

本文档分两部分，介绍了FFMedia中常用的数据类型和组件模型。

数据类型	
class MediaBuffer	MediaBuffer是所有组件中buffer的数据结构。
class VideoBuffer	VideoBuffer派生自MediaBuffer，是用作存储视频图像的buffer数据结构。
struct ImagePara	ImagePara是所有视频处理组件中描述图像数据格式的数据结构，是各个组件重要的初始化参数。
组件模型	
Class ModuleMedia	所有组件均派生自ModuleMedia类，ModuleMedia的成员中包含一个消费者队列，记录该组件的所有消费者；一个MediaBuffer队列，记录该组件所分配的buffer. MediaBuffer队列中存储当前组件的输出数据，MediaBuffer队列同时也是该组件的所有消费者的输入
Class ModuleCam	输入源组件，支持MIPI CSI摄像头和USB摄像头。通过V4L2接口操作获取数据
Class ModuleFileReader	输入源组件，读入文件，支持裸流文件和mp4、mkv文件。
Class ModuleRtspClient	输入源组件，拉取RTSP流，支持TCP/UDP/多播协议流
Class ModuleAacDec	数据处理组件，aac格式的音频解码和播放。
Class ModuleAacEnc	数据处理组件，aac格式的音频编码。
Class ModuleMppDec	数据处理组件，支持H264、H265、MJPEG解码。
Class ModuleMppEnc	数据处理组件，支持H264、H265、MJPEG编码。
Class ModuleRGA	数据处理组件，支持图像颜色格式转换、缩放、旋转等操作。
Class ModuleDrmDisplay	输出组件，DRM显示输出。
Class ModuleFileWriter	输出组件, 将数据存入到文件，支持MP4、MKV等格式文件存储
Class ModuleRtspServer	输出组件, RTSP推流。
Class ModuleRtmpServer	输出组件, RTMP推流。

class MediaBuffer

MediaBuffer是所有组件中buffer的数据结构，MediaBuffer包含一个指向一段内存空间的指针，其它成员用于描述该内存空间中所存储数据的附加属性。

方法	功能	参数	参数说明	返回值
MediaBuffer(size_t _size = 0);	构造函数	size_t _size = 0	通过malloc分配的buffer size, 默认为0, 不分配	
allocBuffer	申请内存	size_t _size	通过malloc分配buffer	
fillToEmpty	根据颜色格式，填充buffer为黑色			
getIndex	Index是记录当前MediaBuffe在组件的MediaBuffer队列中的序号			
setIndex				
getData	Data指向分配的buffer空间			
setData				
getSize	Size表示buffer的总大小			
setSize				
getActiveData	ActiveData指向buffer中有效数据的地址			
setActiveData				
getActiveSize	ActiveSize指向buffer中有效数据的大小			
setActiveSize				
getPUtimestamp	PUtimestamp是图像数据的显示时间戳			
setPUtimestamp				
getDUtimestamp	DUtimestamp是图像数据的解码时间戳			
setDUtimestamp				
getPrivateData	PrivateData指向一个void* 内存，用户可自定义记录关联的数据			
setPrivateData				
getExtraData	ExtraData功能与PrivateData类似，用来记录关联的数据			
setExtraData				
getEos	Eos用来记录当前buffer是否为数据流的最后一帧数据			
setEos				
getStatus	Status记录当前buffer的状态，为保留函数，目前没有流程会对其赋值			
setStatus				

increaseRefCount	RefCount记录当前buffer被其它消费者组件应用的次数，为内部函数，不建议使用。			
decreaseRefCount				
getRefCount				
setRefCount				
getMediaBufferType	MediaBufferType记录当前buffer中存储的数据的媒体类型，分为视频数据、音频数据、其它数据			
setMediaBufferType				

class VideoBuffer

VideoBuffer派生自MediaBuffer，是用作存储视频图像的buffer数据结构。buffer有两种类型，分别是基于drm的dma buffer、malloc申请的堆buffer

方法	功能	参数	参数说明	返回值
VideoBuffer	构造函数	BUFFER_TYPE type	申请的buffer类型	
getMppBuf	MppBuf是用于Mpp编解码时，传递给编解码器的buffer结构			
setMppBuf				
getDrmBuf	DrmBuf是buffer的drm数据结构			
setDrmBuf				
getBufFd	BufFd是buffer的drm文件句柄			
setBufFd				
getImagePara	ImagePara是buffer中存储的图像数据的格式参数。			
setImagePara				
getBufferType	BufferType是描述buffer类型是堆buffer还是dma buffer			
setBufferType				

struct ImagePara

ImagePara是所有视频处理组件中描述图像数据格式的数据结构，是各个组件重要的初始化参数。

成员	说明	
width	图像实际宽度	
height	图像实际高度	
hstride	图像虚宽，在图像处理过程中，大多数场景需要将图像的宽高以2^n对齐，因此会产生图像的虚宽虚高。	
vstride	图像虚高	
v4l2Fmt	图像颜色格式，以linux/videodev2.h中定义的颜色格式表示，在图像处理中，会涉及到Mpp、RGA、DRM等格式的转换	

Class ModuleMedia

所有组件均派生自ModuleMedia类，ModuleMedia的成员中包含一个消费者队列，记录该组件的所有消费者；一个MediaBuffer队列，记录该组件所分配的buffer. MediaBuffer队列中存储当前组件的输出数据，MediaBuffer队列同时也是该组件的所有消费者的输入buffer。

方法	功能	参数	参数说明	返回值
ModuleMedia	构造函数	const char* name = NULL	组件名称	
init	虚函数，初始化组件			0: 正常； 其它: 异常
start	启动组件工作线程			
stop	停止组件工作线程			
setProductor	设置组件的生产者	shared_ptr<ModuleMedia> module	生产者的对象指针	
getProductor	获取组件的生产者			生产者的对象指针
addConsumer	为组件添加一个消费者	shared_ptr<ModuleMedia> consumer	消费者的对象指针	
removeConsumer	删除组件的一个消费者	shared_ptr<ModuleMedia> consumer	消费者的对象指针	
getConsumer	获取组件的一个指定消费者	uint16_t index	消费者在当前组件的消费者队列中的序号	消费者的对象指针
getConsumersCount	获取组件的消费者个数			组件的消费者个数
setInputImagePara	设置组件的输入数据的图像格式	ImagePara		
getInputImagePara	获取组件的输入数据的图像格式			ImagePara
setOutputImagePara	设置组件的输出数据的图像格式	ImagePara		
getOutputImagePara	获取组件的输出数据的图像格式			ImagePara
setBufferSize	设置组件的单个buffer的大小，只有当组件不能够通过InputImagePara计算出单个buffer大小时，才需要调用此方法	size_t size		
getBufferSize	获取组件的单个buffer的大小。			size_t size
setBufferCount	设置组件的buffer个数			
getBufferCount	获取组件的buffer个数			
getBufferFromIndex	获取组件的一个指定buffer	uint16_t index	buffer在当前组件的buffer队列中的序号	buffer对象指针
getName	获取组件的名称			const char*
getIndex	获取当前组件在其生产者的消费者队列的序号			int index

getModuleStatus	获取组件当前的工作状态			STATUS_CREATED, STATUS_STARTED, STATUS_EOS, STATUS_STOPED,
getMediaType	获取组件输入数据的媒体类型			BUFFER_TYPE_VIDEO, BUFFER_TYPE_AUDIO, BUFFER_TYPE_ETC
setSynchronize	设置音视频同步组件			
setOutputDataCallback	为组件添加回调函数，处理当前组件的输出数据，每次生产数据时，均会被调用	void_object_p ctx, callback_handler callback	回调函数上下文 回调函数	
addExternalConsumer	为组件添加一个外部的消费者。其功能与添加回调相似，两者区别在于组件可以添加多个外部消费者，但是只能添加一个回调函数。	const char* name, void_object_p external_consume_ctx, callback_handler external_consume	外部消费者名称， 外部消费者上下文， 外部消费者的数据处理函数	外部消费者的对象指针
dumpPipe	打印出以当前组件为输入源的整个Pipe结构			
dumpPipeSummary	打印Pipe结构，以及Pipe中各个节点的运行统计数据			

Class ModuleCam

ModuleCam是输入源组件，支持MIPI CSI摄像头和USB摄像头。通过V4L2接口操作获取数据

方法	功能	参数	参数说明	返回值
ModuleCam	构造函数	const char* dev	摄像头的设备节点。	

Class ModuleFileReader

ModuleFileReader是读入文件组件，支持裸流文件和mp4、mkv文件。

方法	功能	参数	参数说明	返回值
ModuleFileReader	构造函数	const char* path bool loop_play	文件的路径 是否循环播放	
audioExtraData	获取音频额外数据			
audioExtraDataSize	获取音频额外数据大小			
videoExtraData	获取视频额外数据			
videoExtraDataSize	获取视频额外数据大小			
setFileReaderSeek	设置文件读取时间点	int64_t ms_time	切换到该时间点读取	
getFileReaderMaxSeek	获取文件最大时长		媒体文件持续时长	

Class ModuleRtspClient

ModuleRtspClient是拉取RTSP流的组件，支持TCP/UDP/多播协议流

方法	功能	参数	参数说明	返回值
ModuleRtspClient	构造函数	const char* url, RTSP_STREAM_TYPE_stream_type = RTSP_STREAM_TYPE_UDP, bool enable_video = true, bool enable_audio = false	rtsp流的地址 rtsp流的传输协议，默认为UDP流  是否拉取视频流，默认为真 是否拉取音频流，默认为假	
videoExtraData	获取视频额外数据			
videoExtraDataSize	获取视频额外数据大小			
audioExtraData	获取音频额外数据			
audioExtraDataSize	获取音频额外数据大小			
audioChannel	获取音频通道数			
audioSampleRate				
videoFPS	获取视频帧率			
setTimeOutSec	设置连接、接收数据超时时间		默认2秒	

Class ModuleAacDec

ModuleAacDec是aac格式的音频解码和播放组件。

方法	功能	参数	参数说明	返回值
ModuleAacDec	构造函数	const uint8_t* _extradata, unsigned _extradata_size, int _sample_rate, int nb channels = -1	音频额外数据 额外数据大小 样品速率，可不用设置，从音频数据解析 通道数，可不用设置，从音频数据解析	
setAlsaDevice	设置alsa设备	string dev	设置alsa设备，开启音频播放	

Class ModuleAacEnc

ModuleAacEnc是aac格式的音频编码组件。

方法	功能	参数	参数说明	返回值
ModuleAacEnc	构造函数	SampleFormat_fmt, int_sample_rate, int_nb_channels	样品格式: SAMPLE_FMT_S16, SAMPLE_FMT_NONE 样品速率: 通道数: 1~8	
setAot	设置、获取aot	int		
getAot				
setBitrate	设置位速率			
getBitrate				
setAfterburner				
getAfterburner				
setEldSbr				
getEldSbr				
setVbr				
gerVbr				

Class ModuleMppDec

ModuleMppDec是视频解码组件，支持H264、H265、MJPEG解码。

方法	功能	参数	参数说明	返回值
ModuleMppDec	构造函数	const ImagePara& input_para	输入数据流的图像参数，通过图像参数判断解码器类型	
ModuleMppDec	构造函数	const ImagePara& input_para, DecodeType type	输入数据流的图像参数 解码器类型	

Class ModuleMppEnc

ModuleMppEnc是视频解码组件，支持H264、H265、MJPEG编码。

方法	功能	参数	参数说明	返回值
ModuleMppEnc	构造函数	EncodeType type const ImagePara& input_para int fps = 30 int gop = 60 int bps = 2048 EncodeRcMode mode = ENCODE_RC_MODE_CBR, EncodeQuality quality = ENCODE_QUALITY_BEST, EncodeProfile profile = ENCODE_PROEFILE_HIGH	编码器类型 输入数据的图像参数 编码帧率，默认为30fps 两个I帧之前的间隔帧数，默认为60 编码的码率 码率控制模式，默认为固定码率模式  编码质量，只在可变码率模式时有效  H264/265编码时的profile参数	

Class ModuleRGA

ModuleRGA是图像处理中，颜色格式转换、缩放等操作的组件。

方法	功能	参数	参数说明	返回值
ModuleRga	构造函数			
ModuleRga	构造函数	const ImagePara& input_para, const ImagePara& output_para, RgaRotate rotate	输入数据的图像参数 输出数据的图像参数 图像旋转角度	
setSrcPara	设置输入数据的图像参数			
setDstPara	设置输出数据的图像参数			
setRotate	设置输出数据的图像旋转角度			
newModuleMediaBuffer	申请rga模块buf	VideoBuffer::BUFFER_TYPE buffer_type	申请buf数据类型	
exportUseMediaBuffer	导出rga内部buf	shared_ptr<MediaBuffer> match buffer,	rga内部buf 通过申请或导出接口的inbuf	



Class ModuleDrmDisplay

ModuleDrmDisplay是显示输出的组件，支持多个图像输出到一个plane中，或者多个图像显示到独立的plane中。plane相当于图像显示的画布，可以是全屏的，也可以是非全屏的，plane的总个数是由系统决定的。window位于plane上，对应一个具体的显示画面，一个plane中可以有多个window。

方法	功能	参数	参数说明	返回值
ModuleDrmDisplay	构造函数	const ImagePara& input_para	输入数据的图像参数	
setPlanePara	设置显示plane的参数	uint32_t fmt, uint32_t plane_id = 0, PLANE_TYPE plane_type, uint32_t plane_zpos	图像显示的格式，如NV2，ARGB32等 显示设备的plane id，默认为0，自动选择 显示plane的类型，默认为overlay图层 显示plane的zpos, 默认自动选择zpos	
getDisplayPlaneSize	获取显示plane的大小，显示设备 支持的最大图像显示尺寸			
setPlaneSize	设置显示plane的大小。			
setWindowSize	设置窗口的位置和大小	uint32_t x, uint32_t y, uint32_t w, uint32_t h		

Class ModuleFileWriter

ModuleFileWriter是将数据存入到文件的组件，支持MP4、MKV等格式文件存储

方法	功能	参数	参数说明	返回值
ModuleFileWriter	构造函数	const ImagePara& para string path	输入数据的图像参数 保存文件的路径	
restart	重新写入文件	string file_name	保存文件的路径	
setMaxFrameCount	设置文件最大帧数，超过则另开文件写入			
setVideoParameter	设置视频参数		从图像自动解析，可不用设置	
setVideoExtraData	设置视频额外数据		从视频数据自动生成，可不用设置	
setAudioParameter	设置音频参数			
setAudioExtraData	设置音频额外数据			

Class ModuleRtspServer

ModuleRtspServer是RTSP推流的组件。

方法	功能	参数	参数说明	返回值
ModuleRtspServer	构造函数	const ImagePara& para const char* path int port	输入的图像参数 推流的路径 推流的输出端口	

Class ModuleRtmpServer

ModuleRtmpServer是RTMP推流的组件。

方法	功能	参数	参数说明	返回值
ModuleRtmpServer	构造函数	const ImagePara& para const char* path int port	输入的图像参数 推流的路径 推流的输出端口	
setMaxClientCount	设置最大客户端连接数量，默认10个			
getMaxClientCount	获取最大客户端连接数量			
getCurClientCount	获取当前已连接客户端数量			
setMaxTimeOutCount	设置最大连续发送超时次数，超过则暂停对该客户端发送数据			
getMaxTimeOutCount	获取最大连续发送超时次数			
setTimeoutSec	设置超时时间	int sec, int usec	秒、微秒	