

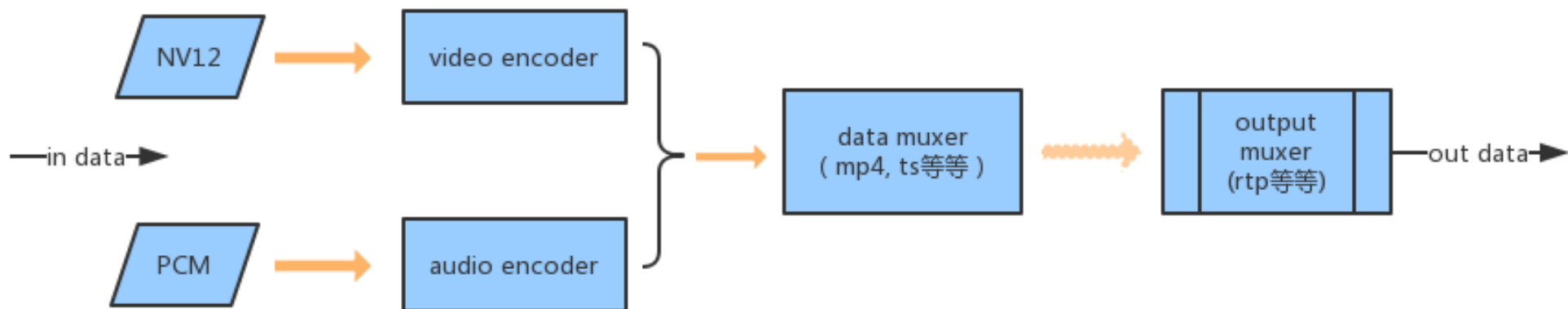
RK1108 音视频开发介绍

Sep., 2016
第三产品部 王航

Agenda

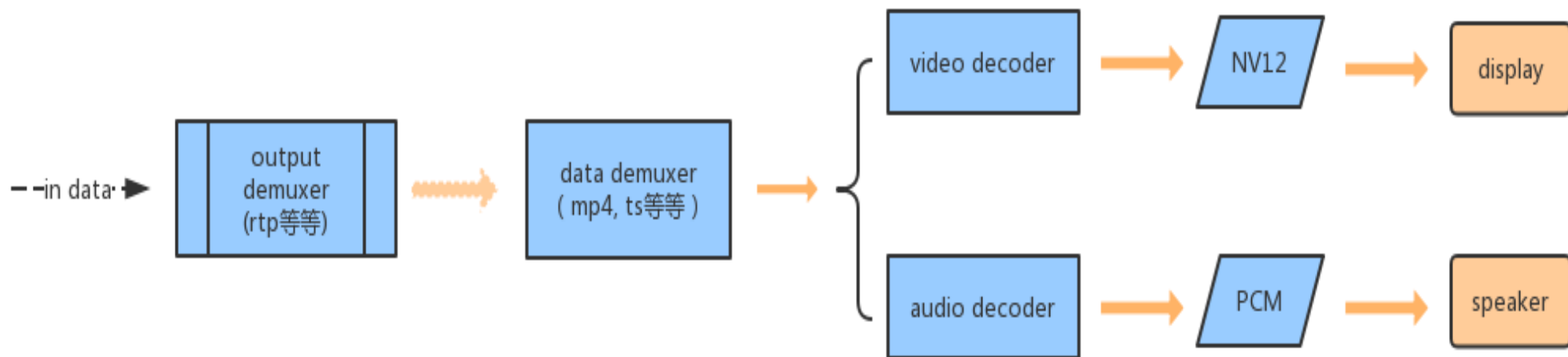
- 音视频开发工作内容（做什么）
- 功能实现（做哪些）
- 选择接口（如何做）
- Q&A

音视频开发内容



编码流程图

音视频开发内容



解码流程图

功能实现

- rk已实现功能

- 多路录像保存到sd卡
- 网络视频客户端实时预览
- 移动侦测录像
- 紧急录像
- 板端视频查看回放

- 客户自己要实现的功能

- ? ?

功能实现

- 编码主要注意点

- 输入源数据多样

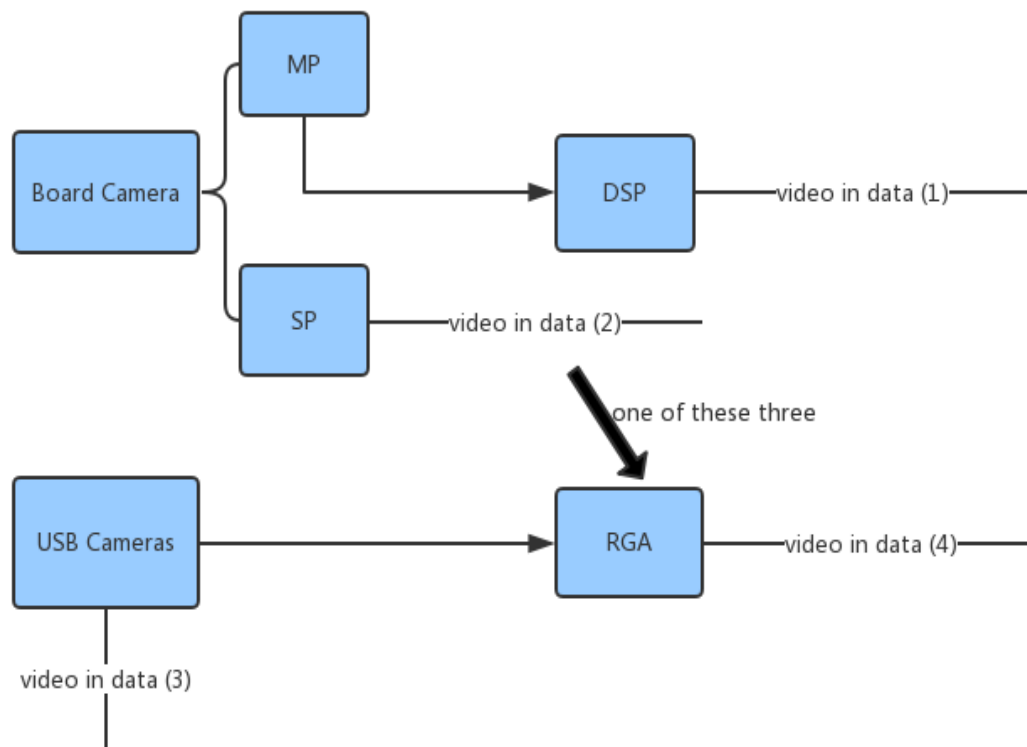
- ◆ 板载Camera

- MP (main path) $\geq 1920 \times 1080$
- SP (self path) $\leq 1920 \times 1080$

- ◆ USB Camera

- ◆ 图像处理后数据

- DSP
- RGA



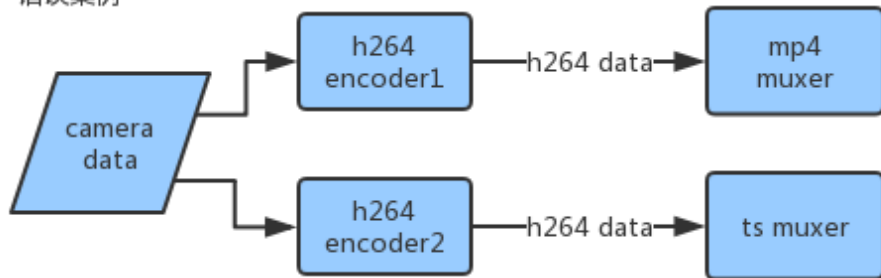
功能实现

- 编码主要注意点

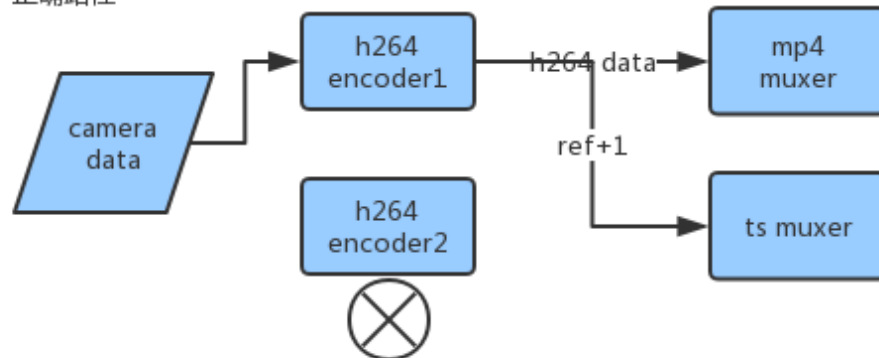
- 功能随意切换，同时减少内存访问负荷

- ◆ 输入源数据与编译器实例一一对应

错误案例



正确路径



选择接口

1. rk已实现的功能接口

—— 适合想直接使用rk demo app来做开发的客户

● 编码mp4并写入sd卡

- int video_record_startrec(void);
 - 函数说明：开始录像并写入文件
- void video_record_stoprec(void);
 - 函数说明： 停止录像

✓ 关键配置（参考video_encode_init函数）

配置encode config, 音视频参数（视频格式固定为nv12）

图像宽高 Width, Height

视频码率 video_bit_rate

视频帧率 stream_frame_rate

video_encoder_level

I帧间隔 video_gop_size

音频码率 audio_bit_rate

音频帧率 audio_sample_rate

音频声道 channel_layout

音频采样数 input_nb_samples

音频数据格式 input_sample_fmt

选择接口

1. rk已实现的功能接口

—— 适合想直接使用rk demo app来做开发的客户

- 图像客户端实时预览

- void video_record_start_ts_transfer(char *url);
 - 函数说明：开始网络传输ts流
 - 输入参数url: 协议以及传输目的地址，比如udp://239.1.1.1:1234
- void video_record_stop_ts_transfer(char sync)
 - 函数说明：停止网络传输ts流

选择接口

1. rk已实现的功能接口

—— 适合想直接使用rk demo app来做开发的客户

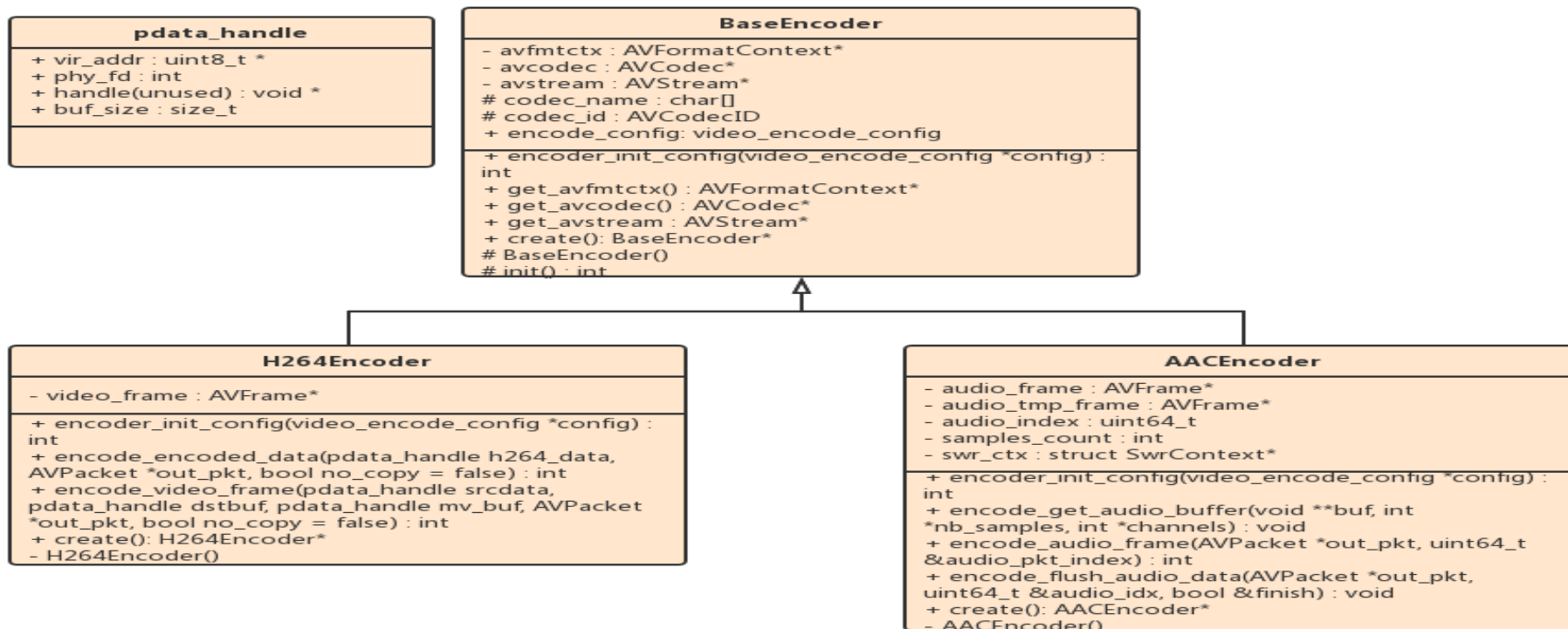
- 音频文件播放
 - int audio_play0(char *audio_file_path)
 - 函数说明：开始网络传输ts流
 - 输入参数audio_file_path: 文件路径地址，比如/usr/share/sounds/click.wav
- 录像文件播放
 - int videoplay_init(char *filepath, HWND hWnd)
 - 函数说明：初始化videoplay实例
 - 输入参数filepath：文件路径地址，比如/mnt/sdcard/nor/20160925.mp4
 - hWnd：ui 句柄
 - int videoplay_exit(void)
 - 函数说明：退出文件播放

选择接口

2. rk已实现的音视频封装类

—— 适合想自己管理上面线程和数据，但又不关心音视频编码封装等细节的客户

● 编码类

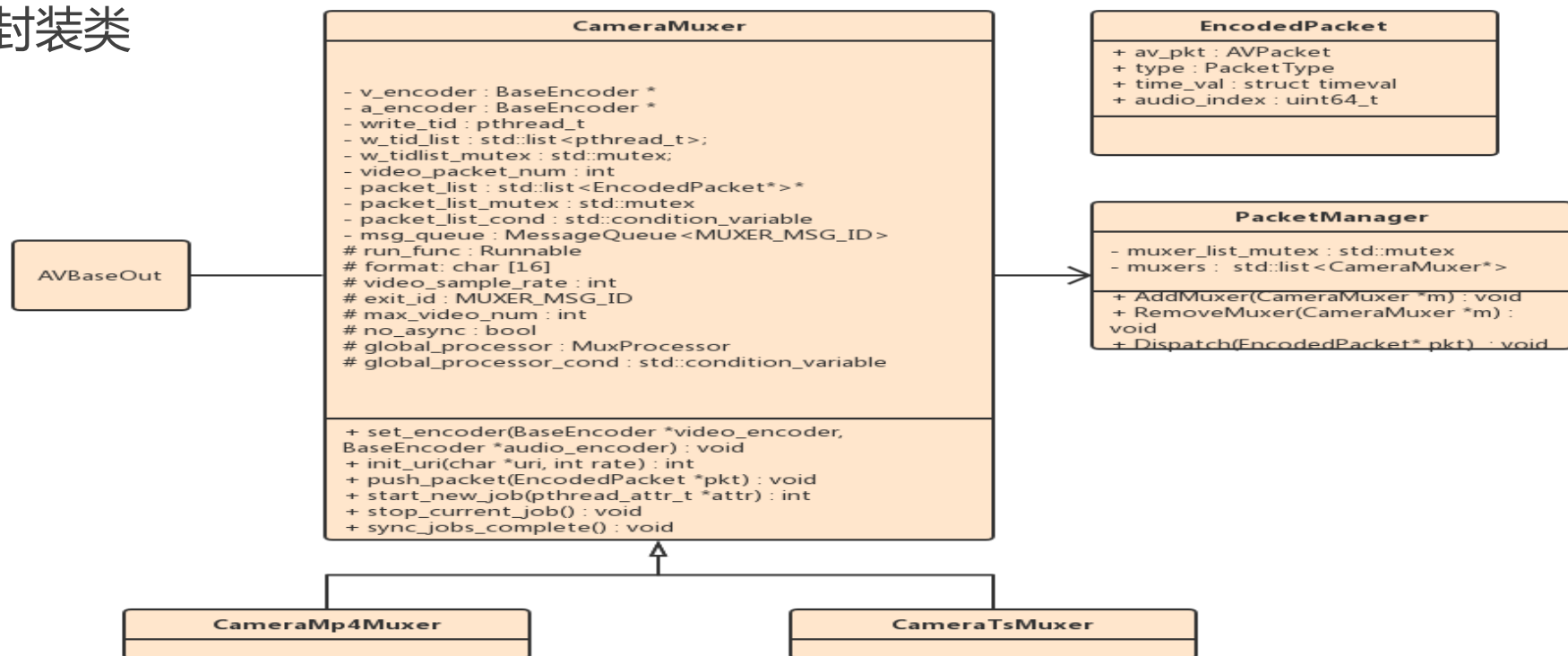


选择接口

2. rk已实现的音视频封装类

—— 适合想自己管理上面线程和数据，但又不关心音视频编码封装等细节的客户

● 文件封装类

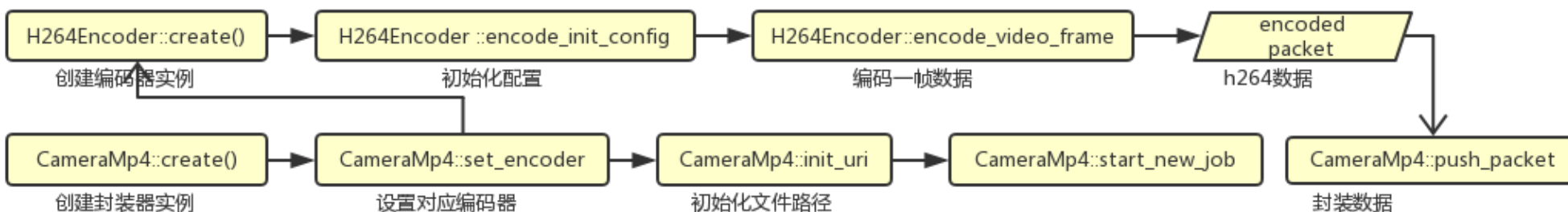


选择接口

2. rk已实现的音视频封装类

—— 适合想自己管理上面线程和数据，但又不关心音视频编码封装等细节的客户

● 使用范例



选择接口

3. ffmpeg

—— 适合想自己管理上面线程和数据，对ffmpeg很熟悉的客户

- ffmpeg优点

- 支持格式全 (mp4 TS ...)
- 扩展性好
 - ◆ 需要新的文件格式支持，只要加上对应的编译选项即可
- 添加新codec较简单
 - ◆ 目前只要对应我们的codec驱动，实现init，encode/decode，deinit
- 网络资源丰富，开发人员易上手
- 对于前期某些功能方便联合测试
 - ◆ 比如 rtp等网络输出协议，ffmpeg有现成的实现，可先拿来提供给手机端开发测试，而无需等待板端实现自己的rtp等传输
- Rk编解码器已集成在里面

选择接口

3. ffmpeg

———— 适合想自己管理上面线程和数据，对ffmpeg很熟悉的客户

- ffmpeg编译

- 参考 `$(ffmpeg code path)/autoconfig.sh`

- ◆ 注意支持格式的版权问题，不需要的不要编译进来，同时也能减少库大小

- ◆ 主要修改 `--enable-encoder`, `--enable-decoder`, `--enable-muxer`, `--enable-demuxer`, `--enable-protocol`

选择接口

3. ffmpeg

—— 适合想自己管理上面线程和数据，对ffmpeg很熟悉的客户

- ffmpeg接口使用

- 参考 \$(ffmpeg code path)/doc/examples/muxing.c和demuxing.c，或者参考BaseEncoder和CameraMuxer，DecodeDemuxer类的实现

- 关键结构体

- ◆ AVFormatContext

- AVStream

- AVCodecContext

- AVInputFormat / AVOutputFormat AVIOContext

- ◆ AVFrame

- ◆ AVPacket

因为硬件需要使用物理连续内存块，修改了AVFrame和AVPacket，增加uint32_t user_reserve_buf[8]以存储描述物理连续内存块的结构体pdata_handle

选择接口

4. rk video codec api

—— 适合只对rk硬件ip感兴趣，同时音视频有自己积累的客户

- 参考 `$(ffmpeg code path)/libavcodec/mpp_encoder.c`和`mpp_decoder.c`
 - 具体api讲解见 《视频编解码 - 陈恒明》

Q&A

Thanks!