num	设备号
fd	打开的文件描述符
show	显示或隐藏图形层
compress	图形层数据压缩开关
get_origin	获取图形层原点
put_origin	设置图形层原点
get_size	获取图形层大小



put_size	设置图形层大小
get_colorkey	获取图形层透明色
put_colorkey	设置图形层透明色
get_alpha	获取图形层 Alpha 值
put_alpha	设置图形层 Alpha 值
get_var_info	获取图形层可变信息
put_var_info	设置图形层可变信息
get_fix_info	获取图形层固定信息
map	映射图形层缓存
unmap	去映射图形层缓存
render	图形层缓存送显
bind	绑定图形层
unbind	解绑图形层
get_layer_info	获取图形层信息
put_layer_info	设置图形层信息

以上表格里的函数指针成员放在 3.5.4 节“FB 设备函数接口”里做具体说明。

3.5.2 VO 设备函数接口

set_public_attr

【说明】

设置设备层公共属性。

QUA_S32 (*set_public_attr)(QUA_VO_DEV dev_num, const qua_vo_pub_attr_t *attr)			
参数名称	输入/输出	描述	返回值
dev_num	输入	设备号	成功: QUA_SUCCESS; 失败: 其它值
attr	输入	设备层公共属性	

【注意】

必须在使能设备层前配置

enable


【说明】

使能设备层。

QUA_S32 (*enable)(QUA_VO_DEV dev_num)			
参数名称	输入/输出	描述	返回值
dev_num	输入	设备号	成功: QUA_SUCCESS; 失败: 其它值

disable

【说明】

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

去使能设备层。

QUA_S32 (*disable)(QUA_VO_DEV dev_num)			
参数名称	输入/输出	描述	返回值
dev_num	输入	设备号	成功: QUA_SUCCESS; 失败: 其它值

【注意】

- 设备禁止前必须先禁止该设备上的视频层。
- 调用设备使能接口后，如果未调用该接口进行禁止，则 VO 设备将一直保持使能状态，并且下次设置设备属性时不会生效。
- 设备禁止后需要重新设置设备公共属性，才可使能设备。

set_video_layer_attr

【说明】

设置视频层属性。

QUA_S32 (*set_video_layer_attr)(QUA_VO_LAYER layer, const qua_vo_video_layer_attr_t *layer_attr, const qua_vo_compress_attr_t *compress_attr)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
layer_attr	输入	视频层属性	
compress_attr	输入	视频层数据是否压缩	

【注意】

设置视频层属性必须在视频层禁止的情况下进行。

get_video_layer_attr

【说明】

获取视频层属性。

QUA_S32 (*get_video_layer_attr)(QUA_VO_LAYER layer, qua_vo_video_layer_attr_t *layer_attr, qua_vo_compress_attr_t *compress_attr)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
layer_attr	输出	视频层属性	
compress_attr	输出	视频层数据是否压缩	

enable_video_layer

【说明】

使能视频层。

QUA_S32 (*enable_video_layer)(QUA_VO_LAYER layer)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS;



失败：其它值

【注意】

- 视频层使能前必须保证该视频层所绑定的设备处于使能状态。
- 视频层使能前必须保证该视频层已经配置。

disable_video_layer

【说明】

去使能视频层。

QUA_S32 (*disable_video_layer)(QUA_VO_LAYER layer)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值

【注意】

- 视频层禁止前必须保证其上的通道全部禁止。
- 如果 VO 作为输出源绑定到其它模块，则在退出业务的时候需要先关闭所绑定的模块，再关闭 VO，以防止绑定模块仍使用 VO 已释放的 VB 资源而引起错误。

bind_video_layer

【说明】

绑定视频层到指定设备。

QUA_S32 (*bind_video_layer)(QUA_VO_LAYER layer, QUA_VO_DEV dev_num)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
dev_num	输入	设备号	

【注意】

- "qm10xh"芯片平台不支持该接口。
- 不能在视频层使能的情况下，调用该接口设置绑定关系。

unbind_video_layer

【说明】

将指定视频层从设备上面解除绑定。

QUA_S32 (*unbind_video_layer)(QUA_VO_LAYER layer, QUA_VO_DEV dev_num)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
dev_num	输入	设备号	

【注意】

- "qm10xh"芯片平台不支持该接口。
- 解除绑定前必须关闭视频层。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

set_video_layer_priority

【说明】

设置视频层的优先级。

QUA_S32 (*set_video_layer_priority)(QUA_VO_LAYER layer, QUA_U32 priority)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
priority	输入	视频层优先级	

【注意】

- “qm10xh”芯片平台不支持该接口。
- 本接口被调用时，视频层必须已经绑定。
- 每个视频层绑定到设备时，都会被默认给定一个优先级。默认的优先级是按照所有优先级从大到小索引的原则设置的，当找到一个最大的优先级，则会对这个视频层进行默认设置。
- 需要关闭所有绑定到该设备上的所有视频层才能修改优先级。

set_video_disp_buflen

【说明】

设置视频层缓存个数。

QUA_S32 (*set_video_disp_buflen)(QUA_VO_LAYER layer, QUA_U32 buflen)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
buflen	输入	缓存个数	

【注意】

调用前需保证视频层未使能。

set_hdmi_fmt

【说明】


设置 HDMI 格式。

QUA_S32 (*set_hdmi_fmt)(QUA_VO_DEV dev_num, const qua_vo_hdmi_fmt_t *hdmi_fmt)			
参数名称	输入/输出	描述	返回值
dev_num	输入	设备号	成功: QUA_SUCCESS; 失败: 其它值
hdmi_fmt	输入	HDMI 格式	

【注意】

仅“qm10xh”芯片平台支持该接口。

3.5.3 VO 通道函数接口

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

set_chn_attr

【说明】

配置指定视频输出通道的属性。

QUA_S32 (*set_chn_attr)(QUA_VO_LAYER layer, QUA_VO_CHN chn, const qua_vo_chn_attr_t *chn_attr)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	
chn_attr	输入	通道属性	

【注意】

- 属性中的优先级，数值越大优先级越高。
- 当多个通道有重叠的显示区域时，优先级高的通道图像将覆盖优先级低的通道。优先级相同的各通道有重叠时，默认通道号大的图像将覆盖通道号小的通道图像。
- 通道显示区域不能超过视频层属性中设定的画布大小。

enable_chn

【说明】

启用指定的视频输出通道。

QUA_S32 (*enable_chn)(QUA_VO_LAYER layer, QUA_VO_CHN chn)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	

【注意】

- 调用前必须使能相应设备上的视频层。
- 通道使能前必须进行通道配置，否则返回通道未配置的错误。

disable_chn

【说明】

禁用指定的视频输出通道。

QUA_S32 (*disable_chn)(QUA_VO_LAYER layer, QUA_VO_CHN chn)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	

set_chn_param

【说明】

配置指定视频输出通道的参数。



QUA_S32 (*set_chn_param)(QUA_VO_LAYER layer, QUA_VO_CHN chn, const qua_chn_param_t *param)

参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	
param	输入	通道参数	

set_chn_disp_pos

【说明】

设置指定视频输出通道的显示位置。

QUA_S32 (*set_chn_disp_pos)(QUA_VO_LAYER layer, QUA_VO_CHN chn, const qua_point_t *pos)

参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	
pos	输入	通道显示位置	

【注意】

- 默认情况下，VO 设备从视频层的左上角开始显示视频层画布上的通道，设置通道属性时设定了通道在画布上的位置，通道在画布上的位置即为默认的显示位置。当调用该接口设置通道的显示位置时，新的显示位置将取代默认值，成为 VO 设备上通道的实际显示位置。
- 该接口设置的显示位置 (x, y) 在通道 disable 之后仍然有效，用户再使能通道，该通道的显示位置仍然为 (x, y)；若用户需要恢复默认的显示位置，需要手动的去设置显示位置为 (-1, -1)。

set_chn_frame_rate

【说明】

设置指定视频输出通道的显示帧率。

QUA_S32 (*set_chn_frame_rate)(QUA_VO_LAYER layer, QUA_VO_CHN chn, QUA_S32 frame_rate)

参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	
frame_rate	输入	显示帧率	

【注意】

需先设置 VO 相应视频输出通道属性，才能调用此接口。

get_chn_frame

【说明】

获取通道帧。

QUA_S32 (*get_chn_frame)(QUA_VO_LAYER layer, QUA_VO_CHN chn,



qua_video_frame_info_t *frame, QUA_S32 m_sec)

参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	
frame	输出	帧信息	
m_sec	输入	设为-1 时, 为阻塞接口; 0 时为非阻塞接口; 大于 0 时为超时等待时间。超时时间的单位为毫秒 (ms)	

【注意】

调用该接口前必须保证通道已经使能。

release_chn_frame

【说明】

释放输出通道图像数据。

QUA_S32 (*release_chn_frame)(QUA_VO_LAYER layer, QUA_VO_CHN chn, const qua_video_frame_info_t *frame)

参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	
frame	输入	帧信息	

【注意】

获取操作应保证与释放操作配对。

show_chn

【说明】

显示指定通道。

QUA_S32 (*show_chn)(QUA_VO_LAYER layer, QUA_VO_CHN chn)

参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	

【注意】

- 调用前需保证视频输出通道所在的视频层已经使能。
- 默认情况下通道处于显示状态。

hide_chn

【说明】

隐藏指定通道。

QUA_S32 (*hide_chn)(QUA_VO_LAYER layer, QUA_VO_CHN chn)

参数名称	输入/输出	描述	返回值
------	-------	----	-----



layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	

get_chn_pts

【说明】

获取指定视频输出通道正在显示图像的时间戳。

QUA_S32 (*get_chn_pts)(QUA_VO_LAYER layer, QUA_VO_CHN chn, QUA_U64 *chn_pts)

参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	
chn_pts	输出	时间戳	

【注意】

- “qm10xh”芯片平台不支持该接口。
- 调用前需保证视频层使能，视频通道使能。
- 解码通过系统绑定的方式向 VO 发送视频帧时，如果解码帧的 pts 为 0，VO 会从 0 开始，按照每帧的播放时间间隔打上时间戳，以确保每帧播放一次。其他情况下不会改动视频源的时间戳。

send_frame

【说明】

将视频图像送入指定输出通道显示。

QUA_S32 (*send_frame)(QUA_VO_LAYER layer, QUA_VO_CHN chn, const qua_video_frame_info_t *frame, QUA_S32 m_sec)

参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	
frame	输入	帧信息	
m_sec	输入	设为-1 时，为阻塞接口；0 时为非阻塞接口；大于 0 时为超时等待时间。超时时间的单位为毫秒（ms）	

【注意】

调用该接口前必须保证通道已经使能。

clear_chn_buffer

【说明】

清空指定输出通道的缓存 buffer 数据。

QUA_S32 (*clear_chn_buffer)(QUA_VO_LAYER layer, QUA_VO_CHN chn, QUA_BOOL clear_all)

参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	



clear_all	输入	为 QUA_TRUE 时，将通道 buffer 中的数据全部清空。为 QUA_FALSE 时，在通道 buffer 中保留一幅图像数据，而将其其他数据清空。	
-----------	----	--	--

fill_chn

【说明】

填充指定通道。

QUA_S32 (*fill_chn)(QUA_VO_LAYER layer, QUA_VO_CHN chn, QUA_U32 rgb)			
参数名称	输入/输出	描述	返回值
layer	输入	视频层号	成功: QUA_SUCCESS; 失败: 其它值
chn	输入	通道号	
rgb	输入	要填充的 RGB 颜色	

【注意】

"mc331x" 芯片平台不支持该接口。

3.5.4 FB 设备函数接口

show

【说明】

显示或隐藏该图形层。

QUA_S32 (*show)(struct qua_mm_fb_device *dev, QUA_BOOL is_show)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
is_show	输入	为 QUA_TRUE 时，显示该图形层。 为 QUA_FALSE 时，隐藏该图形层。	

【注意】


显示时应保证图形层的分辨率不超出设备分辨率。

compress

【说明】

设置该图形层启用压缩功能。

QUA_S32 (*compress)(struct qua_mm_fb_device *dev, QUA_BOOL is_compress)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
is_compress	输入	为 QUA_TRUE 时，该图形层启用压缩。为 QUA_FALSE 时，该图形层不压缩。	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

【注意】

压缩功能默认关闭。

put_origin

【说明】

设置图形层在屏幕上显示的起始点坐标。

QUA_S32 (*put_origin)(struct qua_mm_fb_device *dev, const qua_point_t *pos);			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
pos	输入	坐标原点	

put_size

【说明】

设置屏幕输出尺寸。

QUA_S32 (*put_size)(struct qua_mm_fb_device *dev, const qua_size_t *screen_size)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
screen_size	输入	输出尺寸	

put_colorkey

【说明】

设置该图形层的透明色。

QUA_S32 (*put_colorkey)(struct qua_mm_fb_device *dev, const qua_fb_colorkey_t *colorkey)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
colorkey	输入	透明色	

put_alpha

【说明】

设置当前图形层的 Alpha 功能。

QUA_S32 (*put_alpha)(struct qua_mm_fb_device *dev, const qua_fb_alpha_t *alpha)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
alpha	输入	Alpha 信息	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

put_var_info

【说明】

设置图形层的可变信息。

QUA_S32 (*put_var_info)(struct qua_mm_fb_device *dev, const qua_fb_var_screeninfo_t *var_info)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
var_info	输入	可变信息	

map

【说明】

映射图形层的缓存。

QUA_VOID_PTR (*map)(struct qua_mm_fb_device *dev, QUA_SIZE_T size, QUA_OFF_T offset)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: 返回非空的虚拟地址; 失败: QUA_NULL
size	输入	要映射的大小	
offset	输入	要映射的偏移	

unmap

【说明】

解映射图形层的缓存。


QUA_VOID (*unmap)(struct qua_mm_fb_device *dev, QUA_VOID_PTR vaddr, QUA_SIZE_T size)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	无
vaddr	输入	映射出来的虚拟地址	
size	输入	要解映射的大小	

render

【说明】

将图形层的缓存送去显示。

QUA_S32 (*render)(struct qua_mm_fb_device *dev, const qua_fb_var_screeninfo_t *var_info)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
var_info	输入	可变信息	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

get_var_info

【说明】

获取可变信息。

QUA_S32 (*get_var_info)(struct qua_mm_fb_device *dev, qua_fb_var_screeninfo_t *var_info)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
var_info	输出	可变信息	

【注意】

"rtt"操作系统不支持该接口。

get_fix_info

【说明】

获取固定信息。

QUA_S32 (*get_fix_info)(struct qua_mm_fb_device *dev, qua_fb_fix_screeninfo_t *fix_info)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
fix_info	输出	固定信息	

【注意】

"rtt"操作系统不支持该接口。

bind


【说明】

绑定图形层。

QUA_S32 (*bind)(struct qua_mm_fb_device *dev, QUA_VO_DEV dev_num, QUA_VO_LAYER layer)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
dev_num	输入	设备号	
layer	输入	图形层号	

【注意】

- 仅"rtt"操作系统支持该接口。
- dev_num 设为 0, layer 设为 0, 代表 HD 设备下 OSD 层;
- dev_num 设为 1, layer 设为 1, 代表 SD 设备下 OSD 层;
- dev_num 设为 0, layer 设为 2, 代表 HD 设备下 HC 层;
- dev_num 设为 1, layer 设为 3, 代表 SD 设备下 HC 层。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

unbind

【说明】

解绑图形层。

QUA_S32 (*unbind)(struct qua_mm_fb_device *dev, QUA_VO_DEV dev_num, QUA_VO_LAYER layer)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
dev_num	输入	设备号	
layer	输入	图形层号	

【注意】

仅"rtt"操作系统支持该接口。

get_layer_info

【说明】

获取图形层信息。

QUA_S32 (*get_layer_info)(struct qua_mm_fb_device *dev, qua_fb_layer_info_t *layer_info)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
layer_info	输出	图形层信息	

【注意】

仅"rtt"操作系统支持该接口。

put_layer_info


【说明】

设置图形层信息。

QUA_S32 (*put_layer_info)(struct qua_mm_fb_device *dev, const qua_fb_layer_info_t *layer_info)			
参数名称	输入/输出	描述	返回值
dev	输入	图形层	成功: QUA_SUCCESS; 失败: 其它值
layer_info	输入	图形层信息	

【注意】

仅"rtt"操作系统支持该接口。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

3.6 Camera 模块接口

Camera 模块主要提供 vicap 模块初始化、反初始化、属性配置等接口。所在的头文件路径：
include/camera。

3.6.1 数据结构

qua_mm_vi_device_t

【说明】

vicap 设备的通用表示。

【定义】

```
typedef struct qua_mm_vi_device_t {
    struct qua_mm_device_t parent;
    QUA_S32 (*set_peak_load)(qua_vi_peak_load_info_t *peak_load);
    QUA_S32 (*set_phy_lane)(QUA_U32 dev_id, int num);
    QUA_S32 (*init_dev)(QUA_U32 dev_id, qua_vi_attr_t *dev_attr);
    QUA_S32 (*exit_dev)(QUA_U32 dev_id);
    QUA_S32 (*set_attr)(QUA_U32 dev_id, qua_vi_dev_attr_t *vi_attr);
    QUA_S32 (*get_attr)(QUA_U32 dev_id, qua_vi_dev_attr_t *vi_attr);
} qua_mm_vi_device_t;
```

【描述】

成员名称	描述
parent	vicap 模块的模块指针，应用层无需关注
set_peak_load	设置所有 vicap 设备的负载
set_phy_lane	设置 vicap 的 phy lane 数
init_dev	vicap 设备初始化
exit_dev	vicap 设备退出
set_attr	设置 vicap 硬件模块工作属性
get_attr	获取 vicap 硬件模块工作属性

以上表格里的函数指针成员放在下一节“函数接口”里做具体说明。


qua_vi_size_fps_t

【说明】

定义 vicap 分辨率和帧率。

【定义】

```
typedef struct
{
    QUA_U32 width;
    QUA_U32 height;
```

	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

```

QUA_U32 fps;
QUA_U32 pclk;
}qua_vi_size_fps_t;

```

【描述】

成员名称	描述
width	vicap 采集 sensor 帧的宽
height	vicap 采集 sensor 帧的高
fps	vicap 采集 sensor 帧的帧率
pclk	vicap 采集 sensor 帧 pixel 的时钟

qua_vi_peak_load_info_t

【说明】

定义 vicap 负载信息。

【定义】

```

typedef struct
{
    QUA_U32 pclk_set;
    qua_vi_size_fps_t size_fps_info[3];
}qua_vi_peak_load_info_t;

```

【描述】

成员名称	描述
pclk_set	是否配置 pixel clock;0:使用默认值 1: 使用配置的值
size_fps_info	定义负载信息结构体 mipi0 mipi1 dvp

qua_vi_attr_t

【说明】

定义 vicap 初始化属性。

【定义】

```

typedef struct
{
    QUA_BOOL          using_vb;
    qua_vi_input_data_type_t  data_type_in;
    qua_vi_work_mode_t  work_mode;
    qua_size_t         size;
}qua_vi_attr_t;

```

【描述】

成员名称	描述
using_vb	是否使用 VB 缓存逻辑。取值范围: [0~1]
data_type_in	SENSOR 输入该模块的数据格式
work_mode	配置当前在离线工作模式, 影响内存分配
size	图像幅面, 会影响内存分配, 故需按应用需要的最大幅



面配置

qua_vi_dev_attr_t

【说明】

定义 vicap 设备属性。

【定义】

typedef struct

```

{
    qua_vi_bayer_type_t    bayer_type;
    qua_vi_work_mode_t    work_mode;
    QUA_U8                wdr_mode;
    qua_size_t            in_size;
    qua_vi_crop_cfg_t    crop_size;
    qua_vi_offline_cfg_t  offline_cfg;
}qua_vi_dev_attr_t;

```

【描述】

成员名称	描述
bayer_type	bayer pattern。
work_mode	配置当前在离线工作模式
wdr_mode	输入 sensor 的线性还是 wdr 配置。取值范围：[0~1]
in_size	输入幅面
crop_size	裁剪后幅面
offline_cfg	离线下的配置

3.6.2 函数接口

set_peak_load

【说明】

设置 vicap 的负载。

QUA_S32 (*set_peak_load)(qua_vi_peak_load_info_t *peak_load)

参数名称	输入/输出	描述	返回值
peak_load	输入	负载信息	成功: QUA_SUCCESS; 失败: 其它值

set_phy_lane

【说明】

设置 phy 的 lane 数。

QUA_S32 (*set_phy_lane)(QUA_U32 dev_id, QUA_S32 num)



参数名称	输入/输出	描述	返回值
dev_id	输入	vicap 的 dev id 号	成功: QUA_SUCCESS; 失败: 其它值
num	输入	phy 的 lane 数	

init_dev

【说明】

vicap 设备初始化。

QUA_S32 (*init_dev)(QUA_U32 dev_id, qua_vi_attr_t *dev_attr)

参数名称	输入/输出	描述	返回值
dev_id	输入	vicap 的 dev id 号	成功: QUA_SUCCESS; 失败: 其它值
dev_attr	输入	vicap 初始化参数	

exit_dev

【说明】

vicap 设备退出。

QUA_S32 (*exit_dev)(QUA_U32 dev_id)

参数名称	输入/输出	描述	返回值
dev_id	输入	vicap 的 dev id 号	成功: QUA_SUCCESS; 失败: 其它值

set_attr

【说明】

设置 vicap 硬件模块工作属性。

QUA_S32 (*set_attr)(QUA_U32 dev_id, qua_vi_dev_attr_t *vi_attr)

参数名称	输入/输出	描述	返回值
dev_id	输入	vicap 的 dev id 号	成功: QUA_SUCCESS; 失败: 其它值
vi_attr	输入	vicap 硬件模块工作属性	


get_attr

【说明】

获取 vicap 硬件模块工作属性。

QUA_S32 (*get_attr)(QUA_U32 dev_id, qua_vi_dev_attr_t *vi_attr)

参数名称	输入/输出	描述	返回值
dev_id	输入	vicap 的 dev id 号	成功: QUA_SUCCESS; 失败: 其它值
vi_attr	输入	vicap 硬件模块工作属性	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

3.7 Filter 模块接口

Filter 模块主要提供图形图像的处理接口，如格式转换、缩放、旋转、叠加等功能。所在的头文件路径：`include/filter`。

3.7.1 数据结构

qua_vgs_task_attr_t

【说明】

定义 VGS task 的属性。

【定义】

```
typedef struct qua_vgs_task_attr {
    qua_video_frame_info_t img_in;
    qua_video_frame_info_t img_out;
    QUA_U32 private_data[4];
    QUA_U32 reserved;
} qua_vgs_task_attr_t;
```

【描述】

成员名称	描述
img_in	输入图像属性
img_out	输出图像属性
private_data	与 task 相关的私有数据
reserved	保留项

qua_vgs_line_t

【说明】


定义 VGS 画线操作的相关配置。

【定义】

```
typedef struct qua_vgs_line {
    qua_point_t start_point;
    qua_point_t end_point;
    QUA_U32 thick;
    QUA_U32 color;
} qua_vgs_line_t;
```

【描述】

成员名称	描述
start_point	线的起始点坐标
end_point	线的结束点坐标
thick	线的宽度
color	线的颜色

 旷明智能 - QuaMing Intelligence -	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

qua_vgs_rotate_t

【说明】

定义 VGS 旋转属性。

【定义】

```
typedef struct qua_vgs_rotate {
    QUA_BOOL mirror;
    QUA_BOOL flip;
    QUA_BOOL rotate;
} qua_vgs_rotate_t;
```

【描述】

成员名称	描述
mirror	是否水平镜像
flip	是否垂直镜像
rotate	是否顺时针旋转 90 度

qua_vgs_osd_t

【说明】


定义 VGS OSD 属性。

【定义】

```
typedef struct qua_vgs_osd {
    qua_rect_t rect;
    QUA_U32 bg_color;
    qua_pixel_fmt_t pixel_fmt;
    QUA_U32 phy_addr;
    QUA_U32 stride;
    QUA_U32 bg_alpha;
    QUA_U32 fg_alpha;
} qua_vgs_osd_t;
```

【描述】

成员名称	描述
rect	区域信息
bg_color	背景颜色
pixel_fmt	OSD 像素格式
phy_addr	OSD 物理地址
stride	OSD 跨距
bg_alpha	背景透明度
fg_alpha	前景透明度

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_tde2_surface_t

【说明】

定义 TDE 表面。

【定义】

```
typedef struct qua_tde2_surface
{
    QUA_ULONG          phy_addr;
    qua_tde2_color_fmt_t color_fmt;
    QUA_U32           height;
    QUA_U32           width;
    QUA_U32           stride;
    QUA_U8*           clut_phy_addr;
    QUA_BOOL          ycber_clut;
    QUA_BOOL          alpha_max255;
    QUA_BOOL          alpha_ext_1555;
    QUA_U8            alpha0;
    QUA_U8            alpha1;
    QUA_ULONG         cbc_r_phy_addr;
    QUA_U32           cbc_r_stride;
    QUA_BOOL          fbc_compress;
} qua_tde2_surface_t;
```

【描述】

成员名称	描述
phy_addr	表面物理地址
color_fmt	表面颜色格式，为如下枚举值： <pre>typedef enum qua_tde2_color_fmt { QUA_TDE2_COLOR_FMT_RGB444 = 0, QUA_TDE2_COLOR_FMT_BGR444, QUA_TDE2_COLOR_FMT_RGB555, QUA_TDE2_COLOR_FMT_BGR555, QUA_TDE2_COLOR_FMT_RGB565, QUA_TDE2_COLOR_FMT_BGR565, QUA_TDE2_COLOR_FMT_RGB888, QUA_TDE2_COLOR_FMT_BGR888, QUA_TDE2_COLOR_FMT_ARGB4444, QUA_TDE2_COLOR_FMT_ABGR4444, QUA_TDE2_COLOR_FMT_RGBA4444, QUA_TDE2_COLOR_FMT_BGRA4444, QUA_TDE2_COLOR_FMT_ARGB1555, QUA_TDE2_COLOR_FMT_ABGR1555, QUA_TDE2_COLOR_FMT_RGBA1555,</pre>



QUA_TDE2_COLOR_FMT_BGRA1555,
 QUA_TDE2_COLOR_FMT_ARGB8565,
 QUA_TDE2_COLOR_FMT_ABGR8565,
 QUA_TDE2_COLOR_FMT_RGBA8565,
 QUA_TDE2_COLOR_FMT_BGRA8565,
 QUA_TDE2_COLOR_FMT_ARGB8888,
 QUA_TDE2_COLOR_FMT_ABGR8888,
 QUA_TDE2_COLOR_FMT_RGBA8888,
 QUA_TDE2_COLOR_FMT_BGRA8888,
 QUA_TDE2_COLOR_FMT_RABG8888,
 QUA_TDE2_COLOR_FMT_CLUT1,
 QUA_TDE2_COLOR_FMT_CLUT2,
 QUA_TDE2_COLOR_FMT_CLUT4,
 QUA_TDE2_COLOR_FMT_CLUT8,
 QUA_TDE2_COLOR_FMT_ACLUT44,
 QUA_TDE2_COLOR_FMT_ACLUT88,
 QUA_TDE2_COLOR_FMT_A1,
 QUA_TDE2_COLOR_FMT_A8,
 QUA_TDE2_COLOR_FMT_YCbCr888,
 QUA_TDE2_COLOR_FMT_AYCbCr8888,
 QUA_TDE2_COLOR_FMT_YCbCr422,
 QUA_TDE2_COLOR_FMT_byte,
 QUA_TDE2_COLOR_FMT_halfword,
 QUA_TDE2_COLOR_FMT_JPG_YCbCr400MBP,
 QUA_TDE2_COLOR_FMT_JPG_YCbCr422MBHP,
 QUA_TDE2_COLOR_FMT_JPG_YCbCr422MBVP,
 QUA_TDE2_COLOR_FMT_MP1_YCbCr420MBP,
 QUA_TDE2_COLOR_FMT_MP2_YCbCr420MBP,
 QUA_TDE2_COLOR_FMT_MP2_YCbCr420MBI,
 QUA_TDE2_COLOR_FMT_JPG_YCbCr420MBP,
 QUA_TDE2_COLOR_FMT_JPG_YCbCr444MBP,
 QUA_TDE2_COLOR_FMT_YUV420,
 QUA_TDE2_COLOR_FMT_YUV422,
 QUA_TDE2_COLOR_FMT_BUTT
 } qua_tde2_color_fmt_t;

height	表面高度
width	表面宽度
stride	表面跨距
clut_phy_addr	Clut 表物理地址，用作颜色扩展或颜色校正
ycbcr_clut	lut 表是否位于 YCbCr 空间
alpha_max255	表面 alpha 最大值为 255 还是 128
alpha_ext_1555	是否使能 1555 的 Alpha 扩展。 当表面颜色格式为 ARGB1555 时，该项有效
alpha0	Alpha0 值。取值范围：[0, 255]。 当格式为 ARGB1555 且 bAlphaExt1555 为 TRUE 时，



	该项有效。 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]。 当格式为 ARGB1555 且 bAlphaExt1555 为 TRUE 时，该项有效。 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。
cbr_phy_addr	CbCr 分量的物理地址
cbr_stride	CbCr 分量的跨距
fbc_compress	是否压缩数据

qua_tde2_blend_opt_t

【说明】

定义 TDE alpha 混合操作选项。

【定义】

```
typedef struct qua_tde2_blend_opt
```

```
{
    QUA_BOOL        global_alpha_enable;
    QUA_BOOL        pixel_alpha_enable;
    QUA_BOOL        src1_alpha_pre-multi;
    QUA_BOOL        src2_alpha_pre-multi;
    qua_tde2_blendcmd_t  blend_cmd;
    qua_tde2_blend_mode_t  src1_blend_mode;
    qua_tde2_blend_mode_t  src2_blend_mode;
} qua_tde2_blend_opt_t;
```

【描述】

成员名称	描述
global_alpha_enable	是否使能全局 alpha
pixel_alpha_enable	是否使能象素 alpha
src1_alpha_pre-multi	是否使能 Src1 alpha 预乘
src2_alpha_pre-multi	是否使能 Src2 alpha 预乘
blend_cmd	Alpha 混合命令，为如下枚举值： typedef enum qua_tde2_blendcmd { QUA_TDE2_BLEND_CMD_NONE = 0x0, QUA_TDE2_BLEND_CMD_CLEAR, QUA_TDE2_BLEND_CMD_SRC, QUA_TDE2_BLEND_CMD_SRCOVER, QUA_TDE2_BLEND_CMD_DSTOVER, QUA_TDE2_BLEND_CMD_SRCIN, QUA_TDE2_BLEND_CMD_DSTIN, QUA_TDE2_BLEND_CMD_SRCOUT,



	<pre> QUA_TDE2_BLEND_CMD_DSTOUT, QUA_TDE2_BLEND_CMD_SRCATOP, QUA_TDE2_BLEND_CMD_DSTATOP, QUA_TDE2_BLEND_CMD_ADD, QUA_TDE2_BLEND_CMD_XOR, QUA_TDE2_BLEND_CMD_DST, QUA_TDE2_BLEND_CMD_CONFIG, QUA_TDE2_BLEND_CMD_BUTT } qua_tde2_blendcmd_t; </pre>
src1_blend_mode	<p>Src1 blend 模式选择,在 eBlendCmd = TDE2_BLEND_CMD_CONFIG 时有效,为如下枚举值:</p> <pre> typedef enum qua_tde2_blend_mode { QUA_TDE2_BLEND_ZERO = 0x0, QUA_TDE2_BLEND_ONE, QUA_TDE2_BLEND_SRC2COLOR, QUA_TDE2_BLEND_INV_SRC2COLOR, QUA_TDE2_BLEND_SRC2ALPHA, QUA_TDE2_BLEND_INV_SRC2ALPHA, QUA_TDE2_BLEND_SRC1COLOR, QUA_TDE2_BLEND_INV_SRC1COLOR, QUA_TDE2_BLEND_SRC1ALPHA, QUA_TDE2_BLEND_INV_SRC1ALPHA, QUA_TDE2_BLEND_SRC2ALPHASAT, QUA_TDE2_BLEND_BUTT } qua_tde2_blend_mode_t; </pre>
src2_blend_mode	<p>Src2 blend 模式选择,在 eBlendCmd = TDE2_BLEND_CMD_CONFIG 时有效</p>

qua_tde2_opt_t

【说明】


定义 TDE 操作属性。

【定义】

```
typedef struct qua_tde2_opt
```

```

{
    qua_tde2_alucmd_t        alu_cmd;
    qua_tde2_rop_code_t     rop_code_color;
    qua_tde2_rop_code_t     rop_code_alpha;
    qua_tde2_colorkey_mode_t color_key_mode;
    qua_tde2_colorkey_t     color_key_value;
    qua_tde2_clip_mode_t    clip_mode;
    qua_tde2_rect_t         clip_rect;
    QUA_BOOL                 deflicker;
}
    
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```

qua_tde2_deflicker_mode_t  deflicker_mode;
QUA_BOOL                  resize;
qua_tde2_filter_mode_t    filter_mode;
qua_tde2_mirror_t        mirror;
QUA_BOOL                  clut_reload;
QUA_U8                    global_alpha;
qua_tde2_outalpha_from_t  out_alpha_from;
QUA_U32                   colorize;
qua_tde2_blend_opt_t     blend_opt;
qua_tde2_csc_opt_t       csc_opt;
} qua_tde2_opt_t;

```

【描述】

成员名称	描述
alu_cmd	逻辑运算类型，为如下枚举值： <pre> typedef enum qua_tde2_alucmd { QUA_TDE2_ALUCMD_NONE = 0x0, QUA_TDE2_ALUCMD_BLEND = 0x1, QUA_TDE2_ALUCMD_ROP = 0x2, QUA_TDE2_ALUCMD_COLORIZE = 0x4, QUA_TDE2_ALUCMD_BUTT = 0x8 } qua_tde2_alucmd_t; </pre>
rop_code_color	颜色空间 ROP 类型，为如下枚举值： <pre> typedef enum qua_tde2_rop_code { QUA_TDE2_ROP_BLACK = 0, QUA_TDE2_ROP_NOTMERGEPEN, QUA_TDE2_ROP_MASKNOTPEN, QUA_TDE2_ROP_NOTCOPYPEN, QUA_TDE2_ROP_MASKPENNOT, QUA_TDE2_ROP_NOT, QUA_TDE2_ROP_XORPEN, QUA_TDE2_ROP_NOTMASKPEN, QUA_TDE2_ROP_MASKPEN, QUA_TDE2_ROP_NOTXORPEN, QUA_TDE2_ROP_NOP, QUA_TDE2_ROP_MERGENOTPEN, QUA_TDE2_ROP_COPYPEN, QUA_TDE2_ROP_MERGEPENNOT, QUA_TDE2_ROP_MERGEPEN, QUA_TDE2_ROP_WHITE, QUA_TDE2_ROP_BUTT } qua_tde2_rop_code_t; </pre>
rop_code_alpha	alpha 的 ROP 类型
color_key_mode	color key 方式，为如下枚举值：

	<pre>typedef enum qua_tde2_colorkey_mode { QUA_TDE2_COLORKEY_MODE_NONE = 0, QUA_TDE2_COLORKEY_MODE_FOREGROUND, QUA_TDE2_COLORKEY_MODE_BACKGROUND, QUA_TDE2_COLORKEY_MODE_BUTT } qua_tde2_colorkey_mode_t;</pre>
color_key_value	color key 值
clip_mode	clip 模式，为如下枚举值： <pre>typedef enum qua_tde2_clip_mode { QUA_TDE2_CLIPMODE_NONE = 0, QUA_TDE2_CLIPMODE_INSIDE, QUA_TDE2_CLIPMODE_OUTSIDE, QUA_TDE2_CLIPMODE_BUTT } qua_tde2_clip_mode_t;</pre>
clip_rect	clip 区域
deflicker	是否抗闪烁
deflicker_mode	抗闪烁模式，为如下枚举值： <pre>typedef enum qua_tde2_deflicker_mode { QUA_TDE2_DEFLICKER_MODE_NONE = 0, QUA_TDE2_DEFLICKER_MODE_RGB, QUA_TDE2_DEFLICKER_MODE_BOTH, QUA_TDE2_DEFLICKER_MODE_BUTT } qua_tde2_deflicker_mode_t;</pre>
resize	是否进行缩放
filter_mode	图像滤波模式，为如下枚举值： <pre>typedef enum qua_tde2_filter_mode { QUA_TDE2_FILTER_MODE_COLOR = 0, QUA_TDE2_FILTER_MODE_ALPHA, QUA_TDE2_FILTER_MODE_BOTH, QUA_TDE2_FILTER_MODE_NONE, QUA_TDE2_FILTER_MODE_BUTT } qua_tde2_filter_mode_t;</pre>
mirror	镜像类型，为如下枚举值： <pre>typedef enum qua_tde2_mirror { QUA_TDE2_MIRROR_NONE = 0, QUA_TDE2_MIRROR_HORIZONTAL, QUA_TDE2_MIRROR_VERTICAL, QUA_TDE2_MIRROR_BOTH, QUA_TDE2_MIRROR_BUTT } qua_tde2_mirror_t;</pre>



clut_reload	是否重新加载 Clut 表
global_alpha	全局 alpha 值。取值范围：[0, 128]
out_alpha_from	输出 alpha 来源，为如下枚举值： typedef enum qua_tde2_outalpha_from { QUA_TDE2_OUTALPHA_FROM_NORM = 0, QUA_TDE2_OUTALPHA_FROM_BACKGROUND, QUA_TDE2_OUTALPHA_FROM_FOREGROUND, QUA_TDE2_OUTALPHA_FROM_GLOBALALPHA, QUA_TDE2_OUTALPHA_FROM_BUTT } qua_tde2_outalpha_from_t;
colorize	colorize 的值
blend_opt	Alpha 混合操作选项
csc_opt	CSC 参数选项

qua_tde2_fillcolor_t

【说明】

定义 TDE 填充色属性。

【定义】

```
typedef struct qua_tde2_fillcolor
{
    qua_tde2_color_fmt_t    color_fmt;
    QUA_U32                 fill_color;
} qua_tde2_fillcolor_t;
```

【描述】

成员名称	描述
color_fmt	填充色格式
fill_color	填充颜色值

【注意】

- 填充格式必须与图片格式一致
- 填充值必须与填充格式相匹配

qua_vpu_size_t

【说明】

设置视频输入信息

【定义】

```
typedef struct qua_vpu_size {
    qua_size_t size;
    qua_crop_info_t crop_info;
} qua_vpu_size_t;
```

【描述】



成员名称	描述
size	设置宽高尺寸
crop_info	设置裁剪信息

qua_vpss_grp_attr_t

【说明】

创建 GROUP 属性信息

【定义】

```
typedef struct qua_vpss_grp_attr {
    QUA_U32 max_width;
    QUA_U32 max_height;
    QUA_U32 ycmean_en;
    QUA_U32 ycmean_ds;
    qua_pixel_fmt_t pix_format;
    qua_vpss_die_mode_t die_mode;
    QUA_BOOL nr_enable;
    QUA_BOOL en_enable;
    QUA_BOOL apc_enable;
    QUA_BOOL y_gamma_enable;
    QUA_BOOL purple_enable;
    QUA_BOOL cnr_enable;
    QUA_BOOL chroma_enable;
    QUA_BOOL lc_enable;
    QUA_BOOL hist_enable;
    QUA_BOOL ie_enable;
    QUA_BOOL dci_enable;
    QUA_BOOL es_enable;
} qua_vpss_grp_attr_t;
```

【描述】

成员名称	描述
max_width	宽度
max_height	高度
ycmean_en	ycmean 使能
ycmean_ds	ycmean 采样模式
pix_format	像素格式
die_mode	保留
nr_enable	噪声抑制
en_enable	
apc_enable	自适应色彩校正使能
y_gamma_enable	伽玛校正使能
purple_enable	去紫边使能
cnr_enable	颜色噪声抑制使能
chroma_enable	饱和度使能



lc_enable	
hist_enable	直方图均衡使能
ie_enable	图像增强使能
dci_enable	动态色彩插值使能
es_enable	边缘增强使能

qua_vpu_stream_adv_t

【说明】

vpu stream 结构体

【定义】

```
typedef struct qua_vpu_stream_adv {
    qua_vpu_vo_mode_t data_format;
    qua_size_t size;
    QUA_U64 pts;
    QUA_U32 pool_id;
    union
    {
        qua_vpu_yc_data_t frm_blk;
        qua_vpu_yc_data_t frm_scan;
        qua_vpu_yc_data_t frm_nv16;
        qua_vpu_onebuf_data_t frm_tile192;
        qua_vpu_onebuf_data_t frm_tile224;
        qua_vpu_onebuf_data_t frm_tile256;
        qua_vpu_onebuf_data_t frm_rgb888;
        qua_vpu_onebuf_data_t frm_yuyv;
        qua_vpu_onebuf_data_t frm_y;
        qua_vpu_rrgbbb_data_t frm_rrgbbb;
    };
} qua_vpu_stream_adv_t;
```

【描述】

成员名称	描述
data_format	图像格式
size	图像尺寸
pts	时间戳
pool_id	vb pool id(非 VB 模式建议配置为-1UL)
frm_blk	block 格式输出信息
frm_scan	scan(nv12) 格式输出信息
frm_nv16	nv16 格式输出信息
frm_tile192	tile192 格式输出信息
frm_tile224	tile224 格式输出信息
frm_tile256	tile256 格式输出信息
frm_rgb888	rgb888 格式输出信息
frm_yuyv	yuyv 格式输出信息



frm_y	y 格式输出信息
frm_rrgbb	rrgbb 格式输出信息

qua_vpu_chn_config_t

【说明】

vpu 通道配置结构体

【定义】

```
typedef struct qua_vpu_chn_config {
    qua_size_t chn_size;
    qua_crop_info_t crop_area;
    QUA_U32 stride;
    QUA_U32 offset;
    QUA_U32 depth;
} qua_vpu_chn_config_t;
```

【描述】

成员名称	描述
chn_size	通道输出幅面
crop_area	通道裁剪区域
stride	通道输出 stride
offset	通道输出地址偏移
depth	通道输出队列深度

qua_vpu_ext_chn_config_t

【说明】

vpu 扩展通道配置结构体

【定义】

```
typedef struct qua_vpu_ext_chn_config {
    QUA_U32 bind_chn;
    qua_vpu_chn_config_t chn_info;
} qua_vpu_ext_chn_config_t;
```

【描述】


成员名称	描述
bind_chn	绑定通道
chn_info	通道信息

qua_mm_filter_module_t

【说明】

QuaMM Filter 模块表示。

【定义】

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```
typedef struct qua_mm_filter_module {
    struct qua_mm_module_t parent;
} qua_mm_filter_module_t;
```

【描述】

成员名称	描述
parent	父模块信息，包含打开设备的函数指针。 要打开 VGS 设备，open_device 函数的 dev_id 参数传 QUA_MM_FILTER_DEV_VGS，dev_num 参数传 0； 要打开 TDE 设备，open_device 函数的 dev_id 参数传 QUA_MM_FILTER_DEV_TDE，dev_num 参数传 0

【注意】

“mc6870”芯片平台不支持 VGS 设备。

qua_mm_vgs_device_t

【说明】

定义 VGS 设备所涉及的函数接口。VGS 设备下没有通道的概念。

【定义】

```
typedef struct qua_mm_vgs_device {
    struct qua_mm_device_t parent;

    QUA_S32 (*qua_vgs_begin_job)(QUA_VGS_HANDLE *handle);
    QUA_S32 (*qua_vgs_end_job)(QUA_VGS_HANDLE handle);
    QUA_S32 (*qua_vgs_cancel_job)(QUA_VGS_HANDLE handle);
    QUA_S32 (*qua_vgs_add_scale_task)(QUA_VGS_HANDLE handle, qua_vgs_task_attr_t *task_attr);
    QUA_S32 (*qua_vgs_add_draw_line_task)(QUA_VGS_HANDLE handle, qua_vgs_task_attr_t *task_attr,
    qua_vgs_line_t lines[], QUA_U32 num);
    QUA_S32 (*qua_vgs_add_osd_task)(QUA_VGS_HANDLE handle, qua_vgs_task_attr_t *task_attr,
    qua_vgs_osd_t osd[], QUA_U32 num);
    QUA_S32 (*qua_vgs_add_combine_task)(QUA_VGS_HANDLE handle, qua_video_frame_t *frame_in,
    qua_rect_t *rect_in, qua_video_frame_t *frame_out, qua_rect_t *rect_out);
    QUA_S32 (*qua_vgs_decompress)(QUA_VGS_HANDLE handle, const qua_video_frame_t *frame_in,
    qua_video_frame_t *frame_out);
    QUA_S32 (*qua_vgs_format_convert)(QUA_VGS_HANDLE handle, qua_video_frame_t *frame,
    qua_tde2_surface_t *surface, QUA_BOOL frm_to_sur);
    QUA_S32 (*qua_vgs_format_convert_new)(QUA_VGS_HANDLE handle, qua_video_frame_t *frame_in,
    qua_video_frame_t *frame_out);
    QUA_S32 (*qua_vgs_rotate)(qua_vgs_task_attr_t *task_attr, qua_vgs_rotate_t *rotate);
    QUA_S32 (*qua_vgs_rotate_new)(qua_vgs_task_attr_t *task_attr, qua_vgs_rotation_t rotation);
    QUA_S32 (*qua_vgs_fill_color)(qua_video_frame_info_t *frame_info, qua_rect_t *rect, QUA_U32 rgb);
} qua_mm_vgs_device_t;
```

【描述】

成员名称	描述
parent	父设备信息，包含关闭设备的函数指针



qua_vgs_begin_job	创建 VGS 任务
qua_vgs_end_job	提交 VGS 任务，开始处理添加到该任务的操作
qua_vgs_cancel_job	取消指定的 VGS 任务
qua_vgs_add_scale_task	向指定任务添加缩放操作
qua_vgs_add_draw_line_task	向指定任务添加画线操作
qua_vgs_add_osd_task	向指定任务添加视频叠加操作
qua_vgs_add_combine_task	向指定任务添加视频拼接操作
qua_vgs_decompress	向指定任务添加数据解压操作
qua_vgs_format_convert	向指定任务添加 yuv 和 argb 间的格式转换操作
qua_vgs_format_convert_new	向指定任务添加 yuv 和 argb 间的格式转换操作新接口
qua_vgs_rotate	旋转操作
qua_vgs_rotate_new	旋转操作新接口
qua_vgs_fill_color	填充色操作

以上表格里的函数指针成员放在 3.7.2 节“VGS 设备函数接口”里做具体说明。

qua_mm_tde_device_t

【说明】

定义 TDE 设备所涉及的函数接口。TDE 设备下没有通道的概念。

【定义】

```
typedef struct qua_mm_tde_device_t {
    struct qua_mm_device_t parent;

    QUA_TDE_HANDLE (*qua_tde2_begin_job)();
    QUA_S32 (*qua_tde2_end_job)(QUA_TDE_HANDLE handle, QUA_BOOL sync, QUA_BOOL block,
    QUA_U32 timeout);
    QUA_S32 (*qua_tde2_cancel_job)(QUA_TDE_HANDLE handle);
    QUA_S32 (*qua_tde2_wait_for_done)(QUA_TDE_HANDLE handle);
    QUA_S32 (*qua_tde2_wait_all_done)();
    QUA_S32 (*qua_tde2_reset)();
    QUA_S32 (*qua_tde2_set_deflicker_level)(qua_tde_deflicker_level_t deflicker_level);
    QUA_S32 (*qua_tde2_get_deflicker_level)(qua_tde_deflicker_level_t *deflicker_level);
    QUA_S32 (*qua_tde2_set_alpha_threshold_value)(QUA_U8 threshold_value);
    QUA_S32 (*qua_tde2_get_alpha_threshold_value)(QUA_U8 *threshold_value);
    QUA_S32 (*qua_tde2_set_alpha_threshold_state)(QUA_BOOL en_alpha_threshold);
    QUA_S32 (*qua_tde2_get_alpha_threshold_state)(QUA_BOOL *en_alpha_threshold);
    QUA_S32 (*qua_tde2_quick_copy)(QUA_TDE_HANDLE handle,
        qua_tde2_surface_t *src, qua_tde2_rect_t *src_rect,
        qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect);
    QUA_S32 (*qua_tde2_quick_fill)(QUA_TDE_HANDLE handle,
        qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect,
        QUA_U32 fill_data);
    QUA_S32 (*qua_tde2_quick_resize)(QUA_TDE_HANDLE handle,
        qua_tde2_surface_t *src, qua_tde2_rect_t *src_rect,
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```


        qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect);
QUA_S32 (*qua_tde2_quick_deflicker)(QUA_TDE_HANDLE handle,
        qua_tde2_surface_t *src, qua_tde2_rect_t *src_rect,
        qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect);
QUA_S32 (*qua_tde2_bitblit)(QUA_TDE_HANDLE handle,
        qua_tde2_surface_t *back_ground, qua_tde2_rect_t *back_ground_rect,
        qua_tde2_surface_t *fore_ground, qua_tde2_rect_t *fore_ground_rect,
        qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect, qua_tde2_opt_t *opt);
QUA_S32 (*qua_tde2_solid_draw)(QUA_TDE_HANDLE handle,
        qua_tde2_surface_t *fore_ground, qua_tde2_rect_t *fore_ground_rect,
        qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect,
        qua_tde2_fillcolor_t *fill_color, qua_tde2_opt_t *opt);
QUA_S32 (*qua_tde2_compress)(QUA_TDE_HANDLE handle, qua_tde2_surface_t *src,
qua_tde2_surface_t *dst);
QUA_S32 (*qua_tde2_decompress)(QUA_TDE_HANDLE handle, qua_tde2_surface_t *src,
qua_tde2_surface_t *dst);
} qua_mm_tde_device_t;

```

【描述】

成员名称	描述
parent	父设备信息，包含关闭设备的函数指针
qua_tde2_begin_job	创建 TDE 任务
qua_tde2_end_job	提交 TDE 任务，开始处理添加到该任务的操作
qua_tde2_cancel_job	取消指定 TDE 任务
qua_tde2_wait_for_done	等待指定 TDE 任务的完成
qua_tde2_wait_all_done	等待所有 TDE 任务的完成
qua_tde2_reset	复位 TDE 所有状态
qua_tde2_set_deflicker_level	设置抗闪烁级别
qua_tde2_get_deflicker_level	获取抗闪烁级别
qua_tde2_set_alpha_threshold_value	设置 alpha 判决阈值
qua_tde2_get_alpha_threshold_value	获取 alpha 判决阈值
qua_tde2_set_alpha_threshold_state	设置 alpha 判决状态
qua_tde2_get_alpha_threshold_state	获取 alpha 判决状态
qua_tde2_quick_copy	向指定任务添加快速拷贝操作
qua_tde2_quick_fill	向指定任务添加快速填充操作
qua_tde2_quick_resize	向指定任务添加快速缩放操作
qua_tde2_quick_deflicker	向指定任务添加快速抗闪烁操作
qua_tde2_bitblit	向指定任务添加带附加功能的搬移操作
qua_tde2_solid_draw	向指定任务添加画点画线操作
qua_tde2_compress	向指定任务添加数据压缩操作
qua_tde2_decompress	向指定任务添加数据解压缩操作

以上表格里的函数指针成员放在 3.7.3 节“TDE 设备函数接口”里做具体说明。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024. 08. 01
	版本 Rev.	1. 1. 0

qua_mm_vppu_device_t


【说明】

定义 VPPU 设备所涉及的函数接口。该设备下定义的通道有：VPU 通道、VGS 通道、VPSS 通道。

【定义】

```
typedef struct qua_mm_vppu_device {
    struct qua_mm_device_t parent;
    QUA_S32 (*init)();
    QUA_S32 (*exit)();
    QUA_S32 (*get_grp_fd)(QUA_GROUP group_id);
    QUA_S32 (*set_vi_attr)(QUA_GROUP group_id, qua_vpu_size_t *size);
    QUA_S32 (*get_vi_attr)(QUA_GROUP group_id, qua_vpu_size_t *size);
    QUA_S32 (*create_grp)(QUA_GROUP group_id, qua_vpss_grp_attr_t *grp_attr);
    QUA_S32 (*destroy_grp)(QUA_GROUP group_id);
    QUA_S32 (*enable)(QUA_GROUP group_id, qua_vpu_vi_mode_t vi_mode);
    QUA_S32 (*disable)(QUA_GROUP group_id);
    QUA_S32 (*freeze_video)(QUA_GROUP group_id);
    QUA_S32 (*unfreeze_video)(QUA_GROUP group_id);
    QUA_S32 (*set_global_ldc)(QUA_GROUP group_id, qua_ldc_t *ldc);
    QUA_S32 (*get_global_ldc)(QUA_GROUP group_id, qua_ldc_t *ldc);
    QUA_S32 (*set_global_graph_v2)(QUA_GROUP group_id, qua_logov2_t *logov2);
    QUA_S32 (*get_global_graph_v2)(QUA_GROUP group_id, qua_logov2_t *logov2);
    QUA_S32 (*get_grp_attr)(QUA_GROUP group_id, qua_vpss_grp_attr_t *grp_attr);
    QUA_S32 (*set_grp_attr)(QUA_GROUP group_id, qua_vpss_grp_attr_t *grp_attr);
    QUA_S32 (*start_grp)(QUA_GROUP group_id);
    QUA_S32 (*stop_grp)(QUA_GROUP group_id);
    QUA_S32 (*set_ie)(QUA_GROUP group_id, qua_vpss_ie_param_t *ie_param);
    QUA_S32 (*get_ie)(QUA_GROUP group_id, qua_vpss_ie_param_t *ie_param);
    QUA_S32 (*set_chroma)(QUA_GROUP group_id, qua_vpss_chrome_info_t *chroma_info);
    QUA_S32 (*get_chroma)(QUA_GROUP group_id, qua_vpss_chrome_info_t *chroma_info);
    QUA_S32 (*set_lc)(QUA_GROUP group_id, qua_vpss_lc_info_t *lc_info);
    QUA_S32 (*get_lc)(QUA_GROUP group_id, qua_vpss_lc_info_t *lc_info);
    QUA_S32 (*set_grp_effect)(QUA_GROUP group_id, qua_vpss_grp_effect_t *grp_effect);
    QUA_S32 (*get_grp_effect)(QUA_GROUP group_id, qua_vpss_grp_effect_t *grp_effect);
    QUA_S32 (*set_src_split_info)(QUA_GROUP group_id, qua_vpss_src_split_info_t *split_info);
    QUA_S32 (*get_src_split_info)(QUA_GROUP group_id, qua_vpss_src_split_info_t *split_info);
    QUA_S32 (*send_frame)(QUA_GROUP group_id, qua_video_frame_info_t *frame, QUA_S32 timeout);
    QUA_S32 (*send_user_pic)(QUA_GROUP group_id, qua_vpu_user_pic_t *pic);
    QUA_S32 (*enable_user_frame_rate_ctrl)(QUA_GROUP group_id);
    QUA_S32 (*disable_user_frame_rate_ctrl)(QUA_GROUP group_id);
    QUA_S32 (*get_grp_frame)(QUA_GROUP group_id, qua_video_frame_info_t *frame, QUA_S32
frame_index);
    QUA_S32 (*release_grp_frame)(QUA_GROUP group_id, qua_video_frame_info_t *frame);

```


 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024. 08. 01
	版本 Rev.	1. 1. 0

```

QUA_S32 (*image_convert)(const qua_video_frame_t *frame_in, qua_video_frame_t *frame_out);
} qua_mm_vppu_device_t;

```

【描述】

成员名称	描述
init	VPU 模块初始化
exit	VPU 模块退出
get_grp_fd	获取 group 号
set_vi_attr	设置视频输入属性
get_vi_attr	获取视频输入属性
create_grp	创建视频输入
destroy_grp	销毁视频输入
enable	使能视频处理模块
disable	关闭视频处理模块
freeze_video	视频冻结, 视频处理模块将会向后面数据处理模块, 不断发送最后一帧的数据. 直到调用#FH_VPSS_UnfreezeVideo
unfreeze_video	视频解冻
set_global_ldc	设置镜头畸变校正 (LDC) 属性
get_global_ldc	获取镜头畸变校正 (LDC) 属性
set_global_graph_v2	设置全局图形叠加信息
get_global_graph_v2	获取全局图形叠加信息
get_grp_attr	获取视频输入组属性
set_grp_attr	设置视频输入组属性
start_grp	启动 group 视频模块处理
stop_grp	停止 group 视频模块处理
set_ie	设置图像增强属性
get_ie	获取图像增强属性
set_chroma	设置图像色度
get_chroma	获取图像色度
set_lc	设置图像亮度
get_lc	获取图像亮度
set_grp_effect	设置图像效果
get_grp_effect	获取图像效果
set_src_split_info	设置源图像分离信息
get_src_split_info	获取源图像分离信息
send_frame	用户向视频处理模块发送数据
send_user_pic	提交用户图像给视频处理模块进行处理, 仅在处于非直通模式下有效
enable_user_frame_rate_ctrl	启动用户帧率控制
disable_user_frame_rate_ctrl	禁用用户帧率控制
get_grp_frame	获取 GROUP 图像
release_grp_frame	释放 GROUP 图像
image_convert	图像转换

以上表格里的函数指针成员放在 3.7.4 节“VPPU 设备函数接口”里做具体说明。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_mm_vpu_channel


【说明】

定义 VPU 通道所涉及的函数接口。qm10xh 支持该接口，其他芯片平台不支持该接口。

【定义】

```
typedef struct qua_mm_vpu_channel {
    struct qua_mm_channel_t parent;
    QUA_S16 chn_id;
    QUA_GROUP group_id;

    QUA_S32 (*get_chn_frame_adv)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_stream_adv_t
    *stream_adv, QUA_U32 timeout_ms);
    QUA_S32 (*lock_chn_frame_adv)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_stream_adv_t
    *stream_adv, QUA_U32 timeout_ms, QUA_U32 *handle_lock);
    QUA_S32 (*get_chn_frame_adv_norpt)(QUA_GROUP group_id, QUA_U32 chn_id,
    qua_vpu_stream_adv_t *stream_adv, QUA_U32 timeout_ms);
    QUA_S32 (*lock_chn_frame_adv_norpt)(QUA_GROUP group_id, QUA_U32 chn_id,
    qua_vpu_stream_adv_t *stream_adv, QUA_U32 timeout_ms, QUA_U32 *handle_lock);
    QUA_S32 (*unlock_chn_frame_adv)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_stream_adv_t
    *stream_adv, QUA_U32 lock_handle);
    QUA_S32 (*set_chn_attr)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_chn_config_t *chn_cfg);
    QUA_S32 (*get_chn_attr)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_chn_config_t *chn_cfg);
    QUA_S32 (*set_ext_chn_attr)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_ext_chn_config_t
    *ext_chn_cfg);
    QUA_S32 (*get_ext_chn_attr)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_ext_chn_config_t
    *ext_chn_cfg);
    QUA_S32 (*set_vo_mode)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_vo_mode_t mode);
    QUA_S32 (*set_vo_rotate)(QUA_GROUP group_id, QUA_U32 chn_id, qua_rotate_t rotate);
    QUA_S32 (*open_chn)(QUA_GROUP group_id, QUA_U32 chn_id);
    QUA_S32 (*close_chn)(QUA_GROUP group_id, QUA_U32 chn_id);
    QUA_S32 (*set_frame_ctrl)(QUA_GROUP group_id, QUA_U32 chn_id, QUA_DOUBLE *frame_ctrl);
    QUA_S32 (*get_frame_ctrl)(QUA_GROUP group_id, QUA_U32 chn_id, QUA_DOUBLE *frame_ctrl);
    QUA_S32 (*get_framerate)(QUA_GROUP group_id, QUA_U32 chn_id, QUA_DOUBLE *fps);
    QUA_S32 (*set_chn_crop)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_crop_sel_t sel,
    qua_crop_info_t *crop_info);
    QUA_S32 (*get_chn_crop)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_crop_sel_t sel,
    qua_crop_info_t *crop_info);
    QUA_S32 (*set_scaler_coeff)(QUA_GROUP group_id, QUA_U32 chn_id, QUA_U32 level);
    QUA_S32 (*get_frame_buffer_size)(QUA_GROUP group_id, QUA_U32 chn_id, QUA_U32 bufnum,
    QUA_U32 *needsz);
    QUA_S32 (*framebuffer_register)(QUA_GROUP group_id, QUA_U32 chn_id, QUA_U32 bufnum,
    qua_mem_info_t buf);
    QUA_S32 (*framebuffer_unregister)(QUA_GROUP group_id, QUA_U32 chn_id, qua_mem_info_t
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024. 08. 01
	版本 Rev.	1. 1. 0

*buf);

```

    QUA_S32 (*attach_vb_pool)(QUA_GROUP group_id, QUA_U32 chn_id, QUA_U32 vbpool);
    QUA_S32 (*detach_vb_pool)(QUA_GROUP group_id, QUA_U32 chn_id);
    QUA_S32 (*set_comm_vb_enable)(QUA_GROUP group_id, QUA_U32 chn_id, QUA_U32 enable);
    QUA_S32 (*set_chn_graph_v2)(QUA_GROUP group_id, QUA_U32 chn_id, qua_logov2_t *logov2);
    QUA_S32 (*get_chn_graph_v2)(QUA_GROUP group_id, QUA_U32 chn_id, qua_logov2_t *logov2);
    QUA_S32 (*check_addr)(QUA_GROUP group_id, QUA_U32* addr, QUA_U32 size, QUA_U32
timeout_ms, QUA_U32 *isidle);
    QUA_S32 (*set_chn_vi_sel)(QUA_GROUP group_id, QUA_U32 chn_id, QUA_U32 sel);
    QUA_S32 (*set_chn_apc_attr)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_apc_t *attr);
    QUA_S32 (*get_chn_apc_attr)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_apc_t *attr);
    QUA_S32 (*set_rgb_pre_attr)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_rgb_pre_t *attr);
    QUA_S32 (*get_rgb_pre_attr)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_rgb_pre_t *attr);
} qua_mm_vpu_channel_t;

```

【描述】

成员名称	描述
parent	
chn_id	通道号
group_id	GROUP 号
get_chn_frame_adv	获取视频处理模块输出通道的图像数据
lock_chn_frame_adv	获取视频处理模块通道输出的图像数据，并锁定此缓存不被硬件继续使用。建议至少分配 3buf 及以上才使用此接口
get_chn_frame_adv_norpt	获取视频处理模块通道输出的图像数据
lock_chn_frame_adv_norpt	获取视频处理模块通道输出的图像数据，并锁定此缓存不被硬件继续使用。建议至少分配 3buf 及以上才使用此接口
unlock_chn_frame_adv	释放被 lock_chn_frame_adv 锁定的缓存
set_chn_attr	设置视频处理模块通道属性
get_chn_attr	获取视频处理模块通道属性
set_ext_chn_attr	设置视频处理模块扩展通道属性
set_vo_mode	设置视频处理模块通道输出组织格式
set_vo_rotate	根据后续编码模块处理的旋转角度设置视频处理模块通道输出组织方式
open_chn	开启通道使能
close_chn	关闭通道使能
set_frame_ctrl	设置帧率控制参数
get_frame_ctrl	获取帧率控制参数
get_framerate	获取当前通道帧率统计结果
set_chn_crop	设置通道裁剪属性
get_chn_crop	获取通道裁剪属性
set_scaler_coeff	设置图像缩放的滤波器系数，可以将图像效果从平滑到锐利
get_frame_buffer_size	获取对应通道设的内存块数量需要的内存大小



framebuffer_register	给通道输出注册临时内存
framebuffer_unregister	将通道输出临时内存注销
attach_vb_pool	将 VPSS 的通道绑定到某个视频缓存 VB 池中
detach_vb_pool	将 VPSS 的通道从某个视频缓存 VB 池中解绑
set_comm_vb_enable	设置是否允许 VPU 通道使用 CommonVB
set_chn_graph_v2	设置通道图形叠加信息
get_chn_graph_v2	获取通道图形叠加信息
check_addr	检查 VPSS 输入缓存地址是否被使用
set_chn_vi_sel	设置通道输入图像源位置
set_chn_apc_attr	设置通道 APC 属性
get_chn_apc_attr	获取通道 APC 属性
set_rgb_pre_attr	设置 RGB 前处理属性
get_rgb_pre_attr	获取 RGB 前处理属性

以上表格里的函数指针成员放在 3.7.5 节“VPU 通道函数接口”里做具体说明。

qua_mm_vgs_channel_t

【说明】

定义 VGS 通道所涉及的函数接口。mc331x 支持该接口，其他芯片平台不支持该接口。

【定义】

```
typedef struct qua_mm_vgs_channel {
    struct qua_mm_channel_t parent;

    QUA_S32 (*set_chn_sync_mode)(QUA_S32 vgs_chn_id, QUA_BOOL a_sync, QUA_U32 buf_depth);
    QUA_S32 (*set_chn_mode)(QUA_S32 vgs_chn_id, QUA_S32 vgs_path, qua_vgs_chn_mode_t
*chn_mode);
    QUA_S32 (*get_chn_mode)(QUA_S32 vgs_chn_id, QUA_S32 vgs_path, qua_vgs_chn_mode_t
*chn_mode);
    QUA_S32 (*set_chn_crop)(QUA_S32 vgs_chn_id, qua_vgs_crop_t *crop_info);
    QUA_S32 (*get_chn_crop)(QUA_S32 vgs_chn_id, qua_vgs_crop_t *crop_info);
    QUA_S32 (*set_chn_rotate)(QUA_S32 vgs_chn_id, QUA_S32 vgs_path, qua_rotate_t rotate);
    QUA_S32 (*get_chn_rotate)(QUA_S32 vgs_chn_id, QUA_S32 vgs_path, qua_rotate_t *rotate);
    QUA_S32 (*send_frame)(QUA_S32 vgs_chn_id, const qua_video_frame_info_t *frame, QUA_BOOL
play_mode, QUA_U32 timeout);
    QUA_S32 (*query)(QUA_S32 vgs_chn_id, qua_vgs_chn_status_t *chn_sts);
} qua_mm_vgs_channel_t;
```

【描述】

成员名称	描述
parent	qua_mm_channel 指针
set_chn_sync_mode	设置通道同步模式
set_chn_mode	设置通道模式
get_chn_mode	获取通道模式
set_chn_crop	设置通道裁剪信息
get_chn_crop	获取通道裁剪信息



set_chn_rotate	设置通道旋转模式
get_chn_rotate	获取通道旋转模式
send_frame	用户向 vppu 图像处理模块传送图像数据
query	查询 VGS 通道状态

qua_mm_vpss_channel_t

【说明】

定义 VPSS 通道所涉及的函数接口。mc6870 支持该接口，其他芯片平台不支持该接口。

【定义】

```
typedef struct qua_mm_vpss_channel {
    struct qua_mm_channel_t parent;

    QUA_S32 (*get_chn_fd)(QUA_GROUP group_id, QUA_S32 chn_id);
    QUA_S32 (*enable_chn)(QUA_GROUP group_id, QUA_S32 chn_id);
    QUA_S32 (*disable_chn)(QUA_GROUP group_id, QUA_S32 chn_id);
    QUA_S32 (*set_chn_attr)(QUA_GROUP group_id, QUA_S32 chn_id, qua_vpss_chn_attr_t *chn_attr);
    QUA_S32 (*get_chn_attr)(QUA_GROUP group_id, QUA_S32 chn_id, qua_vpss_chn_attr_t *chn_attr);
    QUA_S32 (*set_chn_mode)(QUA_GROUP group_id, QUA_S32 chn_id, qua_vpss_chn_mode_t
*chn_mode);
    QUA_S32 (*get_chn_mode)(QUA_GROUP group_id, QUA_S32 chn_id, qua_vpss_chn_mode_t
*chn_mode);
    QUA_S32 (*set_mosaic)(QUA_GROUP group_id, QUA_S32 chn_id, qua_vpss_mask_info_t *mosaic);
    QUA_S32 (*get_mosaic)(QUA_GROUP group_id, QUA_S32 chn_id, qua_vpss_mask_info_t *mosaic);
    QUA_S32 (*set_pic)(QUA_GROUP group_id, QUA_S32 chn_id, qua_vpss_logo_info_t *logo);
    QUA_S32 (*get_pic)(QUA_GROUP group_id, QUA_S32 chn_id, qua_vpss_logo_info_t *logo);
    QUA_S32 (*set_pic_color)(QUA_GROUP group_id, QUA_S32 chn_id, qua_vpss_logo_color_info_t
*color_info);
    QUA_S32 (*get_pic_color)(QUA_GROUP group_id, QUA_S32 chn_id, qua_vpss_logo_color_info_t
*color_info);
    QUA_S32 (*set_depth)(QUA_GROUP group_id, QUA_S32 chn_id, QUA_U32 depth);
    QUA_S32 (*get_depth)(QUA_GROUP group_id, QUA_S32 chn_id, QUA_U32 *depth);
    QUA_S32 (*get_chn_frame)(QUA_GROUP group_id, QUA_S32 chn_id, qua_video_frame_info_t *frame,
QUA_S32 timeout);
    QUA_S32 (*release_chn_frame)(QUA_GROUP group_id, QUA_S32 chn_id, qua_video_frame_info_t
*frame);
} qua_mm_vpss_channel_t;
```

【描述】

成员名称	描述
parent	
get_chn_fd	获取通道句柄
enable_chn	使能通道
disable_chn	关闭通道
set_chn_attr	设置通道属性



get_chn_attr	获取通道属性
set_chn_mode	设置通道模式
get_chn_mode	获取通道模式
set_mosaic	设置通道马赛克信息
get_mosaic	获取通道马赛克信息
set_pic	设置通道图片叠加信息
get_pic	获取通道图片叠加信息
set_pic_color	设置通道图像颜色
get_pic_color	获取通道图像颜色
set_depth	设置通道图像深度
get_depth	获取通道图像深度
get_chn_frame	获取通道图像帧信息
release_chn_frame	释放通道图像帧

3.7.2 VGS 设备函数接口

qua_vgs_begin_job

【说明】

创建一个 VGS 任务。

QUA_S32 (*qua_vgs_begin_job)(QUA_VGS_HANDLE *handle)			
参数名称	输入/输出	描述	返回值
handle	输出	用以返回任务句柄	成功: QUA_SUCCESS; 失败: 其它值

【注意】

必须判断函数返回成功后才能使用 handle。

qua_vgs_end_job

【说明】

提交 VGS 任务，开始处理添加到该任务的操作。


QUA_S32 (*qua_vgs_end_job)(QUA_VGS_HANDLE handle)			
参数名称	输入/输出	描述	返回值
handle	输入	任务句柄	成功: QUA_SUCCESS; 失败: 其它值

【注意】

如果此接口返回失败，必须调用 qua_vgs_cancel_job 接口取消掉 handle 标识的 job。否则会导致 handle 标识的 job 不能再被循环利用。

qua_vgs_cancel_job

【说明】

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

取消指定的 VGS 任务。

QUA_S32 (*qua_vgs_cancel_job)(QUA_VGS_HANDLE handle)			
参数名称	输入/输出	描述	返回值
handle	输入	任务句柄	成功: QUA_SUCCESS; 失败: 其它值

qua_vgs_add_scale_task

【说明】

向指定任务添加缩放操作。

QUA_S32 (*qua_vgs_add_scale_task)(QUA_VGS_HANDLE handle, qua_vgs_task_attr_t *task_attr)			
参数名称	输入/输出	描述	返回值
handle	输入	任务句柄	成功: QUA_SUCCESS; 失败: 其它值
task_attr	输入输出	VGS task 属性	

【注意】

- 如果此接口返回失败，如不需再添加其他任务，可以调用 qua_vgs_end_job 接口提交已经添加的 task，否则必须调用 qua_vgs_cancel_job 接口取消掉 handle 标识的 job。否则会导致 handle 标识的 job 不能再被循环利用。
- task 任务支持非 VB 内存的物理地址，此时用户只需配置正确的物理地址即可，无需配置 PoolId 和虚拟地址，但用户需要自行保证物理连续的内存大小足够。

qua_vgs_add_draw_line_task

【说明】

向指定任务添加画线操作。

QUA_S32 (*qua_vgs_add_draw_line_task)(QUA_VGS_HANDLE handle, qua_vgs_task_attr_t *task_attr, qua_vgs_line_t lines[], QUA_U32 num)			
参数名称	输入/输出	描述	返回值
handle	输入	任务句柄	成功: QUA_SUCCESS; 失败: 其它值
task_attr	输入输出	VGS task 属性	
lines	输入	画线属性配置结构体数组	
num	输入	画线属性数组的成员数	

qua_vgs_add_osd_task

【说明】

向指定任务添加叠加操作。

QUA_S32 (*qua_vgs_add_osd_task)(QUA_VGS_HANDLE handle, qua_vgs_task_attr_t *task_attr, qua_vgs_osd_t osd[], QUA_U32 num)			
参数名称	输入/输出	描述	返回值
handle	输入	任务句柄	成功: QUA_SUCCESS;



task_attr	输入输出	VGS task 属性	失败：其它值
osd	输入	叠加属性配置结构体数组	
num	输入	叠加属性数组的成员数	

qua_vgs_add_combine_task

【说明】

向指定任务添加视频拼接任务。

QUA_S32 (*qua_vgs_add_combine_task)(QUA_VGS_HANDLE handle, qua_video_frame_t *frame_in, qua_rect_t *rect_in, qua_video_frame_t *frame_out, qua_rect_t *rect_out)

参数名称	输入/输出	描述	返回值
handle	输入	任务句柄	成功：QUA_SUCCESS； 失败：其它值
frame_in	输入	输入视频帧	
rect_in	输入	用于拼接的区域	
frame_out	输出	输出视频帧	
rect_out	输入	拼接到的位置区域	

qua_vgs_decompress

【说明】

向指定任务添加数据解压操作。

QUA_S32 (*qua_vgs_decompress)(QUA_VGS_HANDLE handle, const qua_video_frame_t *frame_in, qua_video_frame_t *frame_out)

参数名称	输入/输出	描述	返回值
handle	输入	任务句柄	成功：QUA_SUCCESS； 失败：其它值
frame_in	输入	待解压的帧数据	
frame_out	输出	解压后的帧数据	

【注意】

- “qm10xh”芯片平台不支持该接口。
- frame_in 和 frame_out 的尺寸、格式必须一致。

qua_vgs_format_convert

【说明】

向指定任务添加格式转换操作，支持 yuv 转 argb，或者 argb 转 yuv。

QUA_S32 (*qua_vgs_format_convert)(QUA_VGS_HANDLE handle, qua_video_frame_t *frame, qua_tde2_surface_t *surface, QUA_BOOL frm_to_sur)

参数名称	输入/输出	描述	返回值
handle	输入	任务句柄	成功：QUA_SUCCESS； 失败：其它值
frame	输入输出	YUV 格式数据	
surface	输入输出	ARGB 格式数据	
frm_to_sur	输入	为 QUA_TRUE，则 YUV 转成	



ARGB; 为 QUA_FALSE, 则 ARGB
转成 YUV

【注意】

- "qm10xh"芯片平台不支持该接口。
- frame 和 surface 对应的尺寸必须一致。

qua_vgs_format_convert_new**【说明】**

向指定任务添加格式转换操作新接口。

QUA_S32 (*qua_vgs_format_convert_new)(QUA_VGS_HANDLE handle, qua_video_frame_t *frame_in, qua_video_frame_t *frame_out)			
参数名称	输入/输出	描述	返回值
handle	输入	任务句柄	成功: QUA_SUCCESS; 失败: 其它值
frame_in	输入	原始图像	
frame_out	输出	目标图像	

【注意】

"mc331x"芯片平台不支持该接口。

qua_vgs_rotate**【说明】**

旋转操作。

QUA_S32 (*qua_vgs_rotate)(qua_vgs_task_attr_t *task_attr, qua_vgs_rotate_t *rotate)			
参数名称	输入/输出	描述	返回值
task_attr	输入输出	Task 属性	成功: QUA_SUCCESS; 失败: 其它值
rotate	输入	旋转属性	

【注意】

"qm10xh"芯片平台不支持该接口。

qua_vgs_rotate_new**【说明】**

旋转操作新接口。

QUA_S32 (*qua_vgs_rotate_new)(qua_vgs_task_attr_t *task_attr, qua_vgs_rotation_t rotation)			
参数名称	输入/输出	描述	返回值
task_attr	输入输出	Task 属性	成功: QUA_SUCCESS; 失败: 其它值
rotation	输入	旋转方向, 为如下枚举值: typedef enum qua_vgs_rotation { QUA_VGS_ROTATION_NONE, QUA_VGS_ROTATION_90, };	



	QUA_VGS_ROTATION_180, QUA_VGS_ROTATION_270, QUA_VGS_ROTATION_MIRROR, QUA_VGS_ROTATION_FLIP, QUA_VGS_ROTATION_MIR_ROT, QUA_VGS_ROTATION_FLP_ROT, QUA_VGS_ROTATION_BUTT } qua_vgs_rotation_t;
--	--

【注意】

“mc331x”芯片平台不支持该接口。

qua_vgs_fill_color

【说明】

填色操作。

QUA_S32 (*qua_vgs_fill_color)(qua_video_frame_info_t *frame_info, qua_rect_t *rect, QUA_U32 rgb)

参数名称	输入/输出	描述	返回值
frame_info	输入	原始图像帧	成功: QUA_SUCCESS; 失败: 其它值
rect	输入	原始图像的待填充区域	
rgb	输入	填充颜色的 rgb888 值	

【注意】

- “mc331x”芯片平台的“linux”操作系统不支持该接口。
- 实际填充的是 YUV 数据，内部会将 RGB888 转成 YUV 处理。

3.7.3 TDE 设备函数接口

qua_tde2_begin_job

【说明】


新建一个 TDE 任务。

QUA_TDE_HANDLE (*qua_tde2_begin_job)()

参数名称	输入/输出	描述	返回值
			失败 : QUA_TDE2_INVALID_HANDLE 成功: 其它值

【注意】

一个 TDE 任务可以包含多个 TDE 操作,提交任务后, TDE 操作按添加的顺序执行。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_tde2_end_job

【说明】

提交 TDE 任务，可指定为是否同步或者是否阻塞，阻塞时可以设置等待时间。

QUA_S32 (*qua_tde2_end_job)(QUA_TDE_HANDLE handle, QUA_BOOL sync, QUA_BOOL block, QUA_U32 timeout)			
参数名称	输入/输出	描述	返回值
handle	输入	TDE 任务句柄	成功: QUA_SUCCESS; 失败: 其它值
sync	输入	是否同步	
block	输入	是否阻塞	
timeout	输入	超时时间, 单位: ms	

【注意】

- sync 设为 QUA_TRUE 时, 该函数调用不会立即返回, 而会等到对应 job 中的所有操作处理完成后再返回, block 和 timeout 参数无效;
- sync 设为 QUA_FALSE 时, 此时函数何时返回需结合 block 和 timeout 参数来确定;
- block 设为 QUA_TRUE 时, 该函数不会立即返回, 只有在对应 job 中的操作都完成、等待超时或者等待被打断时才会返回。超时返回时, 若任务没有完成, 则还会继续执行直到完成;
- block 设为 QUA_FALSE 时, 该函数会立即返回。
- 提交任务后, 此任务对应的 handle 会变为无效, 再次提交会返回失败。

qua_tde2_cancel_job

【说明】

取消指定的 TDE 任务, 包括已加入到该任务中的操作。

QUA_S32 (*qua_tde2_cancel_job)(QUA_TDE_HANDLE handle)			
参数名称	输入/输出	描述	返回值
handle	输入	TDE 任务句柄	成功: QUA_SUCCESS; 失败: 其它值

【注意】

- 在向 TDE 任务添加操作时, 如果添加失败, 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。
- 已经提交的任务不能再取消。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。
- 当添加操作出错时, 必须调用此接口进行取消, 否则会造成 job 的堆积, 最终导致没有空闲的 job 可用而使 TDE 无法处理新的任务。

qua_tde2_quick_copy

【说明】

向指定任务添加快速拷贝操作。将指定基地址的表面的指定区域, 拷贝到确定目的地址的指定区域的内存中。

QUA_S32 (*qua_tde2_quick_copy)(QUA_TDE_HANDLE handle, qua_tde2_surface_t *src,
--



qua_tde2_rect_t *src_rect, qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect)

参数名称	输入/输出	描述	返回值
handle	输入	TDE 任务句柄	成功: QUA_SUCCESS; 失败: 其它值
src	输入	源表面	
src_rect	输入	源表面操作区域	
dst	输出	目标表面	
dst_rect	输入	目标表面操作区域	

【注意】

- 表面信息由 qua_tde2_surface_t 确定，其包含表面的基本信息：像素宽度、像素高度、每行的跨度、颜色格式、存放的物理地址等；
- 操作区域由 qua_tde2_rect_t 确定，其描述表面中参与本次操作的矩形范围：起始位置和尺寸信息；
- 通过指定不同的操作区域，用户可指定表面的全部或部分参与操作。
- 此接口使用的是直接 DMA 搬移，因此性能优于 qua_tde2_bitblit 搬移。
- 快速拷贝操作不支持格式转换，源表面和目标表面格式必须一致。
- 快速拷贝不支持缩放功能，因此如果源和目的的操作区域尺寸不一致，则按照两者最小的公共区域进行拷贝搬移。
- 操作区域要和表面有公共区域，否则会返回错误。

qua_tde2_quick_fill**【说明】**

向指定任务添加快速填充操作。将数据值填充到以确定目的地址的指定操作区域的内存中，可实现颜色填充的功能。

QUA_S32 (*qua_tde2_quick_fill)(QUA_TDE_HANDLE handle, qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect, QUA_U32 fill_data)

参数名称	输入/输出	描述	返回值
handle	输入	TDE 任务句柄	成功: QUA_SUCCESS; 失败: 其它值
dst	输入输出	目标表面	
dst_rect	输入	目标表面操作区域	
fill_data	输入	填充值	

【注意】

此操作直接将 fill_data 填充在表面的指定区域内，故填充的颜色值必须与表面格式对应。

qua_tde2_quick_resize**【说明】**

向指定任务添加快速缩放操作。将确定基地址的表面以指定源区域的尺寸缩放至指定目标区域的尺寸，并将结果拷贝到确定目的地址对应的操作区域的内存中。

QUA_S32 (*qua_tde2_quick_resize)(QUA_TDE_HANDLE handle, qua_tde2_surface_t *src, qua_tde2_rect_t *src_rect, qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect)

参数名称	输入/输出	描述	返回值
handle	输入	TDE 任务句柄	成功: QUA_SUCCESS; 失败: 其它值
src	输入	源表面	



src_rect	输入	源表面操作区域	
dst	输出	目标表面	
dst_rect	输入	目标表面操作区域	

【注意】

- 缩放时源表面和目标表面可以为同一表面，但操作区域不能有重叠。
- 如果源表面和目标表面的格式不相同，则自动进行格式转换。

qua_tde2_quick_deflicker

【说明】

向指定任务添加快速抗闪烁操作。将确定基地址的表面的指定区域进抗闪烁，将结果拷贝到确定目的地址的表面所指定的操作区域的内存中。

QUA_S32 (*qua_tde2_quick_deflicker)(QUA_TDE_HANDLE handle, qua_tde2_surface_t *src, qua_tde2_rect_t *src_rect, qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect)

参数名称	输入/输出	描述	返回值
handle	输入	TDE 任务句柄	成功: QUA_SUCCESS; 失败: 其它值
src	输入	源表面	
src_rect	输入	源表面操作区域	
dst	输出	目标表面	
dst_rect	输入	目标表面操作区域	

【注意】

- 抗闪烁只按垂直方向进行滤波。
- 抗闪烁源表面和目标表面可以为同一表面，但操作区域不能有重叠。
- 如果指定的输入区域与输出不一致，则会进行缩放处理。
- 如果源表面和目标表面的格式不相同，则会进行格式转换处理。


qua_tde2_bitblit

【说明】

向指定任务添加带附加功能的搬移操作。将前景表面与背景表面的指定区域进行运算，将运算后的表面拷贝到目标表面的指定区域中。

QUA_S32 (*qua_tde2_bitblit)(QUA_TDE_HANDLE handle, qua_tde2_surface_t *back_ground, qua_tde2_rect_t *back_ground_rect, qua_tde2_surface_t *fore_ground, qua_tde2_rect_t *fore_ground_rect, qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect, qua_tde2_opt_t *opt)

参数名称	输入/输出	描述	返回值
handle	输入	TDE 任务句柄	成功: QUA_SUCCESS; 失败: 其它值
back_ground	输入	背景表面	
back_ground_rect	输入	背景表面操作区域	
fore_ground	输入	前景表面	
fore_ground_rect	输入	前景表面操作区域	
dst	输出	目标表面	
dst_rect	输入	目标表面操作区域	
opt	输入	附加参数	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

【注意】

- 背景表面的指定区域和目标表面的指定区域必须一致。
- 附加功能信息由 qua_tde2_opt_t 结构确定，其包括是否缩放、是否抗闪烁、是否镜像、是否进行 alpha 混合等，且这些附加功能可以同时进行。
- Alpha 混合操作是指将前景表面和背景表面的像素值按照指定的 alpha 值进行加权求和，得到 alpha 混和后的输出表面，达到两个表面按照一定透明度叠加的效果。输出表面的 alpha 值取决于用户设置的 alpha 混合命令，全局 Alpha 会参与整个叠加过程。
- 当前景表面中的操作区域若和目的表面的操作区域大小不一致时，若指定 qua_tde2_opt_t 中 resize 使能，即将前景的操作区域缩放到目的表面的操作区域大小，再和背景做其他运算；若 resize 未使能，则不会进行前景操作区域的缩放，而是将前景、背景和目的表面的操作区域三者求最小，并将该最小区域作为三个表面的真正操作区域。
- 抗闪烁操作指是否对前景表面的操作区域做抗闪处理，然后再和背景 alpha 叠加等运算，可通过 qua_tde2_opt_t 中 deflicker 指定。
- 镜像功能指将输出结果做水平/垂直方向上的中心对称处理。调用者可通过 qua_tde2_opt_t 中 mirror 配置项指定。
- clip 裁剪功能可以在输出图像时，指定输出其中的一部分到目的表面，相当于对输出结果做了裁剪后才输出。
- 目标表面必须与背景表面的颜色格式一致，前景表面的颜色格式可以与背景/目标表面不一致，这种情况下会进行颜色格式的转换功能
- 背景表面可以与目标表面为同一表面。

qua_tde2_solid_draw

【说明】


向指定任务添加画点画线操作。实现在表面上画点、画线等功能。

QUA_S32 (*qua_tde2_solid_draw)(QUA_TDE_HANDLE handle, qua_tde2_surface_t *fore_ground, qua_tde2_rect_t *fore_ground_rect, qua_tde2_surface_t *dst, qua_tde2_rect_t *dst_rect, qua_tde2_fillcolor_t *fill_color, qua_tde2_opt_t *opt)

参数名称	输入/输出	描述	返回值
handle	输入	TDE 任务句柄	成功：QUA_SUCCESS； 失败：其它值
fore_ground	输入	前景表面	
fore_ground_rect	输入	前景表面操作区域	
dst	输出	目标表面	
dst_rect	输入	目标表面操作区域	
fill_color	输入	填充色参数	
opt	输入	附加参数	

【注意】

- 当前景表面为 NULL 且操作结构 opt 为 NULL 时，该接口可实现单纯的色彩填充功能，即与 qua_tde2_quick_fill 实现的功能一样。
- 当前景表面不为 NULL 时，操作属性 opt 一定不能为 NULL，可实现前景表面的指定区域与填充色做 alpha 叠加等操作，结果输出到目的表面的指定区域。
- 该接口绘制矩形或者水平/垂直直线的方法是通过设置填充矩形的宽/高来完成。如：垂直直线就是绘制宽度为 1 像素的矩形。

	QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
		版本 Rev.	1.1.0

qua_tde2_compress

【说明】

向指定任务添加数据压缩操作。

QUA_S32 (*qua_tde2_compress)(QUA_TDE_HANDLE handle, qua_tde2_surface_t *src, qua_tde2_surface_t *dst)			
参数名称	输入/输出	描述	返回值
handle	输入	TDE 任务句柄	成功: QUA_SUCCESS; 失败: 其它值
src	输入	源表面	
dst	输出	目标表面	

【注意】

- 此接口只添加单纯的压缩数据操作，所以源表面必须与目标表面尺寸一样。
- 缺省了操作区域相关参数，操作区域是整幅表面。

qua_tde2_decompress

【说明】

向指定任务添加数据解压缩操作。

QUA_S32 (*qua_tde2_decompress)(QUA_TDE_HANDLE handle, qua_tde2_surface_t *src, qua_tde2_surface_t *dst)			
参数名称	输入/输出	描述	返回值
handle	输入	TDE 任务句柄	成功: QUA_SUCCESS; 失败: 其它值
src	输入	源表面	
dst	输出	目标表面	

【注意】

- 此接口只添加单纯的压缩数据操作，所以源表面必须与目标表面尺寸一样。
- 缺省了操作区域相关参数，操作区域是整幅表面。

3.7.4 VPPU 设备函数接口

init


【说明】

VPU 模块初始化

QUA_S32 (*init)();			
参数名称	输入/输出	描述	返回值
	无	VPU 初始化	成功: QUA_SUCCESS; 失败: 其它值

【注意】

qm10xh 支持该接口，其他芯片平台不支持。

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

exit

【说明】

VPU 模块退出。

QUA_S32 (*exit());			
参数名称	输入/输出	描述	返回值
	无	VPU 退出	成功: QUA_SUCCESS; 失败: 其它值

【注意】

qm10xh 支持该接口，其他芯片平台不支持。

set_vi_attr

【说明】

设置视频输入属性

QUA_S32 (*set_vi_attr)(QUA_GROUP group_id, qua_vpu_size_t *size);			
参数名称	输入/输出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
size	输入	视频输入幅面	

【注意】

qm10xh 支持该接口，其他芯片平台不支持。

get_vi_attr

【说明】

获取视频输入属性

QUA_S32 (*get_vi_attr)(QUA_GROUP group_id, qua_vpu_size_t *size);			
参数名称	输入/输出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
size	输入	视频输入幅面	

【注意】

qm10xh 支持该接口，其他芯片平台不支持。

create_grp

【说明】

创建视频输入

QUA_S32 (*create_grp)(QUA_GROUP group_id, qua_vpss_grp_attr_t *grp_attr);			
参数	输入/输出	描述	返回值



名称	出		
group_id	输入	GROUP 号	成功: QUA_SUCCESS;
grp_attr	输入	视频输入幅面相关信息	失败: 其它值

destroy_grp

【说明】

销毁视频输入

QUA_S32 (*destroy_grp)(QUA_GROUP group_id);

参 数 名称	输入/输 出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值

enable

【说明】

使能视频处理模块

QUA_S32 (*enable)(QUA_GROUP group_id, qua_vpu_vi_mode_t vi_mode);

参 数 名称	输入/输 出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
vi_mode	输入	视频处理模块工作模式	

disable

【说明】

禁止视频处理模块

QUA_S32 (*disable)(QUA_GROUP group_id);

参 数 名称	输入/输 出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值

send_frame

【说明】

用户向视频处理模块发送数据

QUA_S32 (*send_frame)(QUA_GROUP group_id, qua_video_frame_info_t *frame,
QUA_S32 timeout);

参 数	输入/输	描述	返回值
-----	------	----	-----



名称	出		
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
frame	输入	用户图像信息	
timeout	输入	设置超时时间	

【注意】

mc6870, mc331x 支持该接口。

send_user_pic**【说明】**

提交用户图像给视频处理模块进行处理, 仅在处于非直通模式下有效。

QUA_S32 (*send_user_pic)(QUA_GROUP group_id, qua_vpu_user_pic_t *pic);

参数名称	输入/输出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
pic	输入	用户图像信息	

【注意】

qm10xh 支持该接口, 其他芯片平台不支持该接口。

3.7.5 VPU 通道函数接口**get_chn_frame_adv****【说明】**

获取视频处理模块输出通道的图像数据

QUA_S32 (*get_chn_frame_adv)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_stream_adv_t *stream_adv, QUA_U32 timeout_ms);

参数名称	输入/输出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
chn_id	输入	通道号	
stream_adv	输入	通道的图像数据信息	
timeout_ms	输入	接口阻塞超时时间	

get_chn_frame_adv_norpt**【说明】**

获取视频处理模块输出通道的图像数据

QUA_S32 (*get_chn_frame_adv_norpt)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_stream_adv_t *stream_adv, QUA_U32 timeout_ms);



参数名称	输入/输出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
chn_id	输入	通道号	
stream_adv	输入	通道的图像数据信息	
timeout_ms	输入	接口阻塞超时时间	

【注意】

- timeout_ms 为 0 时不返回已经已经被获取过的帧
- timeout_ms 不为 0 时若前一帧未被取走则直接返回前一帧，否则进入等待
- 此接口不支持并发获取相同通道数据，否则多线程间会相互竞争资源

set_chn_attr**【说明】**

设置视频处理模块通道属性

```
QUA_S32 (*set_chn_attr)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_chn_config_t *chn_cfg);
```

参数名称	输入/输出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
chn_id	输入	通道号	
chn_cfg	输入	通道属性信息	

【注意】

在设置通道属性后，包含帧率控制，视频冻结等通道配置会失效，需要重新设置。

get_chn_attr**【说明】**

获取视频处理模块通道属性

```
QUA_S32 (*get_chn_attr)(QUA_GROUP group_id, QUA_U32 chn_id, qua_vpu_chn_config_t *chn_cfg);
```

参数名称	输入/输出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
chn_id	输入	通道号	
chn_cfg	输入	通道属性信息	

set_ext_chn_attr**【说明】**

设置视频处理模块扩展通道属性

```
QUA_S32 (*set_ext_chn_attr)(QUA_GROUP group_id, QUA_U32 chn_id,
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_vpu_ext_chn_config_t *ext_chn_cfg);

参数名称	输入/输出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
chn_id	输入	通道号	
ext_chn_cfg	输入	扩展通道属性	

open_chn

【说明】

开启通道使能

QUA_S32 (*open_chn)(QUA_GROUP group_id, QUA_U32 chn_id);

参数名称	输入/输出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
chn_id	输入	通道号	


close_chn

【说明】

关闭通道使能

QUA_S32 (*close_chn)(QUA_GROUP group_id, QUA_U32 chn_id);

参数名称	输入/输出	描述	返回值
group_id	输入	GROUP 号	成功: QUA_SUCCESS; 失败: 其它值
chn_id	输入	通道号	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

3.8 AV 扩展模块接口

AV 扩展模块提供芯片未覆盖到的音频、视频格式的编码和解码，以及媒体文件的封装和解析功能，接口的头文件路径：`include/avext/qua_mm_avevt.h`。

3.8.1 数据结构

qua_packet_t

【说明】

音视频数据包定义。

【定义】

```
typedef struct qua_packet {
    uint8_t* data;
    size_t offset;
    int size;
    int capacity;
    uint64_t pts;
    uint64_t dts;
    uint64_t duration;
    uint64_t baseTime;
    int flags;
    size_t streamIndex;
} qua_packet_t;
```

【描述】


成员名称	描述
data	数据指针地址
offset	指针地址
size	数据包长度
capacity	实际数据长度
pts	显示时间戳
dts	解码时间戳
duration	数据包之间间隔
baseTime	数据包基时间
flags	数据包标志
streamIndex	数据包的流索引

qua_audio_config_t

【说明】

音频属性配置。

【定义】

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

```
typedef struct qua_audio_config {
    int in_nb_channels;
    int in_sample_rate;
    int in_sample_fmt;
    int out_nb_channels;
    int out_sample_rate;
    int out_sample_fmt;
    int bit_rate;
    int bits_per_coded_sample;
    int block_align;
    int profile;
    int level;
    uint8_t extra_data[QUA_EXTRADATA_MAXSIZE];
    int extra_data_size;
    qua_aac_transtype_t trans_type;
} qua_audio_config_t;
```

【描述】

成员名称	描述
in_nb_channels	输入通道数
in_sample_rate	输入采样率
in_sample_fmt	输入采样格式，如下： <pre>typedef enum qua_sample_format { QUA_SAMPLE_FMT_NONE = -1, QUA_SAMPLE_FMT_U8, ///< unsigned 8 bits QUA_SAMPLE_FMT_S16, ///< signed 16 bits QUA_SAMPLE_FMT_S32, ///< signed 32 bits QUA_SAMPLE_FMT_FLT, ///< float QUA_SAMPLE_FMT_DBL, ///< double QUA_SAMPLE_FMT_U8P, ///< unsigned 8 bits, planar QUA_SAMPLE_FMT_S16P, ///< signed 16 bits, planar QUA_SAMPLE_FMT_S32P, ///< signed 32 bits, planar QUA_SAMPLE_FMT_FLTP, ///< float, planar QUA_SAMPLE_FMT_DBLP, ///< double, planar QUA_SAMPLE_FMT_S64, ///< signed 64 bits QUA_SAMPLE_FMT_S64P, ///< signed 64 bits, planar</pre>



	<p>QUA_SAMPLE_FMT_NB //< Number of sample formats. DO NOT USE if linking dynamically } qua_sample_format_t;</p>
out_nb_channels	输出通道数
out_sample_rate	输出采样率
out_sample_fmt	输出采样格式
bit_rate	码率
bits_per_coded_sample	每一个采样占用 bit 数
block_align	每一个音频块占用字节数
profile	编解码特性 profile
level	编解码特性 level
extra_data	编解码元数据参数
extra_data_size	编解码元数据参数大小
trans_type	<p>AAC 编码的输出格式，如下：</p> <pre>typedef enum qua_aac_transtype { QUA_TT_UNKNOWN = -1, QUA_TT_MP4_RAW = 0, QUA_TT_MP4_ADIF = 1, QUA_TT_MP4_ADTS = 2, QUA_TT_MP4_LATM_MCP1 = 6, QUA_TT_MP4_LATM_MCP0 = 7, QUA_TT_MP4_LOAS = 10, QUA_TT_DRM = 12, } qua_aac_transtype_t;</pre>

qua_audio_frame_ext_t

【说明】

扩展音频帧属性。

【定义】

```
typedef struct qua_audio_frame_ext {
    uint8_t *data;
    int offset;
    int size;
    int capacity;
    int nb_samples;
    int sample_fmt;
    int sample_rate;
    int nb_channels;
    int flags;
    uint64_t pts;
} qua_audio_frame_ext_t;
```

【描述】

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

成员名称	描述
data	数据指针
offset	数据指针偏移
size	数据帧大小
capacity	数据帧容量
nb_samples	采样数
sample_fmt	采样格式
sample_rate	采样率
nb_channels	通道数
flags	标记
pts	显示时间戳

qua_ext_codec_chn_attr

【说明】

创建软件解码通道的属性

【定义】

```
typedef struct qua_ext_codec_chn_attr {
    QUA_CONST_CHAR* mimetype;
    QUA_VOID* mime_extra;
    QUA_VOID* param;
} qua_ext_codec_chn_attr_t;
```

【描述】

成员名称	描述
mimetype	音视频的 mimetype 类型
mime_extra	mimetype 的额外信息
param	音视频解码的解码参数，参照 qua_audio_config_t 等

qua_mm_ext_module_t

【说明】


获取 AV 扩展的 module

【定义】

```
typedef struct qua_mm_ext_module_t {
    struct qua_mm_module_t parent;
    qua_metaformat_t* (*create_meta_format)(QUA_CHAR* name);
    QUA_S32 (*delete_meta_format)(qua_metaformat_t* meta_format);
} qua_mm_ext_module_t;
```

【描述】

成员名称	描述
parent	qua_mm_module_t
create_meta_format	创建 metaformat，利于 AVEXT 之间传输参数
delete_meta_format	销毁 metaformat

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

所涉及的函数指针描述如下：

create_meta_format:			
参数名称	输入/输出	描述	返回值
name	输入	创建 meta_format 的相关名字	成功：meta_format 指针； 失败：NULL
delete_meta_format:			
参数名称	输入 / 输出	描述	返回值
meta_format	输入	销毁 qua_metaformat_t 指针	成功：QUA_SUCCESS； 失败：QUA_FAILURE

qua_ext_muxer_chn_attr

【说明】

创建封装通道的属性

【定义】

```
typedef struct qua_ext_muxer_chn_attr {
    qua_output_format_t format;
    QUA_CHAR file[1024];
} qua_ext_muxer_chn_attr_t;
```

【描述】

成员名称	描述
format	封装格式，目前支持 MP4，如下： <pre>typedef enum qua_output_format { QUA_OUTPUT_FORMAT_DEFAULT = 0, QUA_OUTPUT_FORMAT_THREE_GPP = 1, QUA_OUTPUT_FORMAT_MPEG_4 = 2, QUA_OUTPUT_FORMAT_AAC_ADTS = 3, QUA_OUTPUT_FORMAT_MPEG2TS = 4, QUA_OUTPUT_FORMAT_MP3 = 5, QUA_OUTPUT_FORMAT_MAX, } qua_output_format_t;</pre>
file	保存文件含文件名的全路径


qua_mm_muxer_device

【说明】

创建封装的设备

【定义】

```
typedef struct qua_mm_muxer_device {
    struct qua_mm_device_t parent;
} qua_mm_muxer_device_t;
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

【描述】

成员名称	描述
parent	创建 device 的结构体

qua_mm_muxer_channel

【说明】

创建封装的通道

【定义】

```
typedef struct qua_mm_muxer_channel {
    struct qua_mm_channel_t parent;
    QUA_VOID* muxer;
    QUA_S32 (*track_meta_data)(QUA_VOID* mux, qua_mediasream_type_t stream_type,
    qua_metaformat_t* track_meta);
    QUA_S32 (*start)(QUA_VOID* mux);
    QUA_S32 (*write_packet)(QUA_VOID* mux, qua_mediasream_type_t stream_type, qua_packet_t*
    packet);
    QUA_S32 (*set_param)(QUA_VOID* mux, QUA_S32 key, QUA_VOID_PTR param);
    QUA_S32 (*get_param)(QUA_VOID* mux, QUA_S32 key, QUA_VOID_PTR param);
    QUA_S32 (*stop)(QUA_VOID* mux);
} qua_mm_muxer_channel_t;
```

【描述】

成员名称	描述
parent	创建 channel 的结构体
muxer	封装实例的指针
track_meta_data	写入封装格式的音频和视频的参数
start	开始启动
write_packet	写入音频或者视频的数据
set_param	设置封装通道的参数
get_param	获取封装通道的参数
stop	停止写入

以上表格里的函数指针成员放在 3.8.2 章节中做具体说明。

qua_mm_extractor_device_t

【说明】

创建封装的设备

【定义】

```
typedef struct qua_mm_extractor_device_t {
    struct qua_mm_device_t parent;
} qua_mm_extractor_device_t;
```

【描述】

成员名称	描述
------	----



parent

创建 device 的结构体

qua_mm_extractor_channel

【说明】

解封装的通道

【定义】

```
typedef struct qua_mm_extractor_channel {
    struct qua_mm_channel_t parent;
    QUA_VOID* extractor;
    QUA_SIZE_T (*track_count)(QUA_VOID* ex);
    QUA_S32 (*get_audio_track_index)(QUA_VOID* ex);
    QUA_S32 (*get_video_track_index)(QUA_VOID* ex);
    QUA_S32 (*meta_data)(QUA_VOID* ex, qua_metaformat_t** meta);
    QUA_S32 (*track_meta_data)(QUA_VOID* ex, QUA_SIZE_T track_index, qua_metaformat_t**
track_meta);
    QUA_SSIZE_T (*read_sample_data)(QUA_VOID* ex, QUA_SIZE_T track_index, qua_packet_t*
packet);
    QUA_S32 (*seek)(QUA_VOID* ex, QUA_U64 time_us, seek_mode_t mode);
    QUA_BOOL (*advance)(QUA_VOID* ex);
} qua_mm_extractor_channel_t;
```

【描述】

成员名称	描述
parent	创建 channel 的结构体
extractor	解封装实例的指针
track_count	获取数据轨道的数量
get_audio_track_index	获取音频流的索引
get_video_track_index	获取视频流的索引
meta_data	获取文件的 meta 信息
track_meta_data	获取视音频流的 meta 信息
read_sample_data	读取音视频流一帧数据
seek	快速查找
advance	快速查找下一个帧数据（暂未使用）

以上表格里的函数指针成员放在 3.8.3 章节中做具体说明。


qua_mm_decoder_device_t

【说明】

创建软件解码的设备

【定义】

```
typedef struct qua_mm_decoder_device_t {
    struct qua_mm_device_t parent;
} qua_mm_decoder_device_t;
```

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

【描述】

成员名称	描述
parent	创建 device 的结构体

qua_mm_decoder_channel

【说明】

创建软件解码通

【定义】

```
typedef struct qua_mm_decoder_channel {
    struct qua_mm_channel_t parent;
    QUA_VOID* decoder;
    QUA_S32 (*start)(QUA_VOID* dec);
    QUA_S32 (*send_packet)(QUA_VOID* dec, qua_packet_t* packet);
    QUA_S32 (*receive_frame)(QUA_VOID* dec, QUA_VOID* frame);
    QUA_S32 (*query)(QUA_VOID* dec);
    QUA_S32 (*flush)(QUA_VOID* dec);
    QUA_S32 (*stop)(QUA_VOID* dec);
} qua_mm_decoder_channel_t;
```

【描述】

成员名称	描述
parent	创建 channel 的结构体
decoder	解码实例的指针
start	启动解码
send_packet	送未解码的数据包
receive_frame	获取解码之后的数据
query	查询（暂未使用）
flush	清空解码实例的缓存数据
stop	解码结束

以上表格里的函数指针成员放在 3.8.4 章节中做具体说明。

qua_mm_encoder_device

【说明】


创建软件编码的设备

【定义】

```
typedef struct qua_mm_encoder_device {
    struct qua_mm_device_t parent;
} qua_mm_encoder_device_t;
```

【描述】

成员名称	描述
parent	创建 device 的结构体

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

qua_mm_encoder_channel

【说明】

创建软件编码通道

【定义】

```
typedef struct qua_mm_encoder_channel {
    struct qua_mm_channel_t parent;
    QUA_VOID* encoder;
    QUA_S32 (*start)(QUA_VOID* enc);
    QUA_S32 (*send_frame)(QUA_VOID* enc, QUA_VOID* frame);
    QUA_S32 (*receive_packet)(QUA_VOID* enc, qua_packet_t* packet);
    QUA_S32 (*query)(QUA_VOID* enc);
    QUA_S32 (*flush)(QUA_VOID* enc);
    QUA_S32 (*stop)(QUA_VOID* enc);
} qua_mm_encoder_channel_t;
```

【描述】

成员名称	描述
parent	创建 channel 的结构体
encoder	编码实例的指针
start	启动编码
send_frame	发送音视频的 RAW 数据
receive_packet	获取编码后的数据包
query	查询（暂未使用）
flush	清空编码实例的缓存数据
stop	编码结束

以上表格里的函数指针成员放在 3.8.5 章节中做具体说明。


3.8.2 媒体封装函数接口

【描述】

将音频流和视频流封装成 MP4，目前支持视频格式 H264，音频格式 AAC。

track_meta_data

QUA_S32 (*track_meta_data)(QUA_VOID* mux, qua_mediastream_type_t stream_type, qua_metaformat_t* track_meta);			
参数名称	输入 / 输出	描述	返回值
mux	输入	封装实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE
stream_type	输入	音频或者视频类型	
track_meta	输入	音频或者视频的编码参数	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

start

QUA_S32 (*start)(QUA_VOID* mux);			
参数名称	输入 / 输出	描述	返回值
mux	输入	封装实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE

write_packet


QUA_S32 (*write_packet)(QUA_VOID* mux, qua_mediastream_type_t stream_type, qua_packet_t* packet);			
参数名称	输入 / 输出	描述	返回值
mux	输入	封装实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE
stream_type	输入	音频或者视频类型, 如下: typedef enum qua_mediastream_type { QUA_STREAM_VIDEO, QUA_STREAM_AUDIO, } qua_mediastream_type_t;	
packet	输入	音频或者视频的数据结构体	

set_param

QUA_S32 (*set_param)(QUA_VOID* mux, QUA_S32 key, QUA_VOID_PTR param);			
参数名称	输入 / 输出	描述	返回值
mux	输入	封装实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE
key	输入	设置参数的类型	
param	输入	设置参数的指针	

get_param

QUA_S32 (*get_param)(QUA_VOID* mux, QUA_S32 key, QUA_VOID_PTR param);			
参数名称	输入 / 输出	描述	返回值
mux	输入	封装实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE
key	输入	设置参数的类型	
param	输入	设置参数是指针	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

stop

QUA_S32 (*stop)(QUA_VOID* mux);			
参数名称	输入 / 输出	描述	返回值
mux	输入	封装实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE

3.8.3 媒体解封装函数接口

【描述】

将媒体文件分离成音频和视频流。

track_count

QUA_SIZE_T (*track_count)(QUA_VOID* ex);			
参数名称	输入 / 输出	描述	返回值
ex	输入	解封装实例的指针	视频流轨道的数量

get_audio_track_index

QUA_S32 (*get_audio_track_index)(QUA_VOID* ex);			
参数名称	输入 / 输出	描述	返回值
ex	输入	解封装实例的指针	音频轨道的索引值

get_video_track_index

QUA_S32 (*get_video_track_index)(QUA_VOID* ex);			
参数名称	输入 / 输出	描述	返回值
ex	输入	解封装实例的指针	视频轨道的索引值

meta_data

QUA_S32 (*meta_data)(QUA_VOID* ex, qua_metaformat_t** meta);			
参数名称	输入 / 输出	描述	返回值
ex	输入	解封装实例的指针	成功: QUA_SUCCESS



meta	输入	从解封装实例获取媒体文件信息	失败: QUA_FAILURE
------	----	----------------	-----------------

track_meta_data

QUA_S32 (*track_meta_data)(QUA_VOID* ex, QUA_SIZE_T track_index, qua_metaformat_t* track_meta);

参数名称	输入 / 输出	描述	返回值
ex	输入	解封装实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE
track_index	输入	音视频轨道的索引	
track_meta	输入	音视频轨道的 meta 信息	

read_sample_data


QUA_SSIZE_T (*read_sample_data)(QUA_VOID* ex, QUA_SIZE_T track_index, qua_packet_t* packet);

参数名称	输入 / 输出	描述	返回值
ex	输入	解封装实例的指针	成功: 数据长度 失败: -1
track_index	输入	音视频轨道的索引	
packet	输入	音视频轨道的数据包	

seek

QUA_S32 (*seek)(QUA_VOID* ex, QUA_U64 time_us, seek_mode_t mode);

参数名称	输入 / 输出	描述	返回值
ex	输入	解封装实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE
time_us	输入	快速查找的时间戳	
mode	输入	快速查找的类型, 如下: <pre>typedef enum seek_mode { SEEK_PREVIOUS_SYNC = 0, SEEK_NEXT_SYNC = 1, SEEK_CLOSEST_SYNC = 2, SEEK_CLOSEST = 3, SEEK_FRAME_INDEX = 4, NONBLOCKING = 5, } seek_mode_t;</pre>	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

advance

QUA_BOOL (*advance)(QUA_VOID* ex); (暂时未用)			
参数名称	输入 / 输出	描述	返回值
ex	输入	解封装实例的指针	成功: QUA_TRUE 失败: QUA_FALSE

3.8.4 媒体解码函数接口

【描述】

提供音频和视频的软件解码（目前只支持音频）

start


QUA_S32 (*start)(QUA_VOID* dec);			
参数名称	输入 / 输出	描述	返回值
dec	输入	解码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE

send_packet

QUA_S32 (*send_packet)(QUA_VOID* dec, qua_packet_t* packet);			
参数名称	输入 / 输出	描述	返回值
dec	输入	解码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE
packet	输入	未解码的音视频数据包	

receive_frame

QUA_S32 (*receive_frame)(QUA_VOID* dec, QUA_VOID* frame);			
参数名称	输入 / 输出	描述	返回值
dec	输入	解码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE
frame	输入	获取解码之后音视频数据，参照 qua_audio_frame_ext_t 等	

 QuaMM 多媒体接口使用说明	生效日期 Effective Date	2024.08.01
	版本 Rev.	1.1.0

query

QUA_S32 (*query)(QUA_VOID* dec); (暂时未用)			
参数名称	输入 / 输出	描述	返回值
dec	输入	解码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE

flush

QUA_S32 (*flush)(QUA_VOID* dec);			
参数名称	输入 / 输出	描述	返回值
dec	输入	解码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE

stop

QUA_S32 (*stop)(QUA_VOID* dec);			
参数名称	输入 / 输出	描述	返回值
dec	输入	解码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE

3.8.5 媒体编码函数接口

【描述】

提供音频和视频的软件编码（目前只支持音频）

start

QUA_S32 (*start)(QUA_VOID* enc);			
参数名称	输入 / 输出	描述	返回值
enc	输入	编码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE

send_frame

QUA_S32 (*send_frame)(QUA_VOID* enc, QUA_VOID* frame);			
--	--	--	--



参数名称	输入 / 输出	描述	返回值
enc	输入	编码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE
frame	输入	音视频源数据, 参照 qua_audio_frame_ext_t 等	

receive_packet

QUA_S32 (*receive_packet)(QUA_VOID* enc, qua_packet_t* packet);

参数名称	输入 / 输出	描述	返回值
enc	输入	编码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE
packet	输入	编码之后的数据包	

query

QUA_S32 (*query)(QUA_VOID* enc); (暂时未用)

参数名称	输入 / 输出	描述	返回值
enc	输入	编码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE

flush

QUA_S32 (*flush)(QUA_VOID* enc);

参数名称	输入 / 输出	描述	返回值
enc	输入	编码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE

stop

QUA_S32 (*stop)(QUA_VOID* enc);

参数名称	输入 / 输出	描述	返回值
enc	输入	编码实例的指针	成功: QUA_SUCCESS 失败: QUA_FAILURE