

CVBS-VGA 视频流播放服务客户端软件详细设计说明书

北京海兰信数据科技股份有限公司

二零二零年一月

CVBS-VGA 视频流播放服务客户端软件详细设计说明书

1. 适用范围
 - 1.1 标识
 - 1.2 概述
 - 1.3 术语和缩略语
 - 1.4 引用文档
2. 运行环境
 - 2.1 硬件环境
 - 2.2 软件环境
 - 2.3 网络环境
 - 2.4 编译语言
3. 概要设计
 - 3.1 软件结构
 - 3.1.1 CVBS模块
 - 3.1.2 VGA_DVI模块
 - 3.1.3 Server701
 - 3.1.4 VideoClient
 - 3.2 报文结构
 - 3.2.1 CVBS图像数据报文
 - 3.2.2 VGA_DVI
 - 3.2.3.1 视频数据
 - 3.2.3 Server701
 - 3.2.4 VideoClient
4. 详细设计
 - 4.1 CVBS详细设计
 - 4.2 VGA_DVI详细设计
 - 4.3 Server701详细设计
 - 4.3.1 TCP相关服务模块
 - 4.3.2 UDP相关服务模块
 - 4.3.3 视频数据处理业务模块
 - 4.3.4 配置文件处理模块
 - 4.4 VideoClient详细设计
 - 4.4.1 视频数据解码模块
 - 4.4.2 TCP客户端模块
 - 4.4.3 UDP相关模块
 - 4.4.4 视频数据处理业务模块
 - 4.4.5 视频播放业务模块
 - 4.4.6 视频客户端模块
5. 流程图、时序图
 - 5.1 视频播放时序图

文档更改记录

版本号	更改描述	更改人	更改日期	备注
v1.0	首次编制	张鹏	2020.01.09	

1. 适用范围

本文档是CVBS-VGA 视频流播放服务客户端软件详细设计方案报告，用于描述和论证该产品的设计方案

本文档的预期读者为客户、公司高层、部门经理、项目经理，研发人员，测试人员等。

1.1 标识

- 标识：文档编号 1.0.0.0
- 标题：CVBS-VGA视频流播放服务与客户端软件详细设计说明

1.2 概述

软件接收CVBS模块和VGA模块发来的H.264压缩视频信息。

本软件用于接收cvbs或vga视频流，具备解码播放的功能。

软件主要分为两大模块：server701，VideoClient

1.3 术语和缩略语

术语和缩略语	定义
CVBS	CVBS信号设置接收模块
VGA_DVI	视频图形阵列
Server701	数字视频接口
VideoClient	视频数据解码播放模块
ffmpeg	开源视频解码库

- CVBS
CVBS 是被广泛使用的标准，也叫做基带视频或RCA视频，是(美国)国家电视标准委员会（NTSC）电视信号的传统图像数据传输方法，它以模拟波形来传输数据。复合视频包含色差（色调和饱和度）和亮度（光亮）信息，并将它们同步在消隐脉冲中，用同一信号传输

- VGADVI

VGA（Video Graphics Array）[视频图形阵列](#)是[IBM](#)于1987年提出的一个使用[模拟信号](#)的电脑显示标准。

- DVI

DVI接口是由1998年9月，在Intel开发者论坛上成立的数字显示工作小组（Digital Display Working Group简称DDWG）发明的一种用于高速传输数字信号的技术，有DVI-A、DVI-D和DVI-I三种不同类型的接口形式。DVI-D只有数字接口，DVI-I有数字和模拟接口，目前应用主要以DVI-D(24+1)为主。

1.4 引用文档

视频服务器设计方案

2. 运行环境

2.1 硬件环境

TODO:补充硬件环境

2.2 软件环境

中标麒麟系统

2.3 网络环境

通信协议：TCP/IP

网络：TODO：待补充

2.4 编译语言

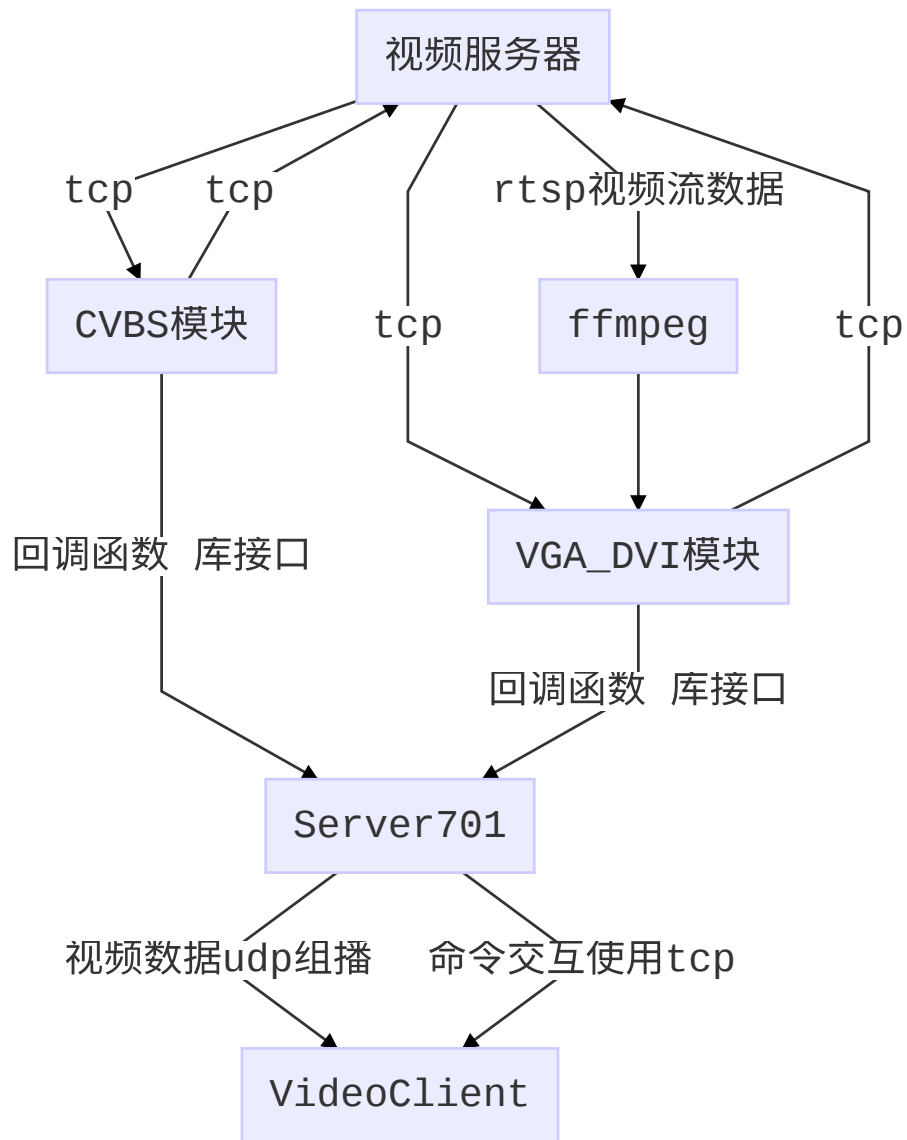
QT C++

3. 概要设计

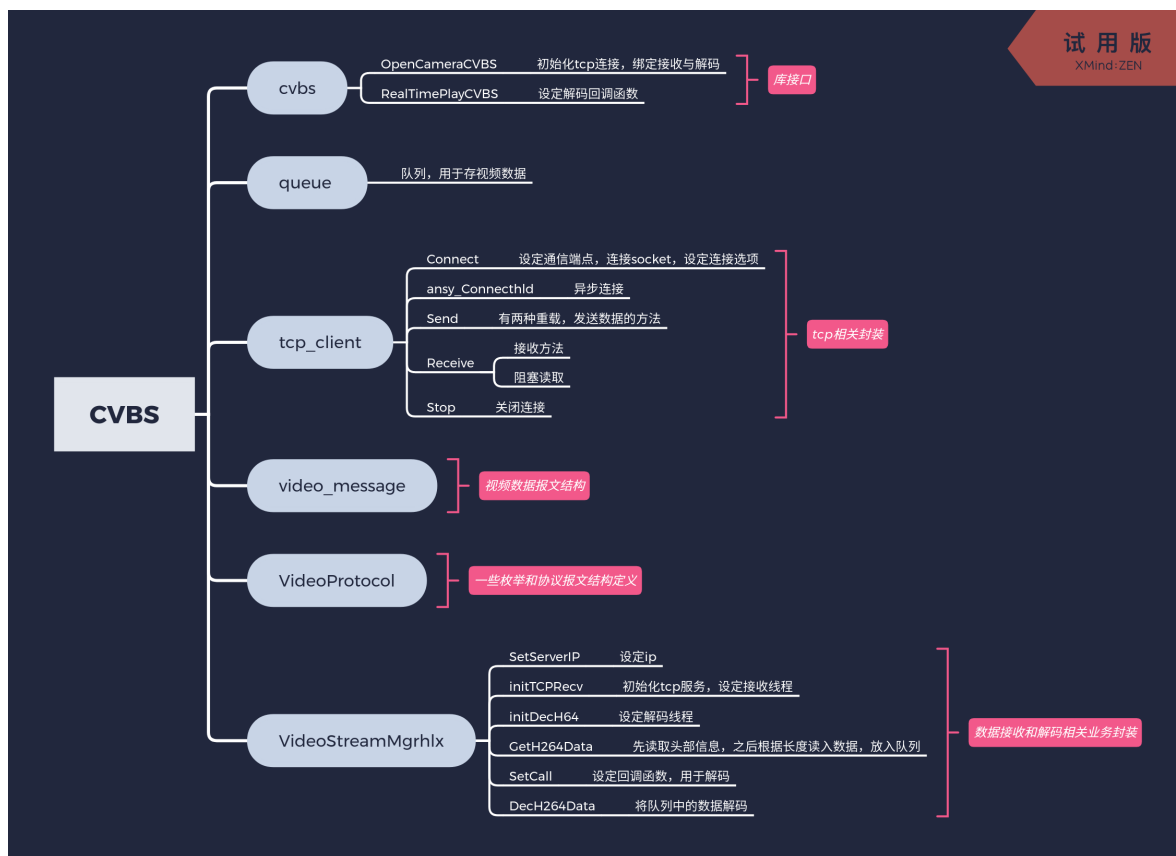
本节提供系统高层体系上的结构和功能分解，关注于系统分解到软件单元（包括子系统和组件），而不关注于单一组件的内部细节。

3.1 软件结构

视频服务器软件由，VideoClinet视频数据解码播放模块、Server701视频数据接收发送模块、CVBS信号接收设置模块、VGA_DVI信号设置接收模块组成



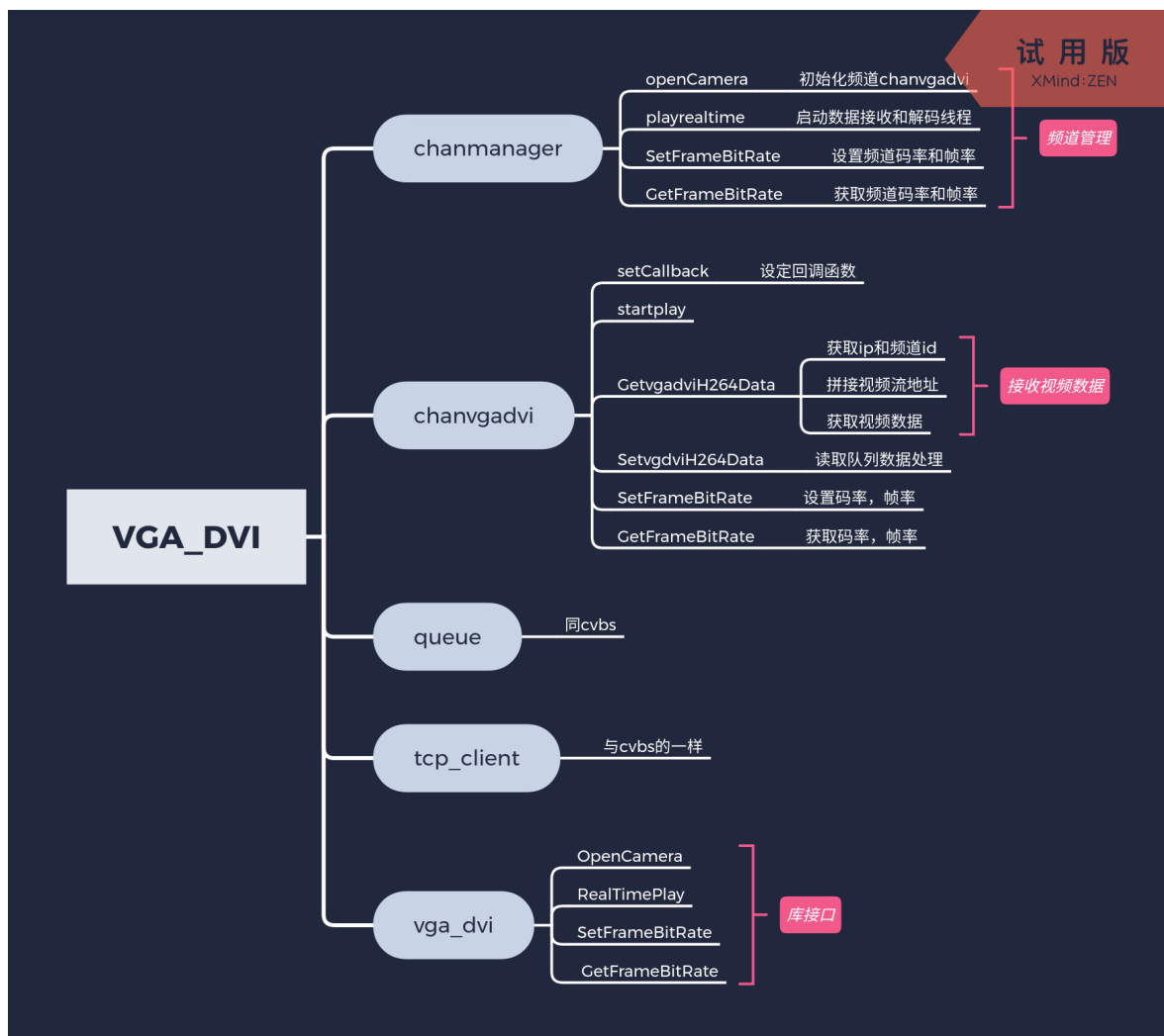
3.1.1 CVBS模块



功能描述

模块名称	功能描述
cvbs	对外库接口定义
queue	线程安全队列
tcp_client	tcp封装模块，用来与卡板交互
video_message	视频数据报文结构定义
VideoProtocol	一些枚举或报文结构定义
VideoStreamMgrhlx	视频数据接收和解码相关业务封装

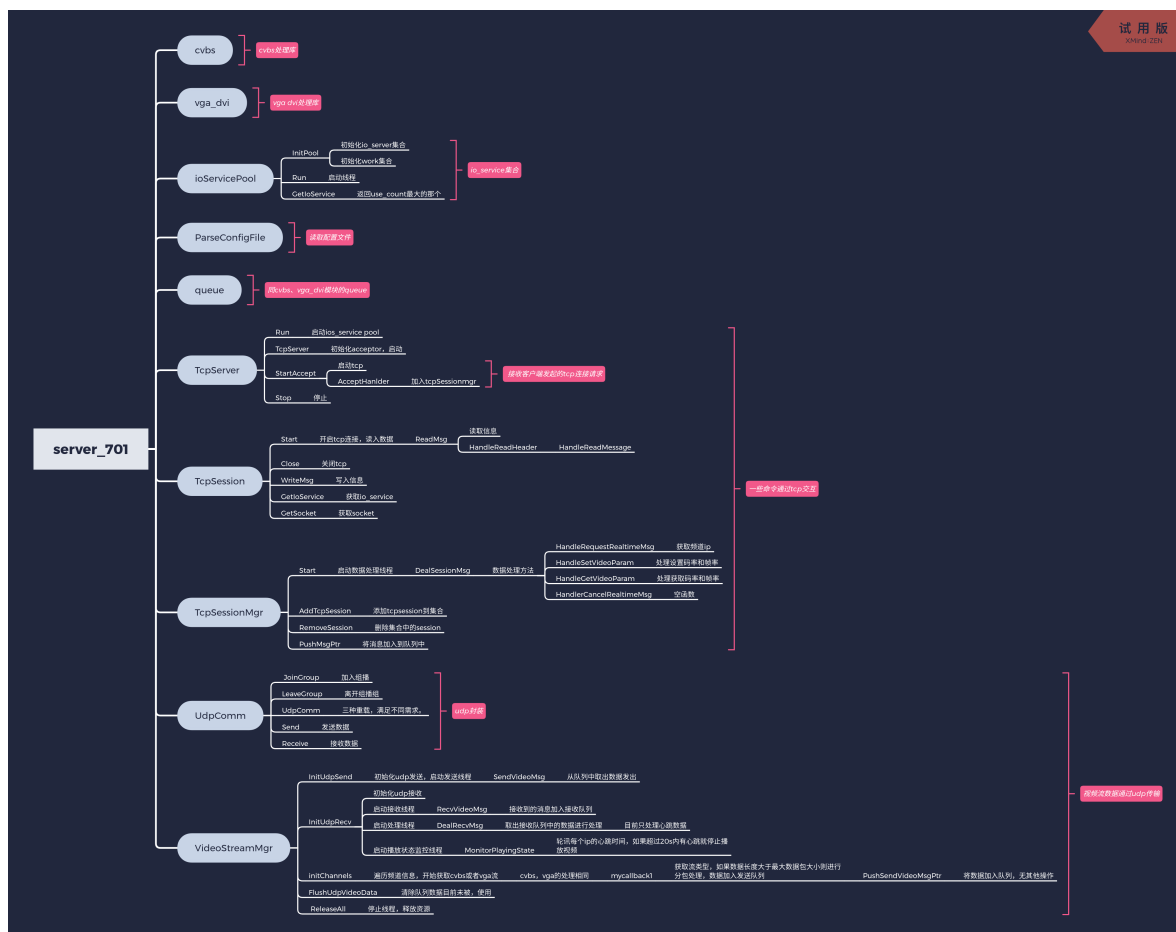
3.1.2 VGA_DVI模块



功能描述

模块名称	功能描述
chanmanager	频道管理（初始化，启动视频数据接收解码，设置码率和帧率等）
chanvgadvi	底层业务封装（chanmanager依赖此模块）
queue	线程安全队列
tcp_client	tcp封装模块，用来与卡板交互
vga_dvi	对外库接口定义

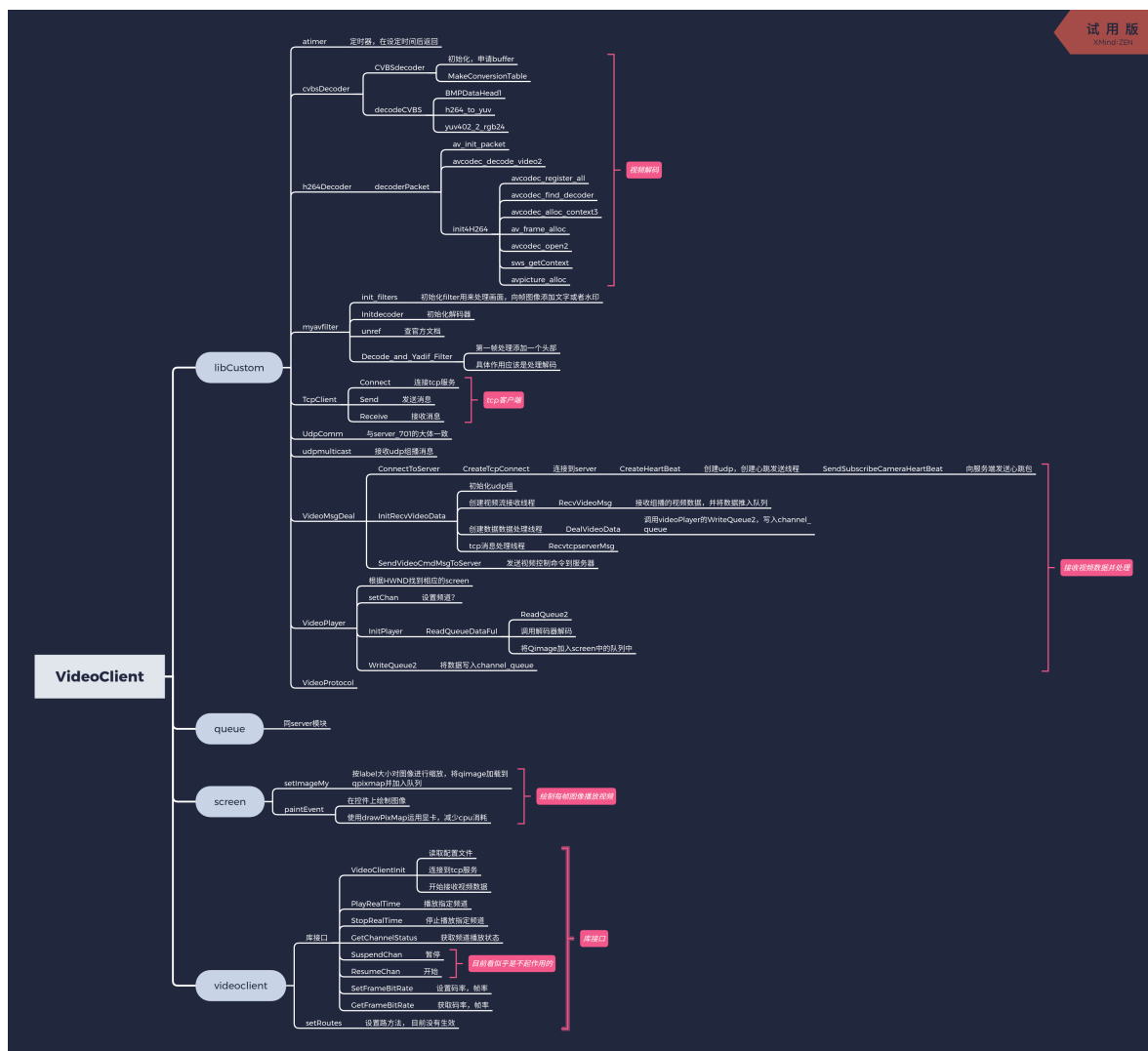
3.1.3 Server701



功能描述

模块名称	功能描述
ioServicePool	boost io_service集合封装，tcp，udp相关io的业务依赖于此
ParseConfigFile	读取xml配置文件（包含视频服务器ip，主播地址，各频道信息，视频类型，频道id等等）
queue	线程安全模块
TcpServer	tcp服务，监听客户端发来的连接请求
TcpSession	tcp业务的一些封装，TcpServer依赖于此
TcpSessionMgr	tcpSession管理，包含接收客户端数据之后的业务处理
UdpComm	udp封装
VideoStreamMgr	视频数据接收发送，负责从卡板接收视频数据并发送给客户端的业务

3.1.4 VideoClient



功能描述

模块名称	功能描述
atimer	定时器
cvbsDecoder	cvbs解码
h264Decoder	h264解码模块（解码依赖于ffmpeg）Client
myavfilter	用来处理视频画面（依赖于ffmpeg）
TcpClient	tcp客户端
UdpComm	udp封装
udpmulticast	udp组播业务，用于接收udp组播消息
VideoMsgDeal	视频流相关业务
VideoPlayer	视频数据解码业务
VideoProtocol	一些枚举和数据结构定义
queue	线程安全队列
screen	将解码的帧绘制到控件实现视频播放
videoclient	对外库接口

3.2 报文结构

3.2.1 CVBS图像数据报文

传输方式: tcp传输

报文结构:

```
{
    unsigned char ulcheck[8]; //信息头 1-8
    unsigned char packType;   //报文标识 9
    unsigned char nLength[4]; //数据包长度 10-13
    unsigned char nSentIP[4]; //发送方ip 14-17
    unsigned char uPort[4];   //侦听端口号 18-21
    unsigned char nChannelCode[4]; //数据通道编码 22-25
    unsigned char *pData;     // 数据
}
```

CVBS模块接收自卡板的视频数据结构

3.2.2 VGA_DVI

传输方式:tcp

报文结构:

```
{
    int m_type; //消息类型
    int buf[10]; //数据
    int buf_len; //数据长度
}
```

用来设置码率，帧率

3.2.3 Server701

3.2.3.1 视频数据

传输方式: udp组播

报文结构

```
{
    int channel; //频道号 0-4
    char type;   //类型 1cvbs; 2 VGA
    int serial_num; //序列号 组包时用，防止乱序
    int total_size;
    int offset_size;
    int cur_size; //视频流数据大小
    char data[1]; //视频流数据
}
```

3.2.4 VideoClient

TODO:目前未发现重要的包文结构

4. 详细设计

4.1 CVBS详细设计

- 设计需求

与视频服务器建立tcp连接，并接收视频数据。外部调用者可传入回调函数来处理接收到的视频数据。

- 功能实现

设计有两个API：

- OpenCameraCVBS(char *ip, unsigned channel)
初始化与服务器的tcp连接，开始接收数据，如果为设置回调函数，那么接收到的数据将不进行处理
- RealTimePlayCVBS(REALDATACALLBACK_HLXMB cb)
传入回调函数，开始处理视频数据

- 文件输出

序号	类型	名称	备注
1	cvbs声明文件	cvbs.h	头文件
2	cvbs定义文件	cvbs.cpp	实现文件
3	queue 线程安全队列声明文件	queue.hpp	头文件
4	tcp_client声明文件	tcp_client.h	头文件
5	tcp_client定义文件	tcp_client.cpp	实现文件
6	video_message声明文件	video_message.hpp	头文件
7	VideoProtocol声明文件	VideoProtocol.h	头文件
8	VideoStreamMgrhlx声明文件	VideoStreamMgrhlx.h	头文件
9	VideoStreamMgrhlx定义文件	VideoStreamMgrhlx.cpp	实现文件

4.2 VGA_DVI详细设计

- 设计需求

利用ffmpeg接收VGA_DVI卡板rtsp协议视频流。外部调用者可利用回调函数处理接收到的视频数据

- 功能实现

设计有4个API：

- OpenCamera(string ip, unsigned int channel);
初始化频道
- RealTimePlay(Handle h, void *pUser, REALDATACALLBACK);
启动视频流数据接收线程，与视频数据处理线程，线程之间利用队列传递视频数据

3. SetFrameBitRate(Handle h, int bit_rate, int frame_rate)

设置码率和帧率

4. GetFrameBitRate(Handle h, int& bit_rate, int& frame_rate)

获取码率和帧率

• 文件输出

序号	类型	名称	备注
1	chanmanager声明文件	chanmanager.h	头文件
2	chanmanager定义文件	chanmanager.cpp	实现文件
3	chanvgadvi声明文件	chanvgadvi.h	头文件
4	chanvgadvi定义文件	chanvgadvi.cpp	实现文件
5	queue声明文件	queue.hpp	头文件
6	tcp_client声明文件	tcp_client.h	头文件
7	tcp_client定义文件	tcp_client.cpp	实现文件
8	vga_dvi声明文件	vag_dvi.h	头文件
9	vga_dvi_global声明文件	vga_dvi.cpp	定义文件
10	vga_dvi_global声明文件	vga_dvi_global.h	头文件

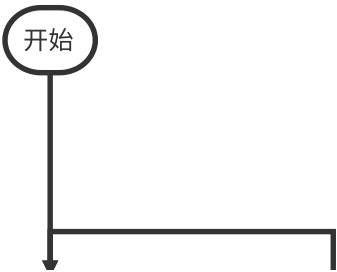
4.3 Server701详细设计

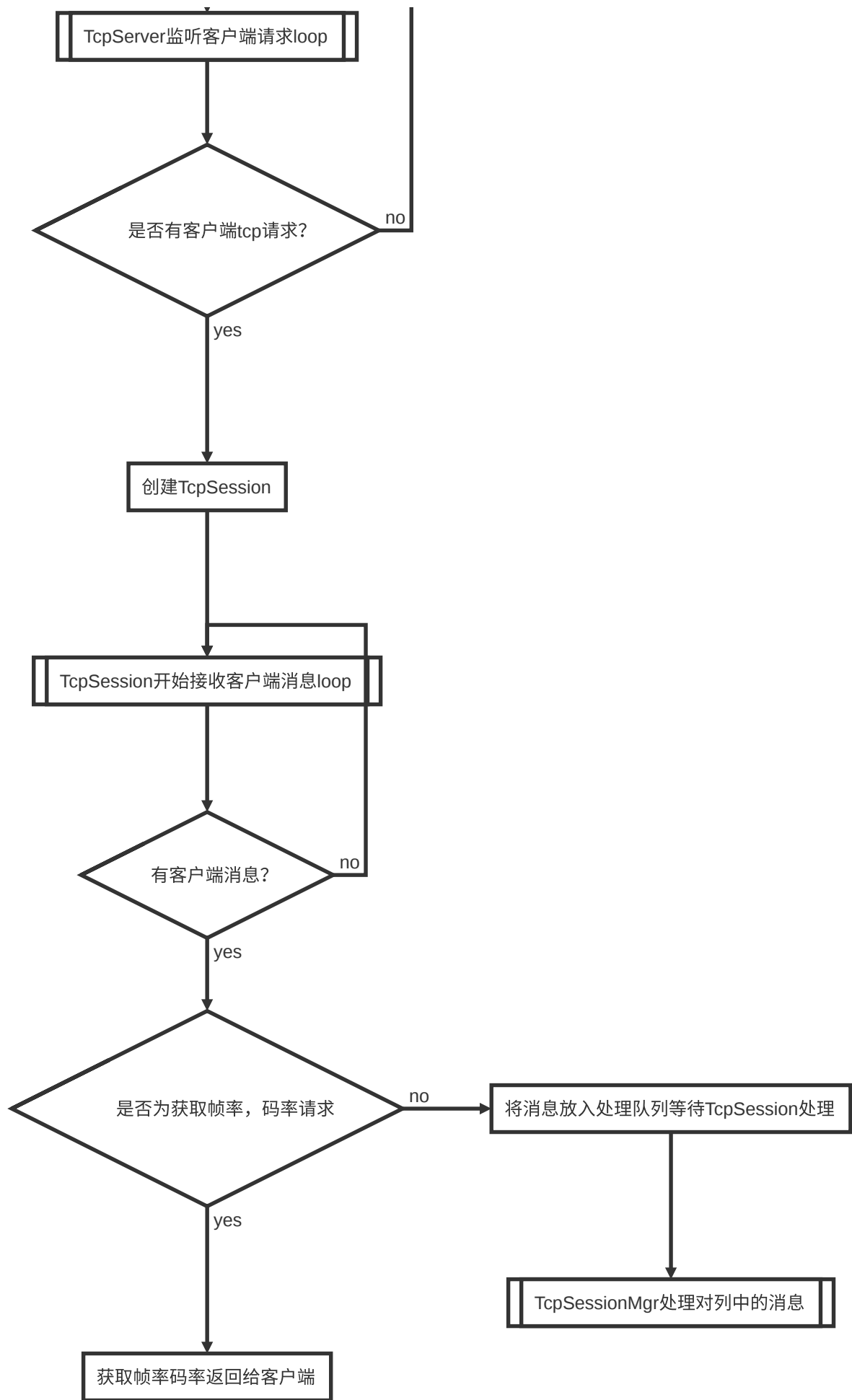
4.3.1 TCP相关服务模块

此模块包含：

- TcpServer: tcp服务端模块，监听客户端发来的连接请求
- TcpSessionMgr: tcp连接管理与业务处理模块
- TcpSession: tcp请求接收与一部分业务处理模块（码率和帧率的获取请求直接本模块处理不交由TcpSessionMgr）
- IoServicePool: 异步服务封装类

流程图





4.3.2 UDP相关服务模块

此模块包含：

- UdpCommon：对udp服务进行封装，包含组播相关服务

4.3.3 视频数据处理业务模块

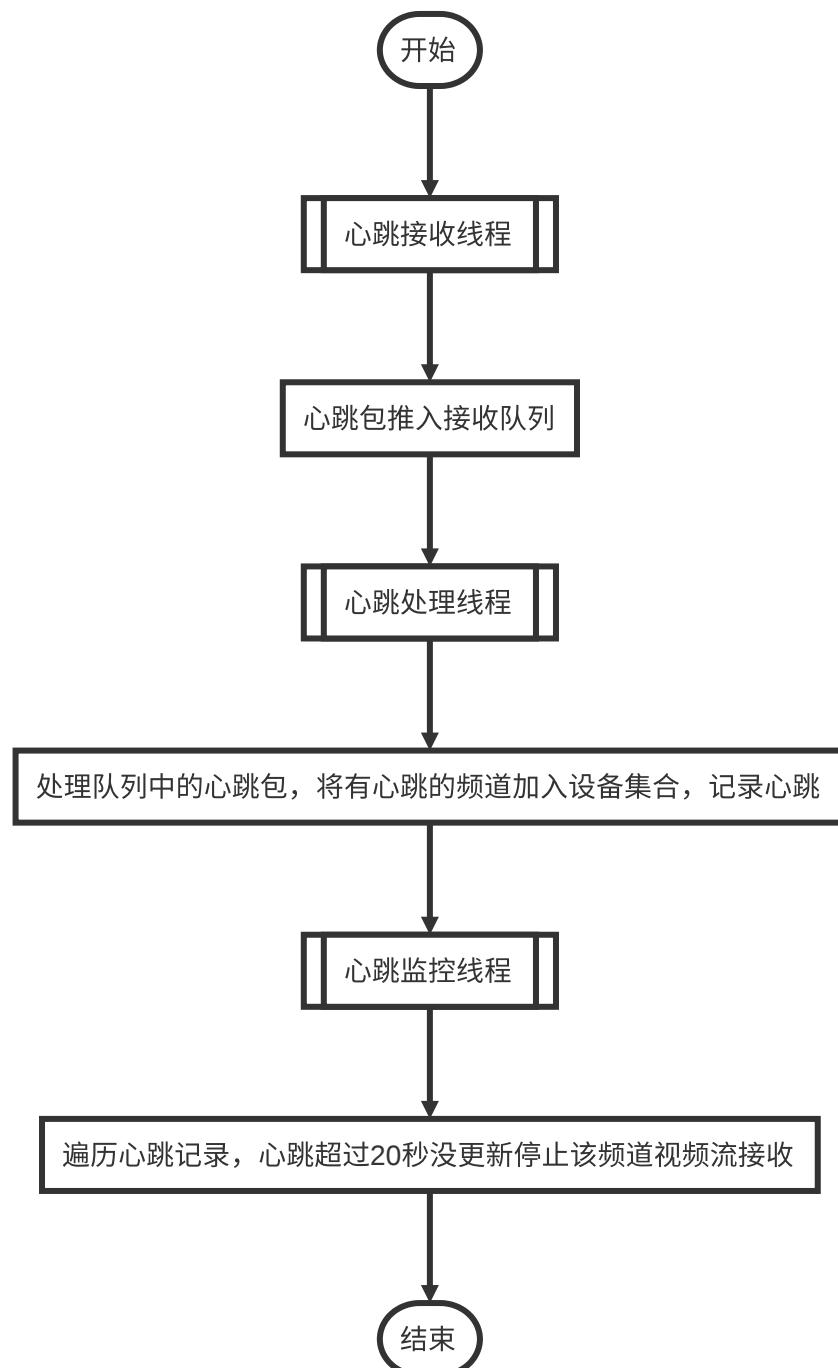
此模块包含：

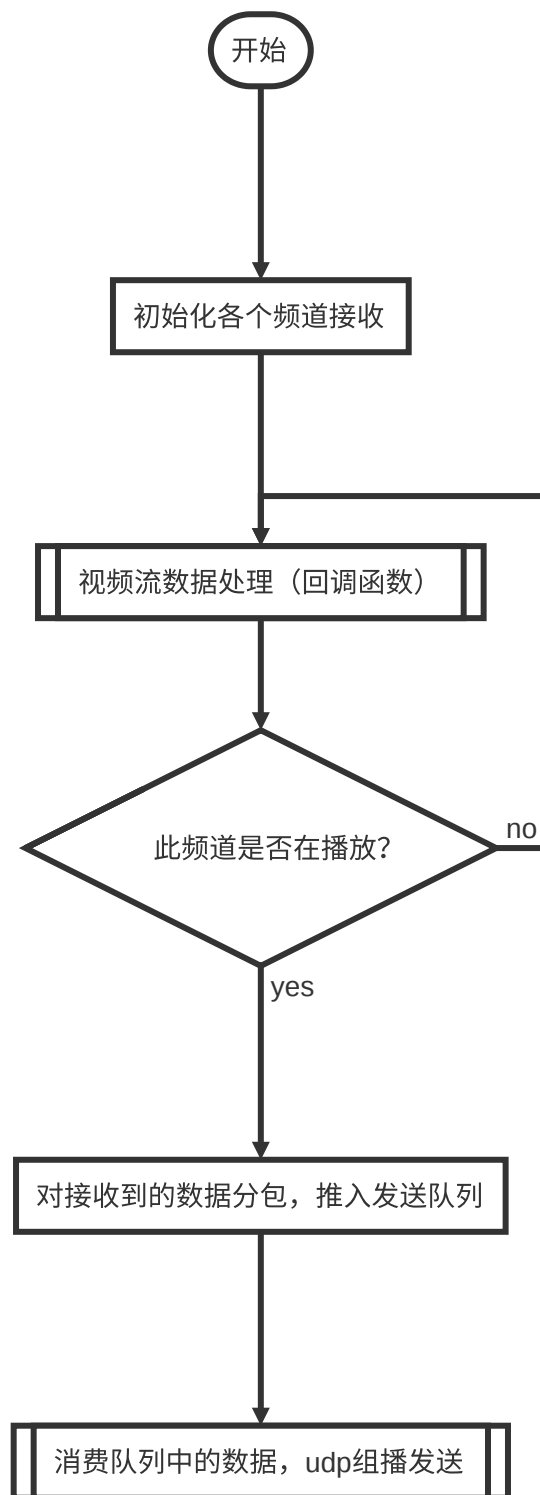
- VideoMessage：视频数据对象用于接收视频数据
- VideoStreamMgr：负责视频数据处理业务，包含从视频服务器接收视频数据与给客户端发送视频数据。还有客户端心跳相关服务。

流程图

模块运作时会先启动视频：流数据发送线程，心跳数据接收线程，初始化频道，每个频道接收用统一回调函数，回调函数根据播放列表来决定是否接收视

心跳流程





4.3.4 配置文件处理模块

此模块包含：

pugixml：xml处理依赖库

ParseConfigFile：配置文件分析模块

配置文件

xml格式,内容如下

```
<?xml version="1.0" encoding="gb2312">jiekou
<Config name="video_config">
<Adress>
    <VideoServerAddr ip="" udp_port="" tcp_port=""/>#视频服务器地址, udp端口, tcp端口
    <MulticastAddr ip="" port="" iplocal="" ttl=""/>#组播地址, 端口号, 本地ip, 时延
</Adress>
<Video>
    <channel id="" protocol="" ip="" ch=""></channel>#频道序号, 类型1cvbs、2vga, 采集卡ip, 采集卡通道号
</Video>
</Config>
```

4.4 VideoClient详细设计

4.4.1 视频数据解码模块

此模块包含:

- cvbsDecoder: 负责解码数据将视频帧并由yuv转换为rgb, 生成Qimage, 此模块依赖于myavfilter
- h264Decoder: 负责将视频数据解码(利用ffmpeg), 生成Qimage
- myavfilter: 利用ffmpeg对视频帧进行解码和特效处理,

cvbs和vga解码的区别

由于cvbs和VGA两个不同厂家提供的采集卡, CVBS厂家有自己的私有协议, h264视频数据有自己独有头部信息, VGA厂家采用的是标准的视频传输协议, 故二者的解码处理不一样

4.4.2 TCP客户端模块

此模块包含:

- TcpClient: 负责与server_701的tcp连接, videoClient与Server701的指令交互通过tcp

4.4.3 UDP相关模块

baohan此模块包含:

- udpCommon: 负责udp连接, 主要用来发送心跳包
- udpmulticast: 负责接收udp组播数据

4.4.4 视频数据处理业务模块

此模块包含:

- VideoMsgDeal: 负责发送udp心跳包, 发送和接收tcp消息, 接收udp视频数据

4.4.5 视频播放业务模块

此模块包含:

- VideoPlayer: 负责将接收到的udp视频数据进行组包, 解码视频数据
- screen: 负责将每帧图片绘制到控件上进行视频播放

4.4.6 视频客户端模块

此模块包含：

- VideoClient：定义了9个API接口
 1. VideoClientInit(int iMaxchan);
初始化udp视频数据接收，建立与server701的tcp连接
 2. PlayRealTime(const int channel, HWND hwnd);
创建videoPlayer实例，接收并播放指定频道的视频
 3. StopRealTime(int channel);
停止播放指定频道视频
 4. GetChanStatus(int channel);
获取指定频道的播放状态
 5. SuspendChan(int channel);
暂停播放
 6. ResumeChan(int channel);
恢复播放
 7. SetFrameBitRate(int channel, int bit_rate, int frame_rate);
设置帧率码率
 8. GetFrameBitRate(int channel, int& bit_rate, int& frame_rate);
获取帧率，码率
 9. GetFrameBitRateCalc(int channel, int& frame_rate)
获取准确的帧率，码率

5. 流程图、时序图

5.1 视频播放时序图

