



旷明智能

QuaPlayer 接口使用说明

Confidential

文档编号:		版本号:	V1.1.1
日期:	2026/3/4	负责人:	多媒体 Team
文档类型:	接口文档		
版权声明:	<p>本文件系旷明智能科技（“QuaMing”）的原创成果，旷明智能科技依法享有该文件的全部版权。对本文件的全部或部分复制，务必事先取得旷明智能科技的书面许可，并向版权所有者予以明确确认。任何侵犯本公司版权等知识产权的行为，本公司均保留依法追究其法律责任的权利。</p> <p>在法律许可的范畴内，特此声明，使用前，请审慎阅读合同条款与条件以及相关说明，并严格遵循本文件中的各项说明。旷明智能科技对于不当行为所导致的后果（包括但不限于电压过高、超频或温度过高），不承担任何责任。</p> <p>旷明智能科技所提供的信息仅作参考或典型应用之用，本文件中的所有声明、信息及建议，均不构成任何明示或暗示的担保。旷明智能科技保留随时变更电路设计和/或规格的权利，且无需另行通知。</p> <p>客户应全权负责获取实施解决方案/产品可能所需的第三方许可，旷明智能科技不承担任何与第三方许可相关的许可费用或特许权使用费。对于任何由所需第三方许可证所涵盖的事项，旷明智能科技不承担任何保证、赔偿或其他义务。</p> <p>凡以任何方式直接或间接使用本文件资料者，均视为自愿接受本文件声明的约束。</p>		

版本	更新人	修订日期	修订内容
V1.0.0	多媒体Team	2024/4/19	初始版本
V1.1.0	多媒体Team	2026/01/23	增加设置fd source 接口 增加参数key描述
V1.1.1	多媒体Team	2026/03/04	增加decoder/vppo 接口描述

Quaming Confidential

目录

1 概述	2
1.1 背景	2
1.2 目的	2
2 总体描述	3
2.1 基本概念	3
2.2 目录介绍	4
3 接口描述	5
4 数据结构	15
5 播放流程	17

Quaming Confidential

1 概述

1.1 背景

旷明 XOS 平台作为一个领先的跨平台操作系统，凭借其高度的灵活性和适应性，广泛应用于各种嵌入式设备，覆盖了从智能家居、工业控制到便携式多媒体终端等多个领域。然而，由于底层芯片架构和操作系统环境的多样性，开发者在实现多媒体播放功能时，通常面临繁琐的适配工作，这不仅降低了开发效率，也增加了维护成本。为了解决这一难题，打造统一、稳定的多媒体播放引擎成为重要需求。

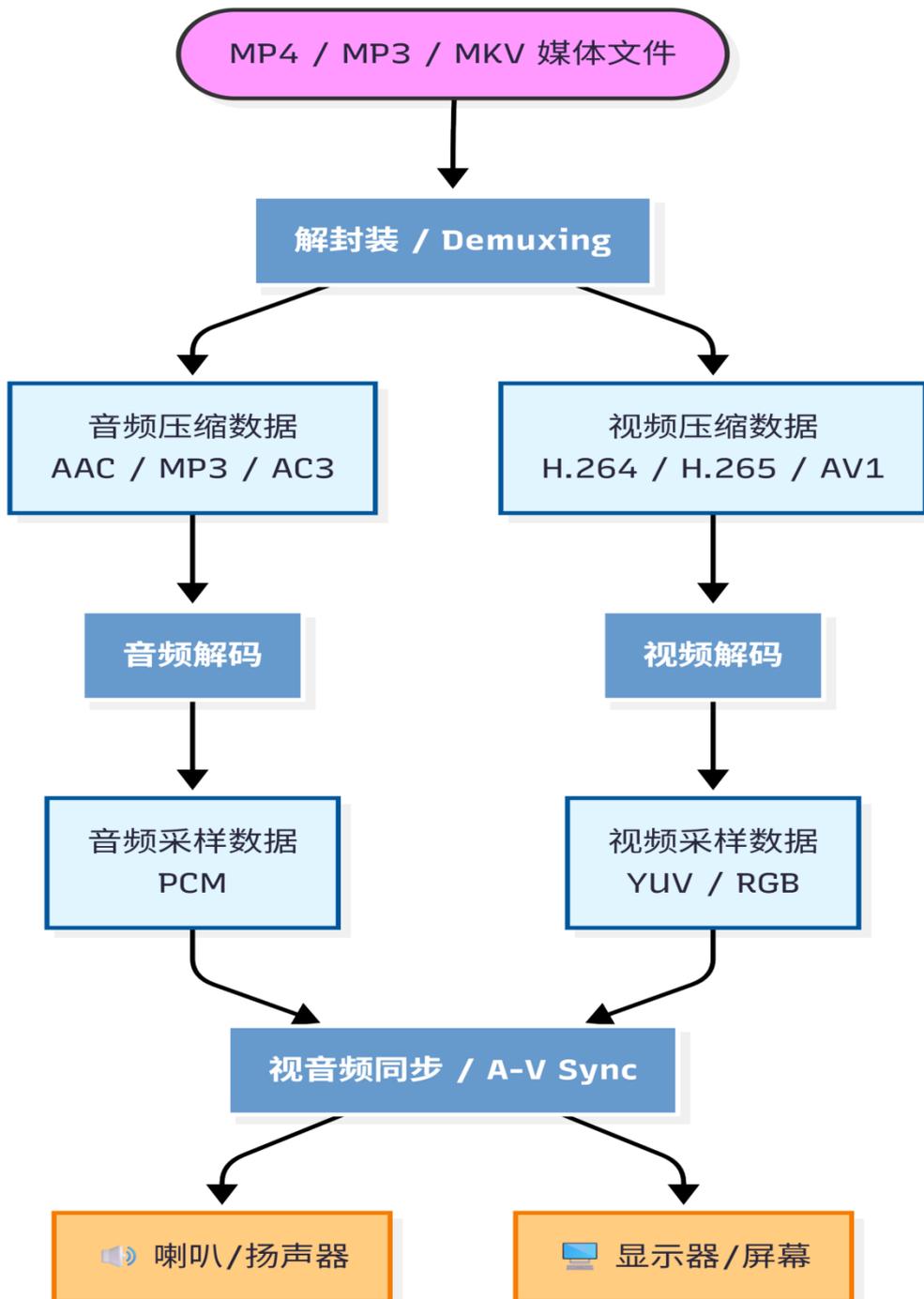
1.2 目的

QuaPlayer 是一个多媒体播放引擎，是在 QuaMM 多媒体接口的基础上开发，定义了一套统一的标准化多媒体播放器接口，旨在为开发者提供一套跨平台、易于使用的 API，隐藏底层实现的细节，使开发者能够更加专注于业务逻辑的实现、内容和体验的创新，而无需重复适配底层差异。这一举措不仅极大简化了多平台开发流程，也为终端用户带来了更加一致和优质的多媒体体验。

2 总体描述

2.1 基本概念

QuaPlayer 的工作流程大致分为初始化、参数设置，准备、播放控制和资源释放。线程负责数据流处理和渲染。数据流方面，音频数据通过环形缓冲区管理，视频数据通过帧队列管理实现帧率控制，并且实现生产者和消费者模型。关键特性包括多线程处理、硬件加速解码、音视频同步、动态参数调整以及事件回调机制。



2.2 目录介绍

对于旷明 XOS 工程，qua_mm_player 头文件接口目录位于 core/package/libnative/qlibquammapi/include 下；demo 样例代码在 core/package/libnative/qlibquammapi/sample 下面。

player/qua_mm_player.h	播放器实例接口
player/qua_mm_player_common.h	应用需要了解的类型和变量
qua_player_single_sample.c	样例 demo 代码
qua_player_dual_sample.c	
qua_player_nalu_sample.c	
qua_player_pipe_sample.c	
qua_player_audio_sample.c	

3 接口描述

提供基于 QuaMM 多媒体框架开发的常用和必要接口。

qua_mm_player_create

【说明】

创建播放器实例。

【定义】

QUA_VOID* qua_mm_player_create(QUA_CONST_CHAR* name)

【描述】

参数名称	输入/输出	描述	返回值
name	输入	播放器名称	成功：播放器实例句柄指针 失败：NULL

qua_mm_player_set_data_source

【说明】

设置播放器本地数据源。

【定义】

QUA_S32 qua_mm_player_set_data_source(QUA_VOID* player, QUA_CONST_CHAR* url)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功：QUA_SUCCESS; 失败：其它值
url	输入	媒体资源全路径	

qua_mm_player_set_fd_source

【说明】

设置播放器文件描述符

【定义】

QUA_S32 qua_mm_player_set_fd_source(QUA_VOID *player, QUA_S32 fd_type, QUA_S32 fd)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功：QUA_SUCCESS; 失败：其它值
fd_type	输入	fd 句柄类型，参考枚举值 qua_player_fd_type	
fd	输入	fd 文件描述符	

【备注】

目前仅支持 pipe 这种类型。

qua_mm_player_prepare

【说明】

准备播放器资源并初始化音视频解码器，显示和音频输出等核心组件。

【定义】

QUA_S32 qua_mm_player_prepare(QUA_VOID* player)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_mm_player_set_parameter

【说明】

设置播放器参数，在播放器过程中或者 prepare 之前设置。

【定义】

QUA_S32 qua_mm_player_set_parameter(QUA_VOID* player, QUA_S32 key, QUA_VOID* value)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功: QUA_SUCCESS; 失败: 其他值
key	输入	参数键值 (参照 qua_parameter_keys)	
value	输入	参数指针	

qua_mm_player_get_parameter

【说明】

获取播放器参数。

【定义】

QUA_S32 qua_mm_player_get_parameter(QUA_VOID* player, QUA_S32 key, QUA_VOID* value)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功: QUA_SUCCESS; 失败: 其他值
key	输入	参数键值 (参照 qua_parameter_keys)	
value	输入	参数指针	

qua_mm_player_start

【说明】

启动播放，必须要在 prepare 成功之后调用。

【定义】

QUA_S32 qua_mm_player_start(QUA_VOID* player)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_mm_player_pause

【说明】

暂停播放，必须要在 start 成功之后调用。

【定义】

QUA_S32 qua_mm_player_pause(QUA_VOID* player)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_mm_player_resume

【说明】

恢复播放，必须要在 pause 成功之后调用。

【定义】

QUA_S32 qua_mm_player_resume(QUA_VOID* player)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_mm_player_stop

【说明】

停止播放，停止播放后，可以调用 qua_mm_player_start 后继续播放。

【定义】

QUA_S32 qua_mm_player_stop(QUA_VOID* player)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_mm_player_reset

【说明】

重置播放器，会释放音视频编解码器，音频输出，显示等资源。

【定义】

QUA_S32 qua_mm_player_reset(QUA_VOID* player)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_mm_player_seek

【说明】

跳转到指定时间点，单位微秒。

【定义】

QUA_S32 qua_mm_player_seek(QUA_VOID* player, QUA_U64 timeUs)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功：QUA_SUCCESS； 失败：其他值
timeUs	输入	跳转时间（单位：微秒）	

【备注】

可精准 seek 到指定 timeUs

qua_mm_player_get_durationUs

【说明】

跳转到指定时间点，单位微秒。

【定义】

QUA_S64 qua_mm_player_get_durationUs(QUA_VOID* player)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	当前媒体文件的时长

qua_mm_player_get_file_meta

【说明】

获取媒体文件元数据。

【定义】

QUA_S32 qua_mm_player_get_file_meta(QUA_VOID* player, QUA_VOID** meta)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功：QUA_SUCCESS； 失败：其他值
meta	输入	媒体文件元数据信息	

qua_mm_player_get_current_position

【说明】

获取当前播放位置单位为微秒。

【定义】

QUA_S64 qua_mm_player_get_durationUs(QUA_VOID* player)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	当前媒体文件的播放时间点微秒

qua_mm_player_set_volume

【说明】

设置播放器音量。

【定义】

QUA_S32 qua_mm_player_set_volume(QUA_VOID* player, QUA_S32 volume)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功: QUA_SUCCESS; 失败: 其他值
volume	输入	音量值, 取值范围【0-100】	

qua_mm_player_set_callback

【说明】

注册播放器时间回调。

【定义】

QUA_S32 qua_mm_player_set_callback(QUA_VOID* player, CallBack cb, QUA_VOID* user)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功: QUA_SUCCESS; 失败: 其它值
cb	输入	回调函数指针	
user	输入	用户自定义数据	

qua_mm_player_destroy

【说明】

销毁播放器实例, 销毁所有的资源, 销毁后 player 指针变成野指针, 需要置空处理, 不可再次使用。

【定义】

QUA_S32 qua_mm_player_destroy(QUA_VOID* player)

【描述】

参数名称	输入/输出	描述	返回值
player	输入	播放器句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_decoder_create

【说明】

根据输入码流的 MIME 类型创建一个通用解码器实例, 内部根据 mimetype 选择软解或硬解, 用于后续的 prepare/start/queue/dequeue 等操作。

【定义】

QUA_VOID *qua_decoder_create(QUA_CONST_CHAR *mimetype, QUA_VOID *mime_extra);

【描述】

参数名称	输入/输出	描述	返回值
mimetype	输入	输入码流的 MIME 类型字符串, 例如 “video/avc”、	成功: 返回解码器句柄; 失败: NULL

		“video/hevc”等	
mime_extra	输入	额外的编解码配置信息、可为 NULL	

qua_decoder_prepare

【说明】

按照传入的解码参数对解码器进行配置和资源准备。必须在 qua_decoder_start 之前调用。对于音频通常传入 qua_audio_config_t，对于视频通常传入 qua_video_config_t。

【定义】

QUA_S32 qua_decoder_prepare(QUA_VOID *dec, QUA_VOID *param)

【描述】

参数名称	输入/输出	描述	返回值
dec	输入	qua_decoder_create 返回的解码器句柄	成功: QUA_SUCCESS; 失败: 其他值
param	输入	解码配置参数指针,类型依具体流类型而定: 音频: qua_audio_config_t 视频: qua_video_config_t	

qua_decoder_start

【说明】

启动解码器，进入工作状态。调用后即可通过 qua_decoder_queue_buffer 输送压缩数据，通过 qua_decoder_dequeue_buffer 获取解码后的帧。

【定义】

QUA_S32 qua_decoder_start(QUA_VOID *dec)

【描述】

参数名称	输入/输出	描述	返回值
dec	输入	解码器句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_decoder_stop

【说明】

停止解码器接收和输出数据，一般在停止播放或切换媒体源时调用

【定义】

QUA_S32 qua_decoder_stop(QUA_VOID *dec);

【描述】

参数名称	输入/输出	描述	返回值
dec	输入	解码器句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_decoder_flush

【说明】

重置解码器内部状态并清空缓冲数据

【定义】

QUA_S32 qua_decoder_flush(QUA_VOID *dec);

【描述】

参数名称	输入/输出	描述	返回值
dec	输入	解码器句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_decoder_set_param

【说明】

运行过程中动态设置解码相关参数

【定义】

QUA_S32 qua_decoder_set_param(QUA_VOID *dec, QUA_S32 key, QUA_VOID *value);

【描述】

参数名称	输入/输出	描述	返回值
dec	输入	解码器句柄	成功: QUA_SUCCESS; 失败: 其他值
key	输入	参数键值, 含义由实现定义	
value	输入	参数值指针, 对应key的类型实现定义	

qua_decoder_get_param

【说明】

查询解码器当前的参数或状态信息

【定义】

QUA_S32 qua_decoder_get_param(QUA_VOID *dec, QUA_S32 key, QUA_VOID *value);

【描述】

参数名称	输入/输出	描述	返回值
dec	输入	解码器句柄	成功: QUA_SUCCESS; 失败: 其他值
key	输入	参数键值	
value	输出	输出参数缓存指针	

qua_decoder_queue_buffer

【说明】

向解码器输入一帧或一段压缩码流数据

【定义】

QUA_S32 qua_decoder_queue_buffer(QUA_VOID *dec, qua_packet_t *packet)

【描述】

参数名称	输入/输出	描述	返回值
dec	输入	解码器句柄	成功: QUA_SUCCESS; 失败: 其他值
packet	输入	输入数据包, 包含数据指针、长度、PTS、是否 EOF 等信息	

【备注】

QUA_EAGAIN 内部输入缓冲区满, 需要先调用 dequeue_buffer 把输出取走, 下次重送这帧数据否则会造成参考不连续导致花屏

qua_decoder_dequeue_buffer

【说明】

从解码器取出一帧解码后的数据

【定义】

QUA_S32 qua_decoder_dequeue_buffer(QUA_VOID *dec, QUA_VOID *frame)

【描述】

参数名称	输入/输出	描述	返回值
dec	输入	解码器句柄	成功: QUA_SUCCESS; 失败: 其他值
frame	输出	输出帧结构体指针	

【备注】

QUA_EAGAIN 内部输出缓冲区空, 当前无解码帧, 需要送入更多输入数据。

qua_decoder_release_buffer

【说明】

释放由解码器内部持有的输出资源

【定义】

QUA_S32 qua_decoder_release_buffer(QUA_VOID *dec, QUA_VOID *frame)

【描述】

参数名称	输入/输出	描述	返回值
dec	输入	解码器句柄	成功: QUA_SUCCESS; 失败: 其他值
frame	输入	需要释放的帧结构体指针, 对应 dequeue_buffer 得到的 `frame`。	

qua_decoder_decode_frame

【说明】

同步方式的解码接口: 将一帧压缩数据解码成一帧输出数据

【定义】

QUA_S32 qua_decoder_decode_frame(QUA_VOID *dec, qua_packet_t *packet, QUA_VOID *frame)

【描述】

参数名称	输入/输出	描述	返回值
dec	输入	解码器句柄	成功: QUA_SUCCESS; 失败: 其他值
packet	输入	输入压缩数据包	
frame	输出	输出帧结构体指针	

【备注】

目前仅支持 QM10XV JPEGD

qua_decoder_destroy

【说明】

销毁解码器实例, 释放内部占用的所有资源, 调用前应确保已停止使用该解码器

【定义】

QUA_S32 qua_decoder_destroy(QUA_VOID *dec)

【描述】

参数名称	输入/输出	描述	返回值
dec	输入	解码器句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_vpvo_init

【说明】

根据显示 ID 初始化 VPPO 实例，创建视频输出通道。需在 `prepare` 之前调用，并可通过 `set_param` 配置显示区域、通道矩形、帧率等参数

【定义】

QUA_VOID* qua_vpvo_init(QUA_CONST_CHAR *display_id)

【描述】

参数名称	输入/输出	描述	返回值
display_id	输入	显示设备 ID 字符串，用于指定输出目标显示，默认单屏为 “id:display0” 依此类推	成功: 返回 VPPO 句柄; 失败: NULL

qua_vpvo_set_param

【说明】

设置 VPPO 运行参数，需在 `prepare` 之前调用

【定义】

QUA_S32 qua_vpvo_set_param(QUA_VOID *vpvo, QUA_S32 key, QUA_VOID *value);

【描述】

参数名称	输入/输出	描述	返回值
vpvo	输入	VPPO 句柄	成功: QUA_SUCCESS; 失败: 其他值
key	输入	参数键值	
value	输入	参数值指针	

qua_vpvo_get_param

【说明】

查询 VPPO 当前参数或状态

【定义】

QUA_S32 qua_vpvo_get_param(QUA_VOID *vpvo, QUA_S32 key, QUA_VOID *value);

【描述】

参数名称	输入/输出	描述	返回值
vpvo	输入	VPPO 句柄	成功: QUA_SUCCESS; 失败: 其他值
key	输入	参数键值	
value	输出	输出参数缓存指针	

qua_vpvo_prepare

【说明】

按已设置的参数准备 VPPO 资源，必须在 start 之前调用

【定义】

QUA_S32 qua_vppo_prepare(QUA_VOID *vppo)

【描述】

参数名称	输入/输出	描述	返回值
vppo	输入	VPPO 句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_vppo_start

【说明】

启动 VPPO，进入可接收帧状态。调用后即可通过 `qua_vppo_send_frame` 送入视频帧。

【定义】

QUA_S32 qua_vppo_start(QUA_VOID *vppo)

【描述】

参数名称	输入/输出	描述	返回值
vppo	输入	VPPO 句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_vppo_send_frame

【说明】

将解码后的视频帧送入 VPPO 进行显示

【定义】

QUA_S32 qua_vppo_send_frame(QUA_VOID *vppo, qua_video_frame_info_t *frame_info)

【描述】

参数名称	输入/输出	描述	返回值
vppo	输入	VPPO 句柄	成功: QUA_SUCCESS; 失败: 其他值
frame_info	输入	视频帧信息，包含图像数据、尺寸、PTS 等	

qua_vppo_stop

【说明】

停止 VPPO 接收和显示帧

【定义】

QUA_S32 qua_vppo_stop(QUA_VOID *vppo)

【描述】

参数名称	输入/输出	描述	返回值
vppo	输入	VPPO 句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_vppo_reset

【说明】

重置 VPPO 内部状态，清空缓冲。一般在 stop 之后

【定义】

QUA_S32 qua_vppo_reset(QUA_VOID *vppo)

【描述】

参数名称	输入/输出	描述	返回值
vppo	输入	VPPO 句柄	成功: QUA_SUCCESS; 失败: 其他值

qua_vppo_deinit

【说明】

销毁 VPPO 实例, 释放资源

【定义】

QUA_S32 qua_vppo_deinit(QUA_VOID *vppo)

【描述】

参数名称	输入/输出	描述	返回值
vppo	输入	VPPO 句柄	成功: QUA_SUCCESS; 失败: 其他值

4 数据结构

enum qua_paramter_keys

【说明】

参数键值定义

【定义】

```
enum qua_parameter_keys {
    KEY_PARAMETER_VO_CHN_RECT,
    KEY_PARAMETER_VO_DISPLAY_RECT,
    KEY_PARAMETER_VO_IMG_SIZE,
    KEY_PARAMETER_VO_ROTATE,
    KEY_PARAMETER_FILTER_CHN_ATTR,
    KEY_PARAMETER_SPEED_RATE,
    KEY_PARAMETER_SET_CURRENT_POSITION_CALLBACK,
    KEY_PARAMETER_VIDEO_SIZE,
    KEY_PARAMETER_VO_DISPLAY_ID,
    KEY_PARAMETER_VO_FRAMERATE,
    KEY_PARAMETER_VIDEO_FRAMERATE,
    KEY_PARAMETER_DUAL_DISPLAY_MODE,
    KEY_PARAMETER_VO_CHN_CROP,
    KEY_PARAMETER_AO_TRACK_TYPE,
    KEY_PARAMETER_VIDEO_BUFFER_CONFIG,
    KEY_PARAMETER_VO_VIDEO_LAYER,
    KEY_PARAMETER_VO_DISPLAY_MODE,
    KEY_PARAMETER_VO_CHN_DISPLAY_RECT
};
```

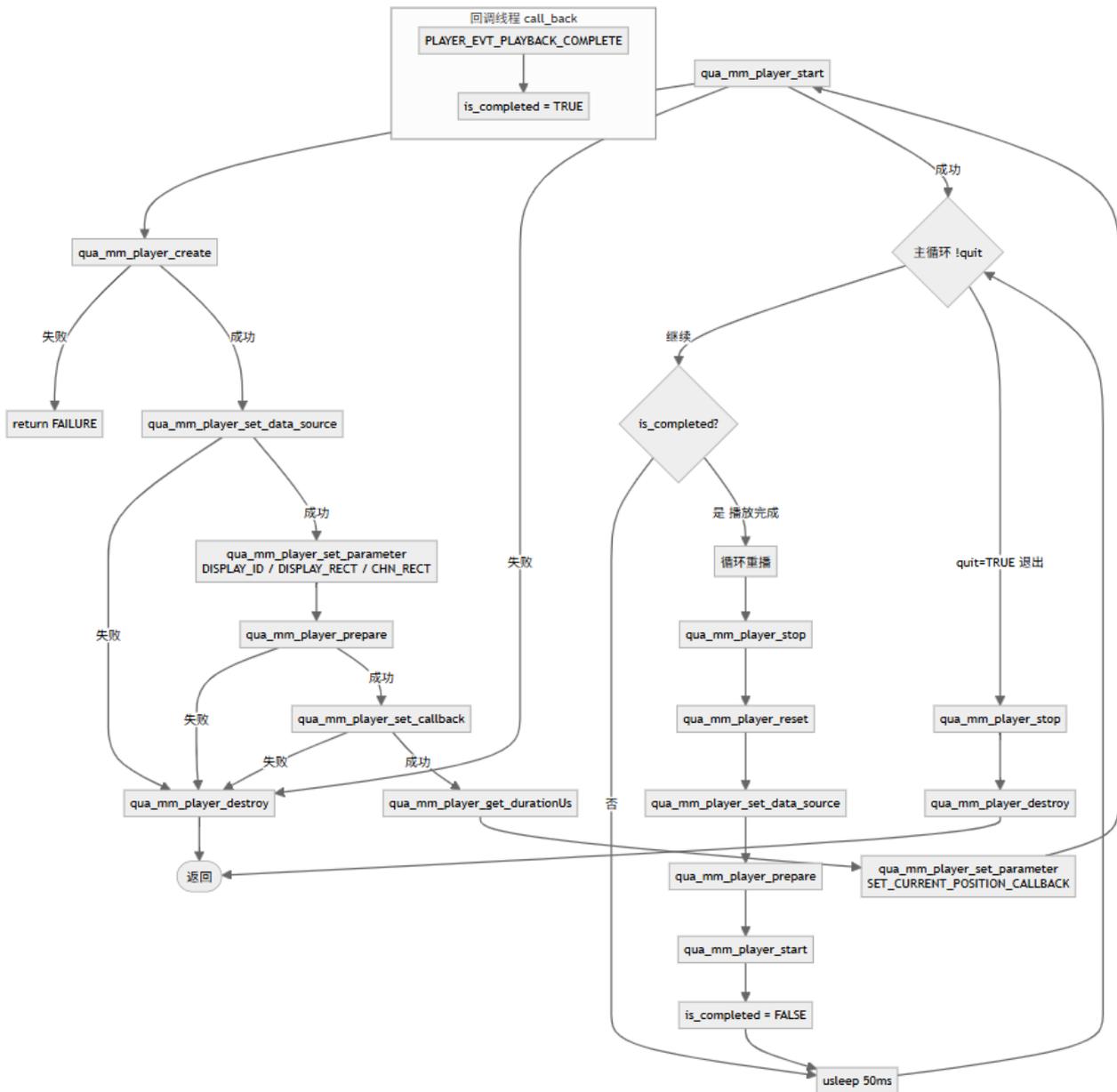
【描述】

参数名称	描述
KEY_PARAMETER_VO_CHN_RECT	单通道区域大小, 通道显示区域不可超过整个显示区域大小, 参数数据结构请参考 qua_rect_t
KEY_PARAMETER_VO_DISPLAY_RECT	显示区域大小, 参数数据结构请参考 qua_rect_t

KEY_PARAMETER_VO_ROTATE	设置视频层旋转角度,参数数据结构参考 qua_rotate_t
KEY_PARAMETER_SET_CURRENT_POSITION_CALLBACK	位置信息上报回调
KEY_PARAMETER_VIDEO_SIZE	获取视频的分辨率, 参数数据结构参考 qua_size_t
KEY_PARAMETER_VO_DISPLAY_ID	设置屏幕显示的 id, 如"id:display0"
KEY_PARAMETER_VO_VIDEO_LAYER	设置视频层 id
KEY_PARAMETER_VO_CHN_DISPLAY_RECT	多通道区域大小, 多通道共享同一个视频层时设置

Quaming Confidential

5 播放流程



注意：

1. QuaPlayer接口支持多实例，但不是无限制多实例，根据平台的内存和硬件能力情况来确定；
2. 视频支持显示区域设置，如果支持多屏，需要设置显示屏的ID，默认是显示在主屏幕，需要在调用prepare之前设置；
3. 视频播放支持90°，180°，270° 旋转；需要在prepare之前设置旋转角度；
4. 目前支持本地视频播放，支持mp3文件流式播放，支持https网络流

ABOUT QUAMING

The majority of the R&D team of the company come from well-known semiconductor companies, such as UNISOC, ASR Microelectronics, and JLQ Technology. The team masters the core technologies of chip design, such as the AI algorithm library, the self-developed AI operating system, the high-performance multimedia library, and the multimedia chip IP. It possesses leading chips and is equipped with its self-developed AI operating system XOS, meeting the demands of intelligence, display and vision. It has released multiple chip solutions, featuring intelligence and low cost.

CONTACT US

For more product info, please contact z@quaming.com, or scan the QR code to follow us on Wechat.

This brief is for reference only and has no commitment. All content Contained herein is subject to changes without notice ©2023 Quaming Intelligent Technology Co., Ltd.

