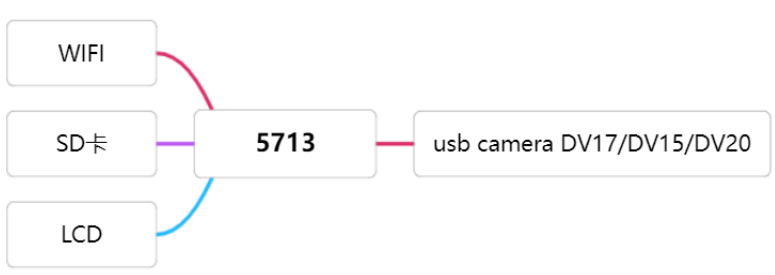
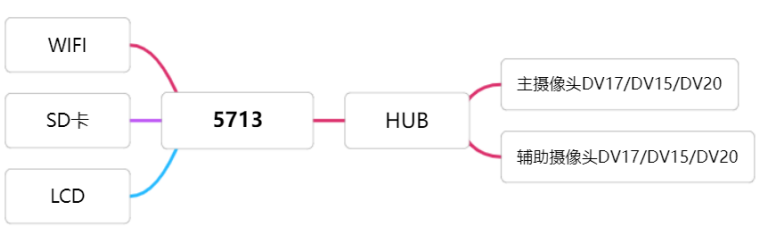
**AC57做UVC主机对接多UVC从设备说明**

1. 背景说明

传统⻔锁猫眼⽅案是前板⼀个USB UVC摄像头，后板是DV17或者DV15接收前板USB UVC图像后进⾏推屏显示或者传到WiFi模块给云端使⽤。方案示意图如下：



由于市场需求变化，我司推出AC57可对接多个USB UVC摄像头的方案，方案示意图如下：



由图可知AC57通过hub支持同时接多个USB UVC摄像头，主机端完成多个USB UVC摄像头枚举并出图。

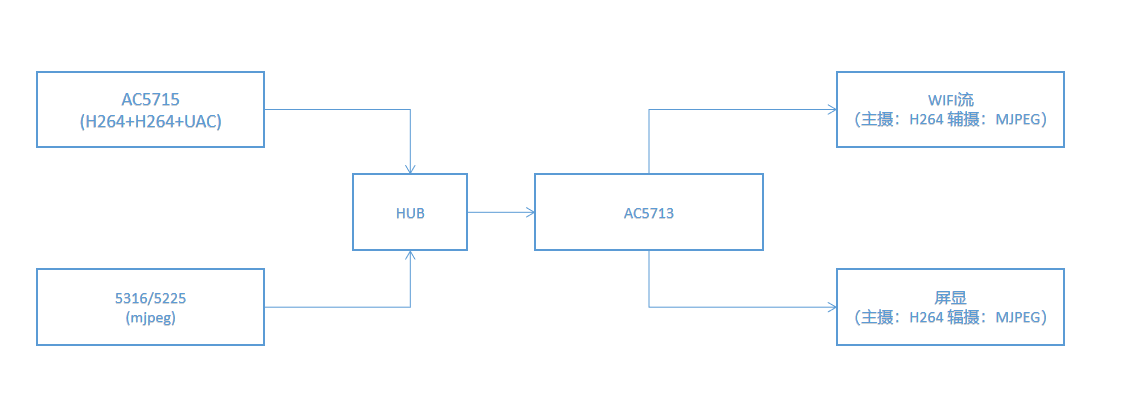
1. 方案列举与说明

目前具体方案支持如下：

1、主摄像头采用AC5715输出双码流方案（H264+H264+UAC）

辅摄像头采⽤5316/5225 单MJPEG码流方案UVC⽅案

方案示意图如下：

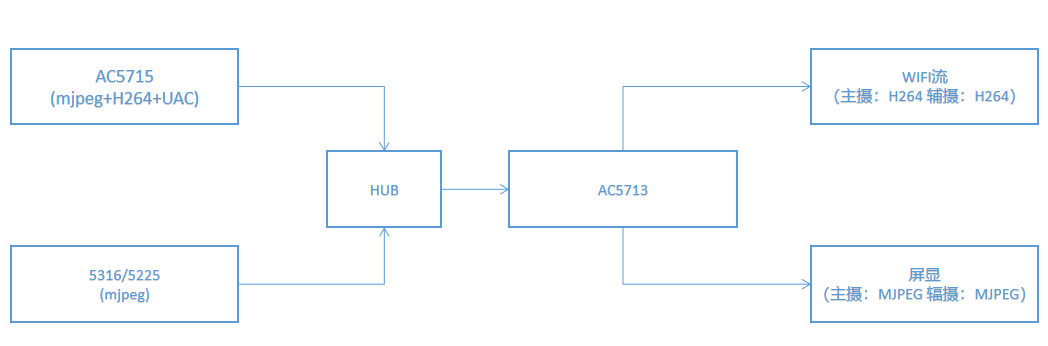


当使用方案1时，由于AC57 H264编解码不能同时进行，当主摄像头进行H264解码屏显显示时，就没有办法对辅摄像头的MJPEG码流进行解码重编码H264，导致WiFi图传只能一路传H264,一路传MJPEG。

2、主摄像头采用AC5715输出双码流方案(mjpeg+H264+UAC)

辅摄像头采⽤5316/5225 单MJPEG码流UVC⽅案

方案示意图如下：

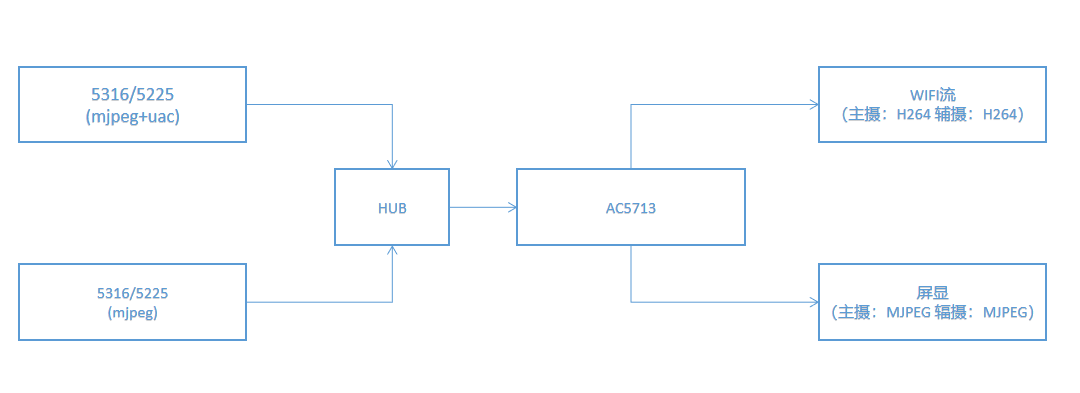


当使用方案2时，由于主摄像头的第一路码流为MJPEG，所以当屏显选择第一路码流解码显示时，辅摄像头的MJPEG码流还可以进行解码重编码H264，WiFi图传就可以传两路H264码流。

3、主摄像头：5316/5225 单MJPEG码流UVC⽅案带UAC功能

辅摄像头：5316/5225 单MJPEG码流UVC⽅案

方案示意图如下：



当使用方案3时，由于主辅摄像头的码流都为MJPEG，所以当屏显的mjpeg解码显示时，主辅摄像头的MJPEG码流还可以进行解码重编码H264，WiFi图传就可以传两路H264码流。

1. AC57 配置UVC从机说明
2. **软件版本**

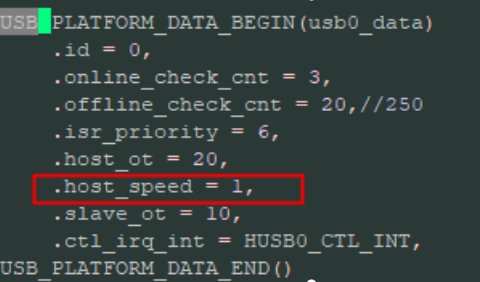
基于gitlab 版本如下：

f2a89293560529e9d1df408bc809365efcbd9a04

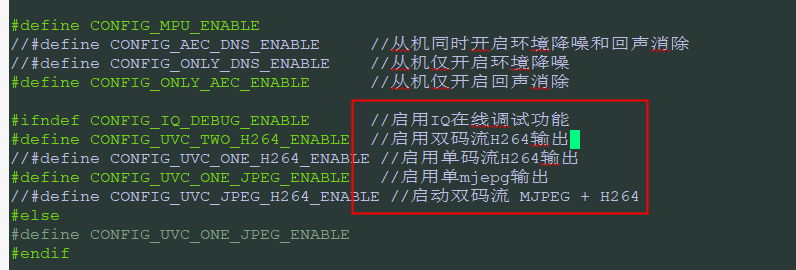
1. **修改说明**

从机使用工程 AC57XX\_CAR\_CAMERA.cbp:

特别是在板级配置文件中，将从机USB设置为2.0协议

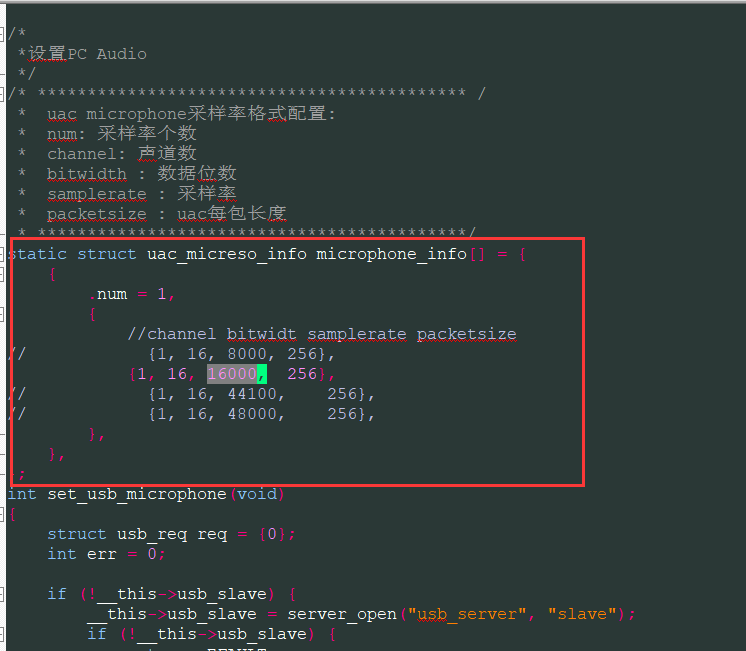


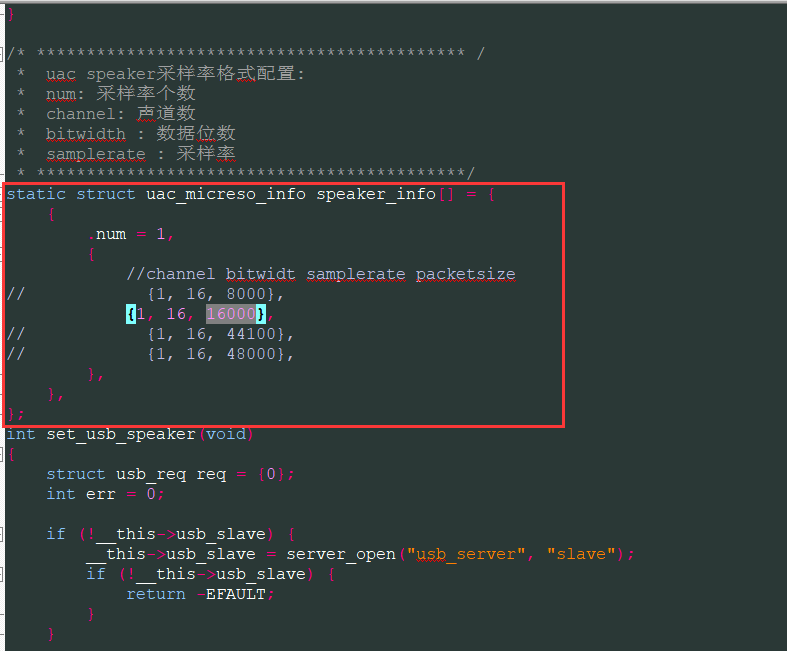
UVC码流输出格式选择,SDK已使用相关宏进行配置，在app\_config.h中，选择从机码流输出格式即可。



注：新版本sdk中，根据从机的功能组合，增加了宏控制，客户只需要简单配置宏，就可以完成从机功能的配置，需要注意的是双h264的从机，在主机枚举时将占据两个UVC号，如果第一路占据UVC0，那么第二路将占据UVC1。

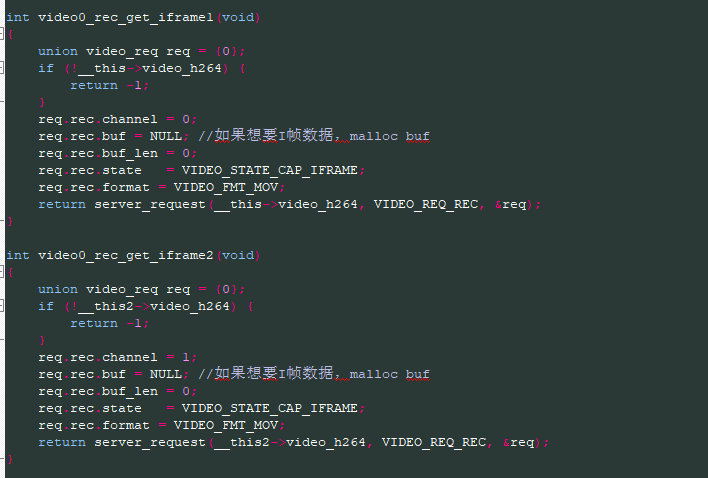
在usb\_app.c中设置uac只保留16000采样率，这个务必修改，当然也可以采用8000采样率，可以降低，但是不能提高。





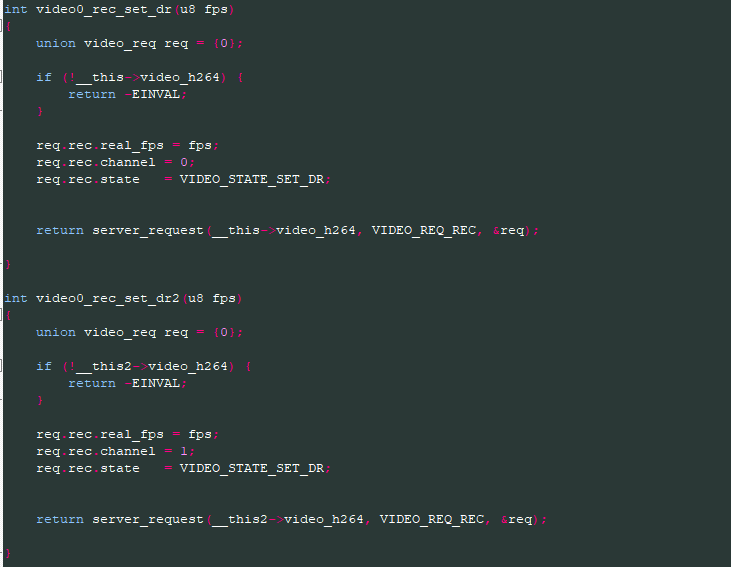
在usb\_cam\_h264.c中，增加强制I帧设置，动态帧率设置，动态码率设置

1、强制I帧设置



针对双H264方案，强制I帧的指令是分开的，需要第一路执行强制I帧的指令，则需要调用video0\_rec\_get\_iframe1,需要第二路执行强制I帧的指令，video0\_rec\_get\_iframe2。

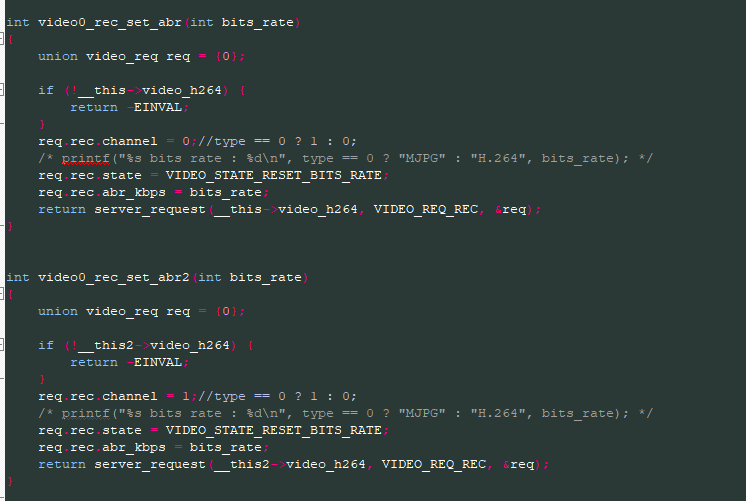
2、动态帧率设置



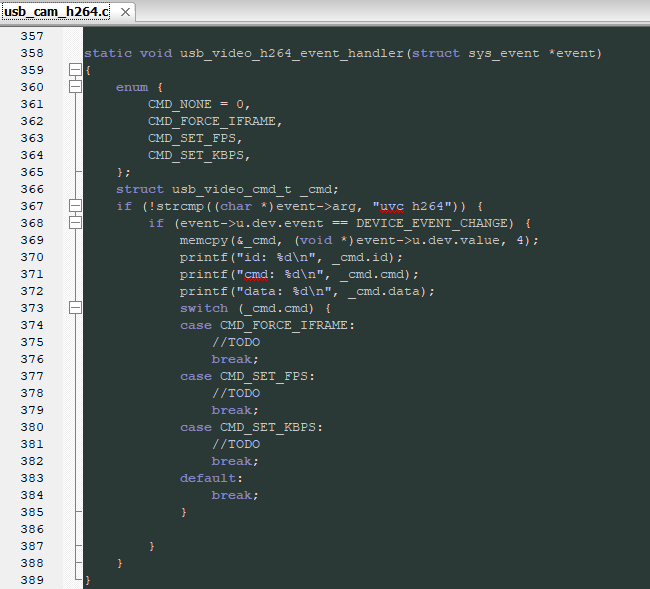
针对双H264方案，动态帧率的指令是分开的，需要第一路执行动态帧率的指令，则需要调用video0\_rec\_set\_dr函数,需要第二路执行动态帧率的指令，video0\_rec\_set\_dr2函数。

3、动态码率设置:

针对双H264方案，动态帧率的指令是分开的，需要第一路执行动态帧率的指令，则需要调用video0\_rec\_set\_abr函数,需要第二路执行动态帧率的指令，video0\_rec\_set\_abr2函数。



在usb\_cam\_h264.c中的usb\_video\_h264\_event\_handler函数中接收来自UVC主机的自定义扩展指令并调用对应的接口执行动作。



发送实例请参考video\_rec.c中video3\_stream\_cmd\_test函数

1. **使用PotPlayer 64 bit验证功能**



注：双路码流都能成功出图并且码率接近设置的值，即功能正常。

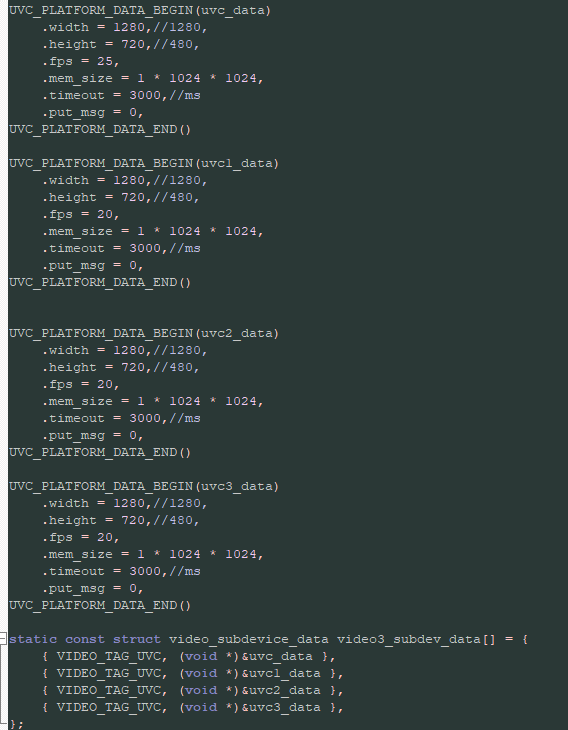
1. AC57 配置UVC主机接收多UVC从设备功能
2. **软件版本**

需要基于gitlab版本修改，版本如下：

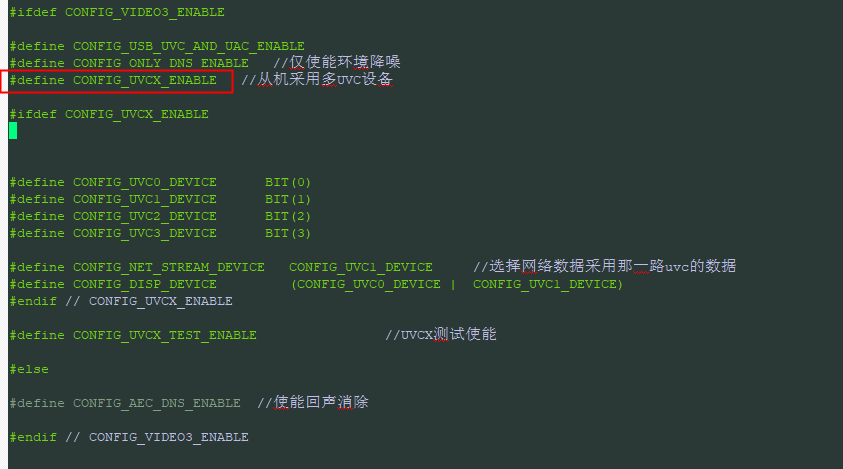
f2a89293560529e9d1df408bc809365efcbd9a04

1. **修改说明**

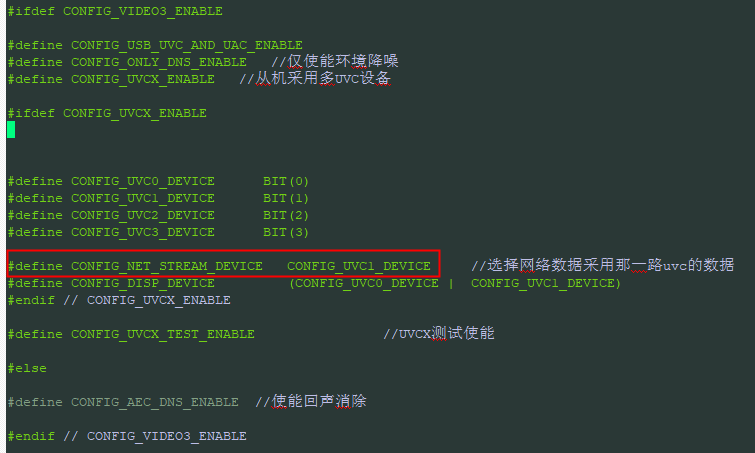
由于多uvc从设备的码流已对接到录像写卡的流程中，所以每增加一个uvc设备板级中需要新增一个uvc\_data，如果板级中少了这个配置会导致功能异常，请客户务必注意。



在app\_config.h启用主机接收多uvc从设备码流配置

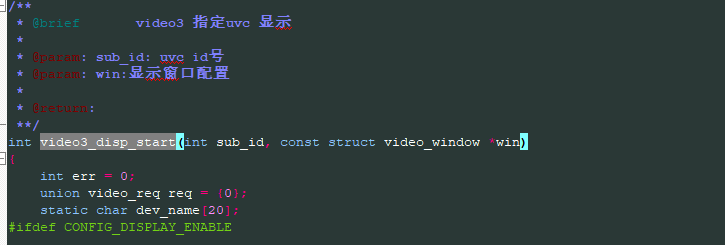


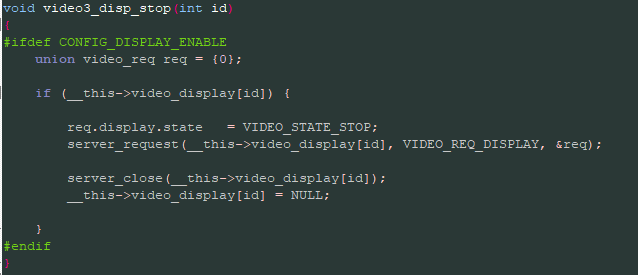
选择网络视频流来自于哪一个uvc设备：



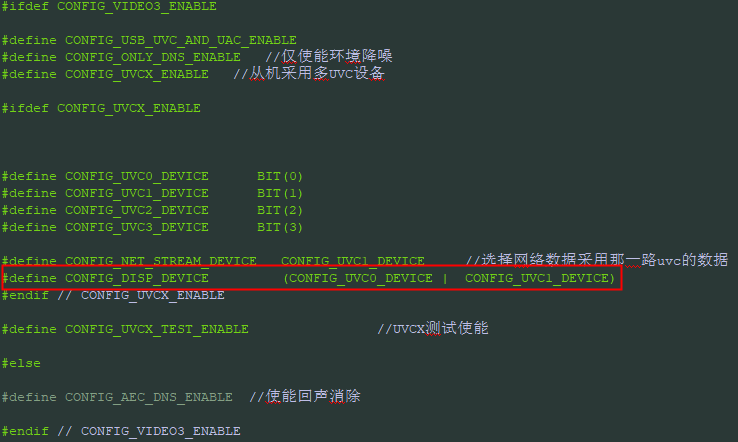
注：本示例只发送一路视频流到网络中，实际使用可根据需要自行选择。

针对video3的屏显启停，新增video3\_disp\_start，video3\_disp\_stop函数，其中subid和id表示的就是uvc的设备号，如下图所示：





选择屏显画面来自于哪一个uvc从设备



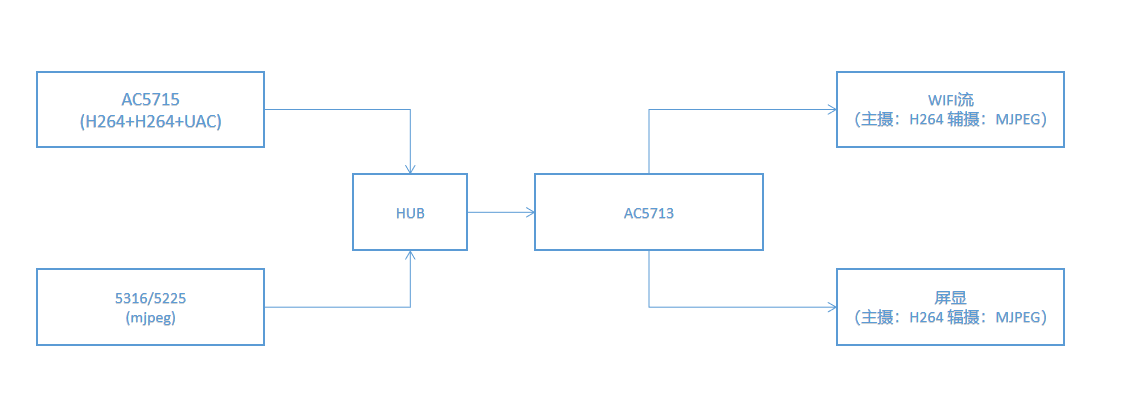
注：目前最多支持4个uvc从设备，按上线顺序分别为uvc0、uvc1、uvc2、uvc3,其中uvc0占据设备号3.0 - 3.4,uvc1占据设备号3.5-3.6，uvc2占据设备号3.7-3.8，uvc3占据设备号3.9-3.A。

下面根据不同方案屏显设备选择进行举例：

1、主摄像头采用AC5715输出双码流方案（H264+H264+UAC）

辅摄像头采⽤5316/5225 单MJPEG码流UVC⽅案

方案示意图：

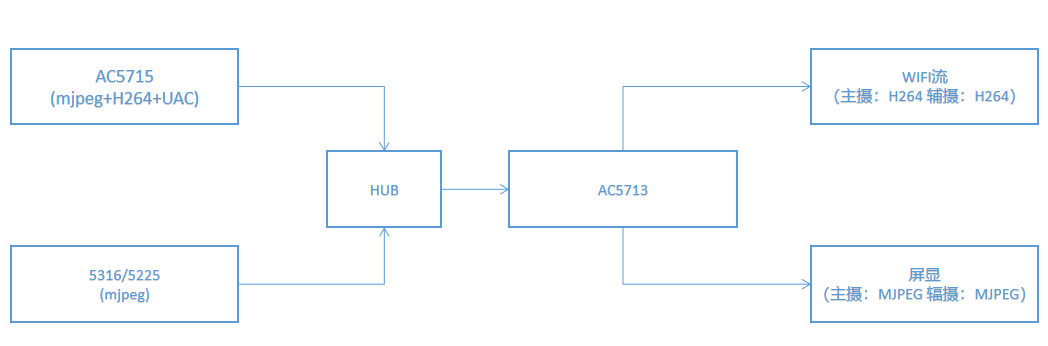


我们先假定主摄像头必须先于辅摄像头上线，AC5715 占据UVC0和UVC1,辅摄像头 5316/5225占据UVC2，那么显示设备只能选择为UVC0 & UVC2。

2、主摄像头采用AC5715输出双码流方案（mjpeg+H264+UAC）

辅摄像头采⽤5316/5225 单MJPEG码流UVC⽅案

方案示意图：

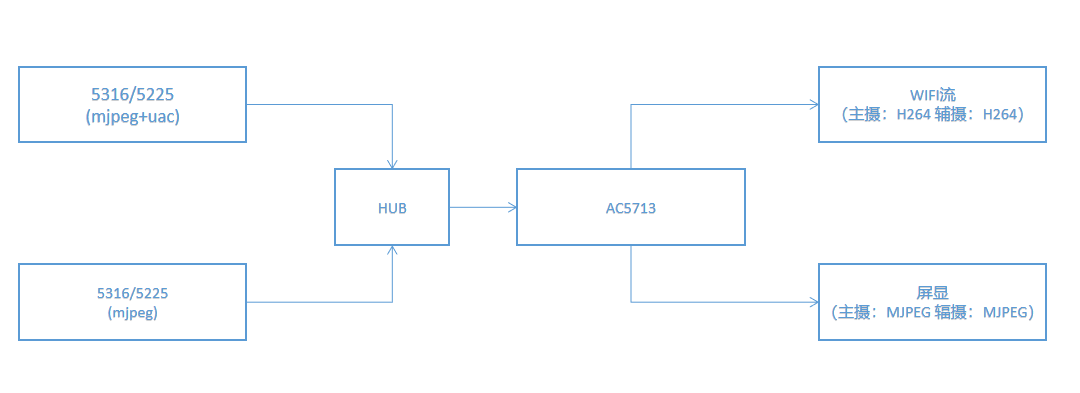


我们先假定主摄像头必须先于辅摄像头上线，AC5715 占据UVC0和UVC1,辅摄像头 5316/5225占据UVC2，可以选择为UVC0 & UVC2。

3、主摄像头：5316/5225 单MJPEG码流UVC⽅案带UAC功能

辅摄像头：5316/5225 单MJPEG码流UVC⽅案

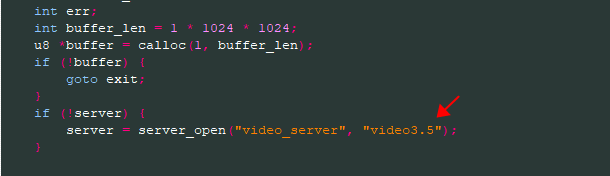
方案示意图：



我们先假定主摄像头必须先于辅摄像头上线，5316/5225占据UVC0,辅摄像头 5316/5225占据UVC1，那么显示设备可以选择为UVC0 & UVC1。

1. DV17 对接多UVC设备主机sdk 讲解
2. **UVC设备录像保存文件说明**

目前UVC设备录像保存文件，区别于以往的就是所传递的设备号不同，按照前述说明，目前第一路需传3.0到3.4，第二路需传3.5-3.6。第三路需传3.7-3.8，第四路需传3.9-3.A.所以客户需要录像录那一路就传那一路的设备号既可以，设备号传如下图所示：



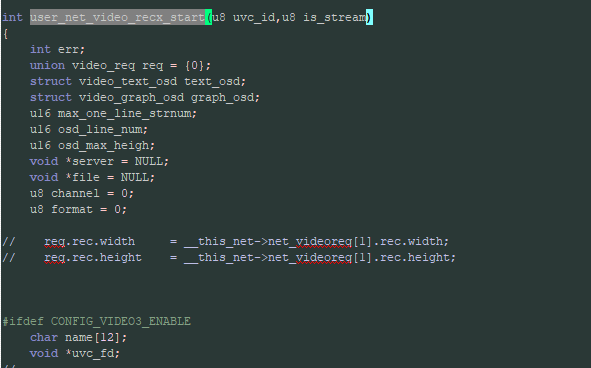
1. **关于获取视频帧和音频帧讲解**

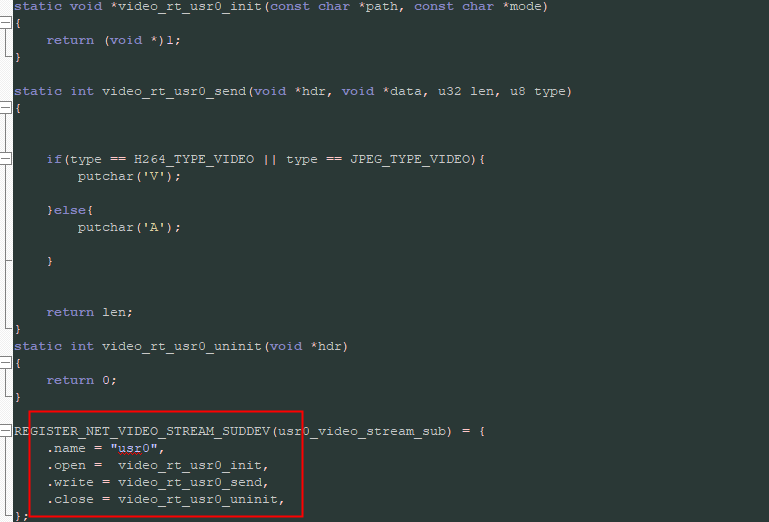
有两种方式可以取视频帧，第一种是通过启动录像的方式，另外一个是通过直接打开uvc设备获取视频帧，这里建议采用第一种方式，可以同时获取到音视频帧。

下面针对第一种取帧方式说明

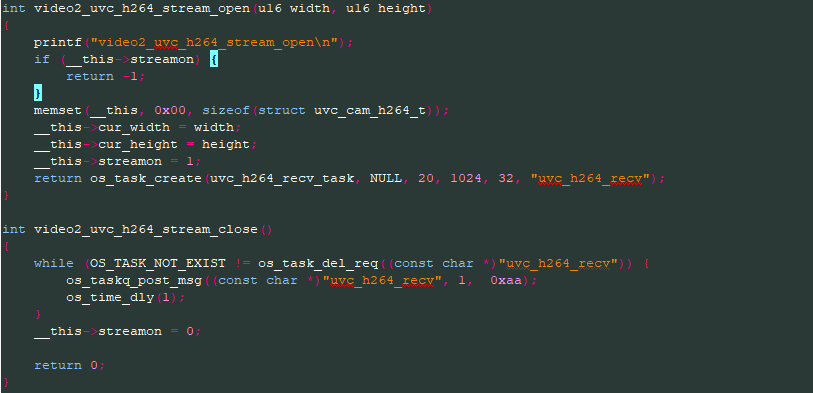
可以参考这个测试函数user\_net\_video\_recx\_start，其中uvc\_id表示要录哪一个uvc设备，is\_stream = 1表示不存卡录像，is\_stream = 0表示不存卡录像。

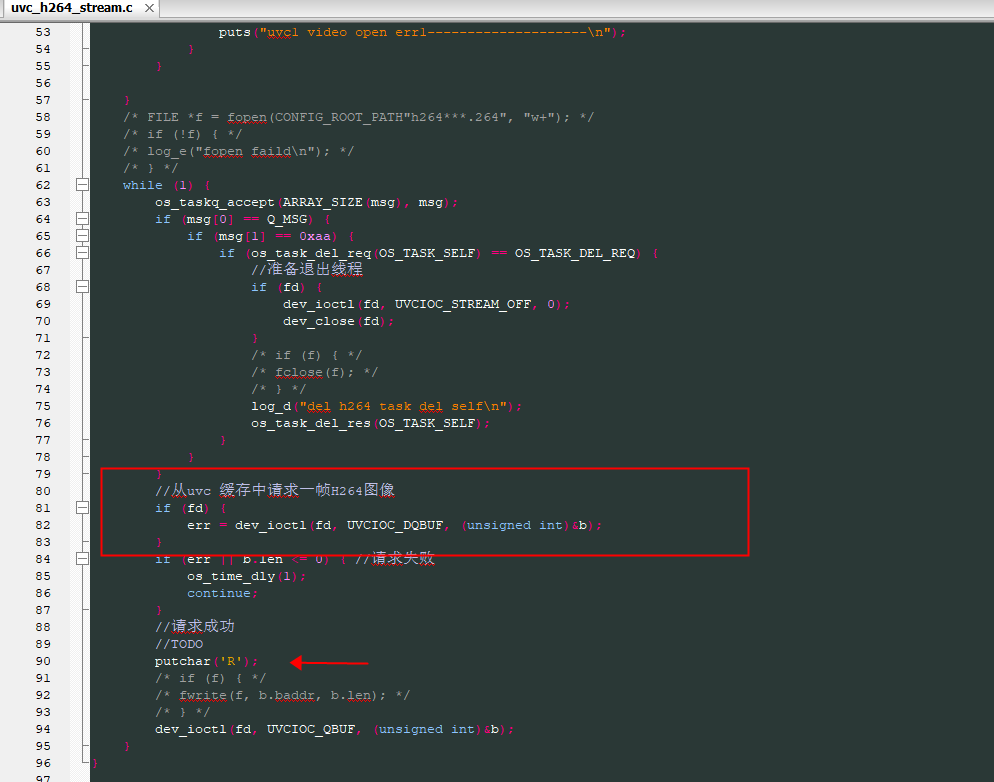
这个测试函数功能主要是启动任意一路uvc设备录像，并将可选择保存成文件或者视频帧回调到以下接口：





第二种方式可以参考apps\ipc\_doorbell\uvc\_h264\_stream.c中的方式，如果没有，看一下其他工程有没有把这个文件获取过来就行，但是这个方式不经过封装器，没办法保存成录像文件，建议采用第一种方式。





1. **关于AC57对接云平台说明**

如果客户使用小草平台对接可以直接使用IPC\_DOORBELL工程，如果客户采用其他平台，则需要使用wifi\_car\_camera工程，对接过程可以参考借鉴IPC\_DOORBELL工程。

一般平台对接可以总结为以下三步：

1. 平台库初始化与平台连接

这一步一般需要先提供工具链给到平台方，由平台方将他们的代码编译成库，并且编译连接通过,调用平台库初始化并完成平台连接后，当前操作完成。

1. 获取AC57的音视频帧

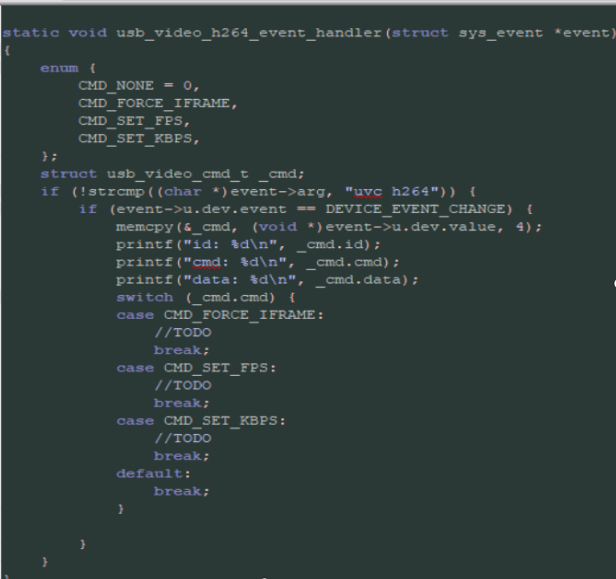
这一步可以参考本文档中关于如何获取音视频帧的说明，建议采用启动录像的方式获取音视频帧

1. 完成相关功能对接

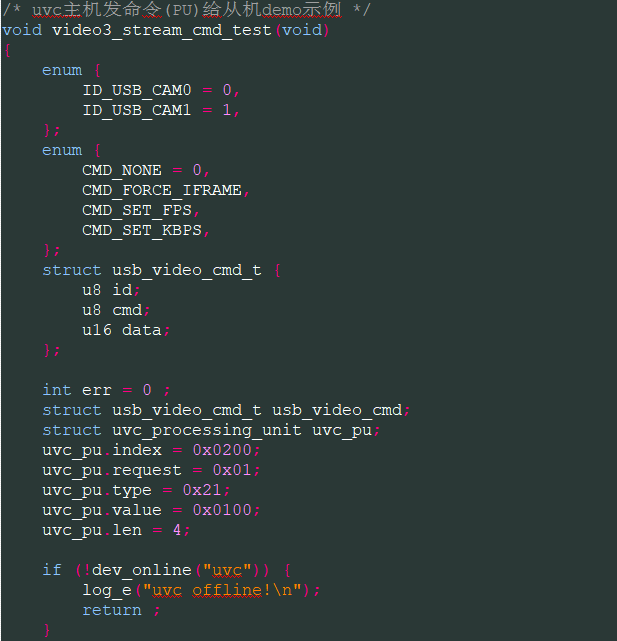
这一步根据平台方的对接文档，完成各个功能模块的对接，最终完成整个平台的接入。

1. **关于UVC扩展单元请求指令发送**

AC5715 UVC从机接收到主机发送过来扩展指令的处理函数如下：



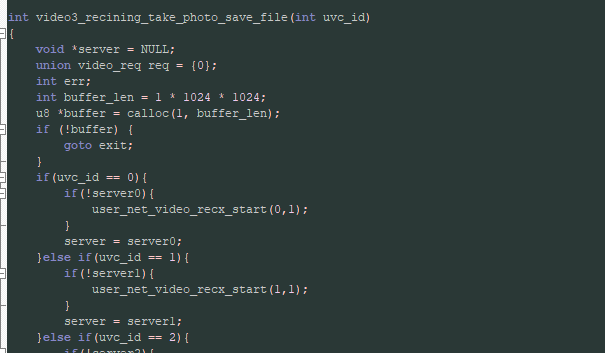
主机发送扩展指令的参考函数如下：



注：由于篇幅有限，请对照sdk查看video3\_stream\_cmd\_test函数。

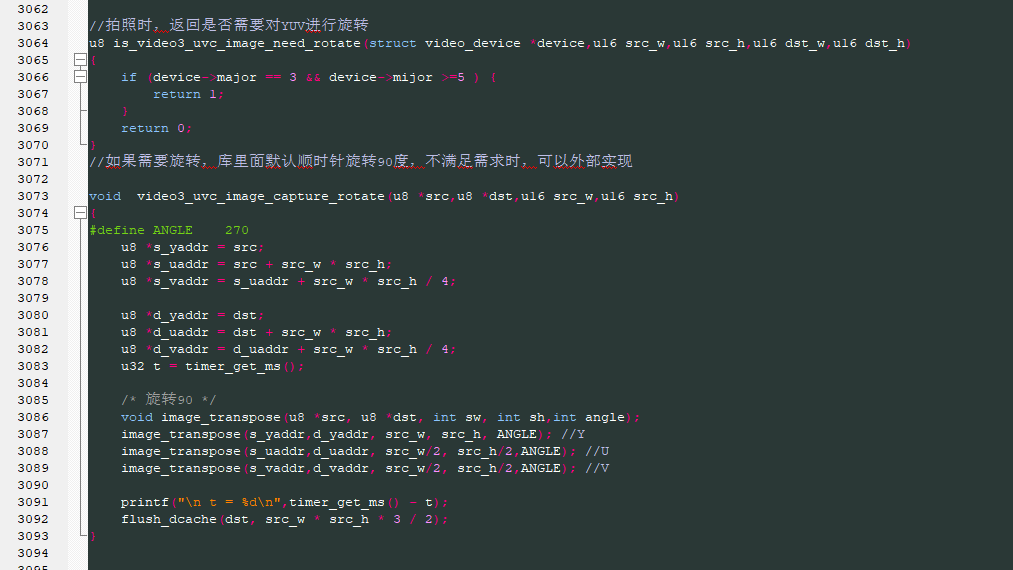
1. **关于拍照功能说明和旋转说明**

以下这个测试函数完成录像中抓拍功能，抓拍功能即拍照需要先启动录像。



注：受限于篇幅，请在sdk中查看该函数

以下两个函数，客户可以根据需求对抓拍是否需要进行软件旋转进行自定义。

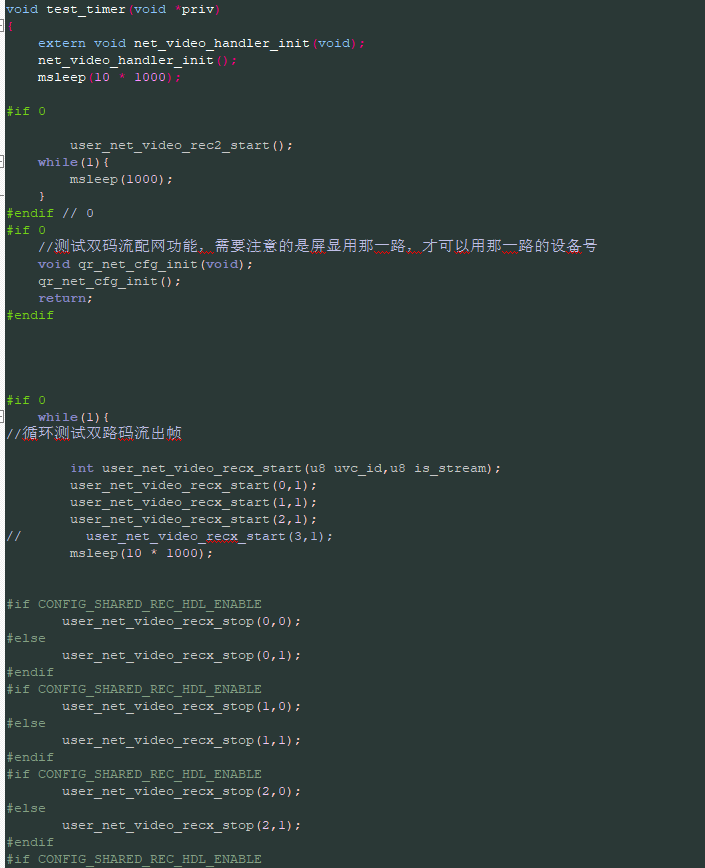


注：如果是接双h264的uvc设备，由于AC57只有一个H264解码模块，导致只能解码一路H264的码流，而拍照功能需要先把H264解码为YUV，然后再进行拍照，导致第一路和第二路的拍照功能都需要依赖于屏显这一路。

1. **关于采用原有测试代码进行功能验证说明**



测试功能代码说明：



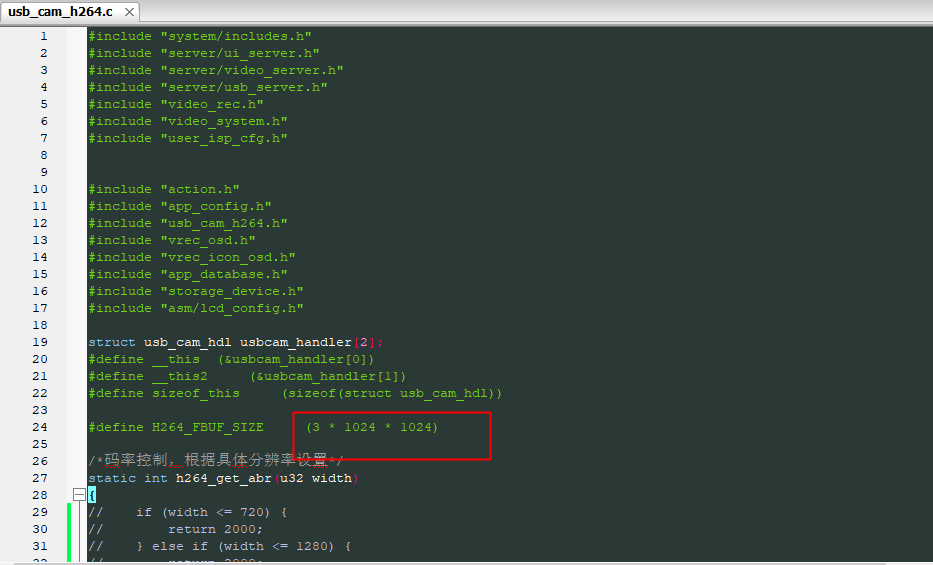
注：由于篇幅有限，更多内容请参照sdk。

1. 常见问题
2. **uac声音异常**

如果app声音异常，多数是由于从机采样率不对导致，建议从机采样率只留下一个16000采样率即可。

1. **开启旋转后，内存不足优化**

减小录像帧buf的大小，到内存足够使用，但是要客户评估测试好，一般码流大的那一路buf设置大一点，码率小的那一路帧率设置小一点

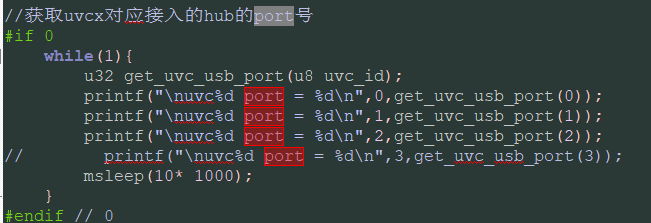


1. **主辅摄像头区分问题**

一般hub芯片会按照port0-portx或者portx-port0的顺序依次进行枚举，但是这个方式存在一定兼容性问题，需要客户根据不同hub芯片自行确定是否具备这种特性。

为了解决不同hub上线顺序兼容性问题，我们提供了usb底层获取port编号的接口，可以获取uvc摄像头是接在hub的port上，方便客户区分主辅摄像头。

数字0、1、2依次代表uvc0、uvc1、uvc2，如果uvc0和uvc1组成双码流的话，获取到的port号是一样的。



珠海市杰理科技股份有限公司

2023年06月26日