JXJ CU代码设计

**目录**

[**一、jxj cu示意图 2**](#_Toc381015357)

[**二、jxj cu 代码模块 3**](#_Toc381015358)

[2.1 IPNC模块 3](#_Toc381015359)

[2.1.1 jxj\_cu\_ipnc\_mng.c文件源码解析 3](#_Toc381015360)

[2.1.1.1 连接控制 3](#_Toc381015361)

[2.1.1.1.1 ipnc连接管理结构体 3](#_Toc381015362)

[2.1.1.1.2 jxj\_cu\_ipnc\_connect连接函数 4](#_Toc381015363)

[2.1.1.1.3 jxj\_cu\_ipnc\_disconnect断开连接函数 4](#_Toc381015364)

[2.1.1.1.4 其他函数 5](#_Toc381015365)

[2.1.1.2 搜索控制 7](#_Toc381015366)

[2.1.1.2.1 jxj\_cu\_ipnc\_start\_search开始搜索 7](#_Toc381015367)

[2.1.1.2.2 jxj\_cu\_ipnc\_stop\_search停止搜索 7](#_Toc381015368)

[2.1.1.2.3 jxj\_cu\_ipnc\_get\_search获得搜索 8](#_Toc381015369)

[2.1.1.2.4 jxj\_cu\_ipnc\_set\_search设置搜索 8](#_Toc381015370)

[2.1.2 jxj\_cu\_ipnc\_logon.c文件源码解析 9](#_Toc381015371)

[2.1.2.1 登录控制 9](#_Toc381015372)

[2.1.2.1.1 jxj\_cu\_ipnc\_logon 登录函数 9](#_Toc381015373)

[2.1.2.1.2 jxj\_cu\_ipnc\_logout 注销登录函数 9](#_Toc381015374)

[2.1.2.1.3 jxj\_cu\_ipnc\_test\_logon登录测试函数 10](#_Toc381015375)

[2.1.2.2参数控制 10](#_Toc381015376)

[2.1.2.2.1 jxj\_cu\_ipnc\_get\_param参数获取函数 10](#_Toc381015377)

[2.1.2.2.2 jxj\_cu\_ipnc\_set\_param参数设置函数 11](#_Toc381015378)

[2.1.2.3 PTZ控制 12](#_Toc381015379)

[2.1.2.3.1 jxj\_cu\_ipnc\_ptz\_ctrl 云台控制函数 12](#_Toc381015380)

[2.1.3 jxj\_cu\_ipnc\_realplay.c文件源码解析 12](#_Toc381015381)

[2.1.3.1 取流控制 12](#_Toc381015382)

[2.1.3.1.1 jxj\_cu\_ipnc\_start\_get\_stream开始取流函数 12](#_Toc381015383)

[2.1.3.1.2 jxj\_cu\_ipnc\_stop\_get\_stream停止取流函数 13](#_Toc381015384)

[2.1.3.1.3 jxj\_cu\_ipnc\_request\_iframe 请求I帧 13](#_Toc381015385)

[2.2 搜索模块 14](#_Toc381015386)

[2.2.1搜索结构体 14](#_Toc381015387)

[2.2.2搜索初始化 15](#_Toc381015388)

[2.2.2.1 jxj\_cu\_search\_init初始化函数 15](#_Toc381015389)

[2.2.2.2 jxj\_cu\_search\_uninit退出函数 15](#_Toc381015390)

[2.2.3开始搜索 15](#_Toc381015391)

[2.2.4停止搜索 16](#_Toc381015392)

[2.2.5获取搜索 16](#_Toc381015393)

[2.2.6搜索配置 17](#_Toc381015394)

[2.3 流模块 17](#_Toc381015395)

[2.3.1 capture\_rtsp\_data.c文件源码解析 18](#_Toc381015396)

[2.3.1.1 RTSP回调接口 18](#_Toc381015397)

[2.3.1.2 RTSP控制结构体 18](#_Toc381015398)

[2.3.1.3 初始化函数 19](#_Toc381015399)

[2.3.1.4 open\_rtsp\_client连接函数 19](#_Toc381015400)

[2.3.1.5 close\_rtsp\_client关闭函数 20](#_Toc381015401)

[2.3.1.6 check\_rtsp\_client取流状态检测函数 20](#_Toc381015402)

[2.3.1.7 retry\_rtsp\_client取流重连函数 21](#_Toc381015403)

[2.3.1.8 request\_iframe\_rtsp\_client请求Ｉ帧 21](#_Toc381015404)

[2.3.2 jxj\_cu\_preview.c文件源码解析 22](#_Toc381015405)

[2.3.2.1 取流结构体 22](#_Toc381015406)

[2.3.2.2 jxj\_cu\_start\_write\_frame开始写帧函数 22](#_Toc381015407)

[2.3.2.3 jxj\_cu\_stop\_write\_frame停止写帧函数 23](#_Toc381015408)

[2.3.2.4 jxj\_cu\_get\_request\_iframe 连接状态检测函数 23](#_Toc381015409)

[2.4 登录模块 24](#_Toc381015410)

[2.4.1登录控制 24](#_Toc381015411)

[2.4.1.1 登录结构体 24](#_Toc381015412)

[2.4.1.2 jxj\_cu\_logon 登录函数 24](#_Toc381015413)

[2.4.1.3 jxj\_cu\_logout 注销登录函数 25](#_Toc381015414)

[2.4.1.4 jxj\_cu\_cfg\_get 获取参数函数 25](#_Toc381015415)

[2.4.1.4 jxj\_cu\_cfg\_set 设置参数函数 26](#_Toc381015416)

[2.4.1.5 jxj\_cu\_logon\_test 登录测试函数 27](#_Toc381015417)

[2.4.2参数控制 27](#_Toc381015418)

[2.4.1.1 jxj\_cu\_param.c文件解析 27](#_Toc381015419)

[2.4.1.1 jxj\_cu\_net\_param.c文件解析 28](#_Toc381015420)

[2.4.3 PTZ控制 28](#_Toc381015421)

[2.4.3.1 jxj\_cu\_ptz\_set 发送ptz控制函数 28](#_Toc381015422)

[2.4.3.2 jxj\_cu\_client\_crtl\_ptz 控制ptz函数 29](#_Toc381015423)

[2.4.3.3 jxj\_cu\_client\_ ptz\_cruise 自动巡航设置函数 29](#_Toc381015424)

# 一、jxj cu示意图





# 二、jxj cu 代码模块

## 2.1 IPNC模块

Ipnc模块包括**连接控制**、**搜索控制**、**登录控制**、**参数控制、PTZ控制、取流控制**, 源码分别位于jxj\_cu/src目录下的jxj\_cu\_ipnc\_mng.c、jxj\_cu\_ipnc\_logon.c、jxj\_cu\_ipnc\_realplay.c这三个文件.

### 2.1.1 jxj\_cu\_ipnc\_mng.c文件源码解析

该文件主要负责**连接控制(链表维护)**与**搜索控制.**

#### 2.1.1.1 连接控制

##### 2.1.1.1.1 ipnc连接管理结构体

typedef struct \_JXJ\_CU\_PROTOCOL\_MNG\_

{

int sock; //套接字

char ipAddr[16]; //ip地址

int port; //端口

char user[32]; //用户名

char passwd[32]; //密码

int status; //状态

int isLogon; //登录标记

int refCount; //连接计数

void \*stream[2]; //存放主次码流句柄. 0主码流, 1次码流

int poolCh; //avpool通道号

struct \_JXJ\_CU\_PROTOCOL\_MNG\_ \*next;

}JXJ\_CU\_PROTOCOL\_MNG;

##### 2.1.1.1.2 jxj\_cu\_ipnc\_connect连接函数

**函数原型**：

int jxj\_cu\_ipnc\_connect(int poolCh, const char \*ipAddr, int port,

const char \*user, const char \*passwd, long \*ipncHandle);

**功能说明**：

为每一个连接jxj cu协议的客户端分配一个JXJ\_CU\_PROTOCOL\_MNG结构体,

该结构体地址会被存放到参数ipncHandle中, 并将新分配的结构体添加到全局链表中.

**参数说明**：

**poolCh：**avpool通道号

**ipAddr：**ip地址

**port：** 端口

**user：** 用户

**passwd：**密码

**ipncHandle：**存放ipnc句柄地址

**返回值**：

成功返回0, 失败返回-1.

**代码解析**：详见**jxj\_cu\_ipnc\_connect**函数源码与注释.

##### 2.1.1.1.3 jxj\_cu\_ipnc\_disconnect断开连接函数

**函数原型**：

int jxj\_cu\_ipnc\_disconnect(long ipncHandle);

**功能说明**：

断开jxj cu连接, 并释放句柄资源.

**参数说明**：

**ipncHandle：** ipnc句柄地址

**返回值**：

成功返回0, 失败返回-1.

**代码解析**：详见**jxj\_cu\_ipnc\_connect**函数源码与注释.

##### 2.1.1.1.4 其他函数

###### 2.1.1.1.4.1 jxj\_cu\_ipnc\_mng\_check\_handle\_valid检测函数

**函数原型**：

int jxj\_cu\_ipnc\_mng\_check\_handle\_valid(JXJ\_CU\_PROTOCOL\_MNG \*pMng);

**功能说明**：

检测pMng参数是否有效(是否存在链表中).

**参数说明**：

pMng**：** JXJ\_CU\_PROTOCOL\_MNG结构体地址

**返回值**：

成功返回1, 失败返回0.

**代码解析**：详见**jxj\_cu\_ipnc\_mng\_check\_handle\_valid**函数源码与注释.

###### 2.1.1.1.4.2 jxj\_cu\_ipnc\_mng\_check\_ipnc\_handle\_valid检测函数

**函数原型**：

int jxj\_cu\_ipnc\_mng\_check\_ipnc\_handle\_valid(JXJ\_CU\_PROTOCOL\_MNG \*pMng,

JXJ\_CU\_STREAM\_S \*pStream);

**功能说明**：

检测pMng参数是否有效(是否存在链表中), pStream值是否与pMng存放的流句柄一致.

**参数说明**：

pMng**：** JXJ\_CU\_PROTOCOL\_MNG结构体地址

pStream**：** JXJ\_CU\_STREAM\_S***(详见流模块)***结构体地址

**返回值**：

成功返回1, 失败返回0.

**代码解析**：详见**jxj\_cu\_ipnc\_mng\_check\_ipnc\_handle\_valid**函数源码与注释.

###### 2.1.1.1.4.3 jxj\_cu\_ipnc\_mng\_ref客户端计数函数

**函数原型**：

void jxj\_cu\_ipnc\_mng\_ref(JXJ\_CU\_PROTOCOL\_MNG \*pMng);

**功能说明**：

refCount计数加一.

**参数说明**：

pMng**：** JXJ\_CU\_PROTOCOL\_MNG结构体地址

**代码解析**：详见jxj\_cu\_ipnc\_mng\_ref函数源码与注释.

###### 2.1.1.1.4.4 jxj\_cu\_ipnc\_mng\_unref客户端清退函数

**函数原型**：

void jxj\_cu\_ipnc\_mng\_unref(JXJ\_CU\_PROTOCOL\_MNG \*pMng);

**功能说明**：

refCount计数减一, 若计数为零，关闭取流, 从链表删除pMng节点并释放资源.

**参数说明**：

pMng**：** JXJ\_CU\_PROTOCOL\_MNG结构体地址

**代码解析**：详见jxj\_cu\_ipnc\_mng\_unref函数源码与注释.

#### 2.1.1.2 搜索控制

##### 2.1.1.2.1 jxj\_cu\_ipnc\_start\_search开始搜索

**函数原型**：

int jxj\_cu\_ipnc\_start\_search(long lHandle);

**功能说明**：

开始搜索, 调用[*jxj\_cu\_search\_start(****详见搜索模块****)*](#_2.2.3开始搜索).

**参数说明**：

lHandle**： 句柄来自**[**jxj\_cu\_ipnc\_connect函数**](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**返回值**：

成功返回0, 失败返回-1.

**代码解析**：详见jxj\_cu\_ipnc\_start\_search函数源码与注释.

##### 2.1.1.2.2 jxj\_cu\_ipnc\_stop\_search停止搜索

**函数原型**：

int jxj\_cu\_ipnc\_stop\_search(long lHandle);

**功能说明**：

停止搜索, 调用[*jxj\_cu\_search\_stop(****详见搜索模块****)*.](#_2.2.4停止搜索)

**参数说明**：

lHandle**：** 句柄来自[jxj\_cu\_ipnc\_connect函数](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**返回值**：

成功返回0, 失败返回-1.

**代码解析**：详见jxj\_cu\_ipnc\_stop\_search函数源码与注释.

##### 2.1.1.2.3 jxj\_cu\_ipnc\_get\_search获得搜索

**函数原型**：

int jxj\_cu\_ipnc\_get\_search(long lHandle, IPNC\_NET\_SEARCH\_RES\_S \*pDev);

**功能说明**：

获得搜索, 调用[*jxj\_cu\_search\_get(****详见搜索模块****)*.](#_2.2.5获取搜索)

**参数说明**：

lHandle**：** 句柄来自[jxj\_cu\_ipnc\_connect函数](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

pDev**：** 存放搜索请求响应包地址空间

**返回值**：

成功且有设备搜索到则返回sizeof(IPNC\_NET\_SEARCH\_RES\_S);

成功且无设备搜索到则返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_ipnc\_get\_search函数源码与注释.

##### 2.1.1.2.4 jxj\_cu\_ipnc\_set\_search设置搜索

**函数原型**：

int jxj\_cu\_ipnc\_set\_search(IPNC\_NET\_SEARCH\_CFG\_REQ\_S\* pIpncSearchCfg);

**功能说明**：

搜索设置, 调用[*jxj\_cu\_search\_set(****详见搜索模块****)*.](#_2.2.6搜索配置)

**参数说明**：

pIpncSearchCfg**：** 搜索配置包

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_ipnc\_set\_search函数源码与注释.

### 2.1.2 jxj\_cu\_ipnc\_logon.c文件源码解析

该文件主要负责**登录控制、参数控制、PTZ控制.**

#### 2.1.2.1 登录控制

##### 2.1.2.1.1 jxj\_cu\_ipnc\_logon 登录函数

**函数原型**：

int jxj\_cu\_ipnc\_logon(long handle, long \*logonHandle);

**功能说明**：

登录, 调用[*jxj\_cu\_logon(****详见登录模块****)*.](#_2.4.1.2_jxj_cu_logon_登录函数)

**参数说明**：

**handle：句柄来自**[**jxj\_cu\_ipnc\_connect函数**](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**logonHandle：用于存放返回的登录句柄地址**

**返回值**：

成功返回连接建立的套接字;

失败返回-1;

**代码解析**：详见jxj\_cu\_ipnc\_logon函数源码与注释.

##### 2.1.2.1.2 jxj\_cu\_ipnc\_logout 注销登录函数

**函数原型**：

int jxj\_cu\_ipnc\_logout(long logonHandle);

**功能说明**：

注销登录, 调用[*jxj\_cu\_logout(****详见登录模块****)*.](#_2.4.1.3_jxj_cu_logout_注销登录函数)

**参数说明**：

**logonHandle：登录句柄地址**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_ipnc\_logout函数源码与注释.

##### 2.1.2.1.3 jxj\_cu\_ipnc\_test\_logon登录测试函数

**函数原型**：

int jxj\_cu\_ipnc\_test\_logon(const char\* DesIP, const char\* User,

const char\* Password, const char\* port);

**功能说明**：

登录测试, 调用[*jxj\_cu\_logon\_test(****详见登录模块****)*.](#_2.4.1.5_jxj_cu_logon_test_登录测试函数)

**参数说明**：

**DesIP：IP地址**

**User**： **用户名**

**Password**：**密码**

**Port**：**端口号**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_ipnc\_test\_logon函数源码与注释.

#### 2.1.2.2参数控制

##### 2.1.2.2.1 jxj\_cu\_ipnc\_get\_param参数获取函数

**函数原型**：

int jxj\_cu\_ipnc\_get\_param(long logonHandle, IPNC\_CMD\_TYPE\_E cmd,

void \*pParam, int dataSize);

**功能说明**：

参数获取, 根据命令码cmd 调用相应的参数获取接口(***各命令码对应的参数接口由静态变量sCmdParam数组控制，位于jxj\_cu\_net\_param.c文件***)

**参数说明**：

**logonHandle：登录句柄地址**

**cmd**： **参数操作命令码(详见枚举**IPNC\_CMD\_TYPE\_E定义**)**

**pParam**：**参数存放空间**

dataSize：**参数结构体大小**

**返回值**：

成功返回参数结构体大小;

失败返回-1;

**代码解析**：详见jxj\_cu\_ipnc\_get\_param函数源码与注释.

##### 2.1.2.2.2 jxj\_cu\_ipnc\_set\_param参数设置函数

**函数原型**：

int jxj\_cu\_ipnc\_set\_param(long logonHandle, IPNC\_CMD\_TYPE\_E cmd,

void \*pParam, int dataSize);

**功能说明**：

参数设置, 根据命令码cmd 调用相应的参数设置接口(***各命令码对应的参数接口由静态变量sCmdParam数组控制，位于jxj\_cu\_net\_param.c文件***)

**参数说明**：

**logonHandle：登录句柄地址**

**cmd**： **参数操作命令码(详见枚举**IPNC\_CMD\_TYPE\_E定义**)**

**pParam**：**参数存放空间**

dataSize：**参数结构体大小**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_ipnc\_set\_param函数源码与注释.

#### 2.1.2.3 PTZ控制

##### 2.1.2.3.1 jxj\_cu\_ipnc\_ptz\_ctrl 云台控制函数

**函数原型**：

int jxj\_cu\_ipnc\_ptz\_ctrl(long logonHandle, void \*ptzPacket);

**功能说明**：

PTZ云台控制, 具体详见相应操作码(IPNC\_PTZ\_CMD\_E).

**参数说明**：

**logonHandle：登录句柄地址**

**ptzPacket**： **PTZ操作数据包**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_ipnc\_ptz\_ctrl函数源码与注释.

### 2.1.3 jxj\_cu\_ipnc\_realplay.c文件源码解析

#### 2.1.3.1 取流控制

##### 2.1.3.1.1 jxj\_cu\_ipnc\_start\_get\_stream开始取流函数

**函数原型**：

int jxj\_cu\_ipnc\_start\_get\_stream(long handle, int chn,

int streamNo, long \*pStreamHandle)

**功能说明**：

开始取流, 调用jxj\_cu\_start\_write\_frame函数(***详见取流模块***)

**参数说明**：

**handle：句柄来自**[**jxj\_cu\_ipnc\_connect函数**](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**chn**： **通道号**

**streamNo：码流号(0主码流, 1次码流)**

**pStreamHandle**：**存放流句柄地址空间**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_ipnc\_start\_get\_stream函数源码与注释.

##### 2.1.3.1.2 jxj\_cu\_ipnc\_stop\_get\_stream停止取流函数

**函数原型**：

int jxj\_cu\_ipnc\_stop\_get\_stream(long handle, long streamHandle)

**功能说明**：

停止取流, 调用jxj\_cu\_stop\_write\_frame函数(***详见取流模块***)

**参数说明**：

**handle：句柄来自**[**jxj\_cu\_ipnc\_connect函数**](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**streamHandle**：**流句柄**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_ipnc\_stop\_get\_stream函数源码与注释.

##### 2.1.3.1.3 jxj\_cu\_ipnc\_request\_iframe 请求I帧

**函数原型**：

int jxj\_cu\_ipnc\_request\_iframe (long handle, long streamHandle)

**功能说明**：

调用[*jxj\_cu\_get\_request\_iframe*函数(***详见取流模块***)](#_2.3.2.4_jxj_cu_get_request_iframe请求).

**参数说明**：

**handle：句柄来自**[**jxj\_cu\_ipnc\_connect函数**](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**streamHandle**：**流句柄**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_ipnc\_request\_iframe函数源码与注释.

## 2.2 搜索模块

搜索模块包括**搜索初始化**、**开始搜索**、**停止搜索**、**获取搜索**、**搜索配置**, 源码位于jxj\_cu/src目录下的***jxj\_cu\_search.c***文件.

搜索逻辑：调用**开始搜索后, 搜索接口会**先将搜索设备保存到全局链表中, 而调用获取搜索则每次只从链表中获取一个搜索设备, 直到所有的搜索设备都获取完, 则返回0. 一旦调用停止搜索则把链表清空掉.

### 2.2.1搜索结构体

**//搜索信息**

typedef struct \_JXJ\_CU\_SEARCH\_INFO\_S\_

{

J\_Device\_T devCfg; //设备信息

char dstId[J\_SDK\_MAX\_ID\_LEN]; //搜索设备标识字符串(搜索配置会用到)

char usr[J\_SDK\_MAX\_NAME\_LEN]; //用户名

char pwd[J\_SDK\_MAX\_PWD\_LEN]; //密码

}JXJ\_CU\_SEARCH\_INFO\_S;

**//搜索链表节点信息**

typedef struct \_JXJ\_CU\_SEARCH\_NODE\_S\_

{

JXJ\_CU\_SEARCH\_INFO\_S devInfo;//搜索信息

struct \_JXJ\_CU\_SEARCH\_NODE\_S\_ \*next;

}JXJ\_CU\_SEARCH\_NODE\_S;

**//用户信息表**

typedef struct \_JXJ\_CU\_SEARCH\_USER\_TABLE\_S\_

{

int id;//用户标识ID

JXJ\_CU\_SEARCH\_NODE\_S \*pLastNode;//记录结束访问的节点

}JXJ\_CU\_SEARCH\_USER\_TABLE\_S;

**//搜索控制结构体**

typedef struct \_JXJ\_CU\_SEARCH\_STRUCT\_

{

volatile long searchHld; //搜索开始句柄

volatile uint8\_t searching; //开始搜索标识

volatile int u32DevCnt; //搜索获得设备数

JXJ\_CU\_SEARCH\_NODE\_S \*pSearchNodeHead;//头节点

JXJ\_CU\_SEARCH\_NODE\_S \*pSearchNodeEnd; //尾节点

pthread\_mutex\_t lock; //操作锁

volatile int userCnt; //用户数

pthread\_mutex\_t userlock; //用户增删锁

JXJ\_CU\_SEARCH\_USER\_TABLE\_S userCT[JXJ\_CU\_SEARCH\_MAX\_CLIENT\_NUM]; //用户信息表

}JXJ\_CU\_SEARCH\_STRUCT;

### 2.2.2搜索初始化

#### 2.2.2.1 jxj\_cu\_search\_init初始化函数

**函数原型**：

int jxj\_cu\_search\_init();

**功能说明**：

初始化全局量gJxjCuSearch, 初始化操作锁与用户锁.

**代码解析**：详见**jxj\_cu\_search\_init**函数源码与注释.

#### 2.2.2.2 jxj\_cu\_search\_uninit退出函数

**函数原型**：

int jxj\_cu\_search\_uninit();

**功能说明**：

清空链表, 摧毁操作锁与用户锁.

**代码解析**：详见**jxj\_cu\_search\_uninit**函数源码与注释.

### 2.2.3开始搜索

**函数原型**：

int jxj\_cu\_search\_start(long userDefId);

**功能说明**：

开始搜索, 将userDefId作为用户标识ID增加用户，调用mb\_query(*搜索到的设备被该函数的回调函数jxj\_cu\_search\_device\_cb保存*)函数开始搜索.

**参数说明**：

**userDefId：句柄来自**[**jxj\_cu\_ipnc\_connect函数**](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_search\_start函数源码与注释.

### 2.2.4停止搜索

**函数原型**：

int jxj\_cu\_search\_stop(long userDefId);

**功能说明**：

停止搜索, 根据userDefId删除用户信息，若搜索用户数为0，则清空链表,并调用mb\_release函数释放搜索.

**参数说明**：

**userDefId：句柄来自**[**jxj\_cu\_ipnc\_connect函数**](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_search\_stop函数源码与注释.

### 2.2.5获取搜索

**函数原型**：

int jxj\_cu\_search\_get(long userDefId, IPNC\_NET\_SEARCH\_RES\_S \*pIpncRes);

**功能说明**：

获得搜索, 根据userDefId获取用户信息，并根据上次访问链表的位置获取设备信息.

**参数说明**：

**userDefId：句柄来自**[**jxj\_cu\_ipnc\_connect函数**](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**pIpncRes：IPNC搜索请求响应包**

**返回值**：

成功且有设备搜索到则返回sizeof(IPNC\_NET\_SEARCH\_RES\_S);

成功且无设备搜索到则返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_search\_get 函数源码与注释.

### 2.2.6搜索配置

**函数原型**：

int jxj\_cu\_search\_set(IPNC\_NET\_SEARCH\_CFG\_REQ\_S \*pIpncReqCfg);

**功能说明**：

搜索设置, 根据获得搜索获得的设备标识字符串dst\_id, 调用mb\_cfg\_set函数发送搜索配置包.

**参数说明**：

**pIpncReqCfg：IPNC搜索配置包**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_search\_set 函数源码与注释.

## 2.3 流模块

流模块包括**RTSP控制、取流控制**, 源码分别位于jxj\_cu/src目录下的capture\_rtsp\_data.c与jxj\_cu\_preview.c这两个文件.

### 2.3.1 capture\_rtsp\_data.c文件源码解析

capture\_rtsp\_data.c文件负责的是**RTSP控制, 与live555库接口通信.**

#### 2.3.1.1 RTSP回调接口

**函数原型**：

typedef int (\*caputure\_data)(unsigned char \*frameBuf, unsigned long frameLen,

unsigned char \*frameHdr, unsigned long hdrLen,

unsigned long frameNo, int frameType, unsigned long long pts, void \*pUserData);

**功能说明**：

RTSP流回调接口(***代码中回调接口负责将帧数据写入到avpool***).

**参数说明**：

**frameBuf: 流数据(*avpool头+帧数据*)**

**frameLen: 流数据长度(*avpool头长度+帧数据长度*)**

**frameHdr: 流数据头(*avpool头*)**

**hdrLen: 流数据头长度**

**frameNo: 帧号**

**frameType: 帧类型(AV\_FRAME\_TYPE\_E)**

**pts: pts数据**

**pUserData: 私有数据(*代码中用于存放avpool句柄ID*)**

**代码解析**：详见jxj\_cu\_preview.c文件下jxj\_write\_video\_data函数源码与注释.

#### 2.3.1.2 RTSP控制结构体

//RTSP 客户端数据 用于与live555通信

typedef struct RTSP\_CLIENT\_DATA

{

pthread\_t pthId; //线程ID

caputure\_data read\_av\_func; //流回调接口

int8\_t pthreadExitFlag; //线程退出标记

char urlBuf[URL\_LENGTH];//url

void \*pUserData; //私有数据

unsigned long buffer\_size; //缓存大小

char quit; //流控制, 非零0则取流连接断开

}RTSP\_CLIENT\_DATA,\*PRTSP\_CLIENT\_DATA;

//RTSP客户端信息

typedef struct \_RTSP\_CLIENT\_INFO\_

{

long id;

char\* url;

void\* pUserData;

}RTSP\_CLIENT\_INFO;

//RTSP客户端结构体 用于控制rtsp 客户端连接及保存每个客户端标识信息

typedef struct \_RTSP\_CLIENT\_CTRL\_

{

RTSP\_CLIENT\_INFO clientInfo[MAX\_CLIENT];//RTSP客户端信息

volatile int userCnt;//客户端数

pthread\_t pthId; //线程ID

pthread\_mutex\_t lock;//操作锁

}RTSP\_CLIENT\_CTRL;

#### 2.3.1.3 初始化函数

**函数原型**：

void rtsp\_client\_init(void);

void rtsp\_client\_uninit(void);

**功能说明**：

初始化全局量RTSP\_CLIENT\_CTRL结构体及锁.

**代码解析**：详见函数源码与注释.

#### 2.3.1.4 open\_rtsp\_client连接函数

**函数原型**：

long open\_rtsp\_client(char \*url, caputure\_data av\_func , void\* pUserData);

**功能说明**：

建立客户端信息及RTSP连接, 并注册流回调函数av\_func.

**参数说明**：

**url: url地址字符串**

**av\_func: 流回调函数(**[***详见caputure\_data说明***](#_2.3.1.1_RTSP回调接口)**)**

**pUserData: 私有数据(*代码中用于存放avpool句柄ID*)**

**返回值**：

成功返回RTSP连接句柄;

失败返回0;

**代码解析**：详见open\_rtsp\_client 函数源码与注释.

#### 2.3.1.5 close\_rtsp\_client关闭函数

**函数原型**：

void close\_rtsp\_client(long handle);

**功能说明**：

关闭RTSP连接, 释放资源.

**参数说明**：

**handle: RTSP连接句柄**

**代码解析**：详见close\_rtsp\_client 函数源码与注释.

#### 2.3.1.6 check\_rtsp\_client取流状态检测函数

**函数原型**：

int check\_rtsp\_client(long handle);

**功能说明**：

检测客户端url取流是否退出.若是连接断开quit非零.

**参数说明**：

**handle: RTSP连接句柄**

**返回值**：

返回1代表取流已断开;

返回0代表流未断开;

**代码解析**：详见check\_rtsp\_client 函数源码与注释.

#### 2.3.1.7 retry\_rtsp\_client取流重连函数

**函数原型**：

int retry\_rtsp\_client(long handle, void\* pUserData);

**功能说明**：

检测客户端url取流是否退出.若是连接断开则重新取流.

**参数说明**：

**handle: RTSP连接句柄**

**pUserData: 私有数据(*代码中用于存放avpool句柄ID*)**

**返回值**：

返回0代表成功;

返回-1代表失败;

**代码解析**：详见retry\_rtsp\_client 函数源码与注释.

#### 2.3.1.8 request\_iframe\_rtsp\_client请求I帧

**函数原型**：

int request\_iframe\_rtsp\_client(long handle);

**功能说明**：

请求I帧.最终调用live555库文件中*rtspRequestIFrame*函数.

**参数说明**：

**handle: RTSP连接句柄**

**返回值**：

返回0代表成功;

返回-1代表失败;

**代码解析**：详见request\_iframe\_rtsp\_client 函数源码与注释.

### 2.3.2 jxj\_cu\_preview.c文件源码解析

jxj\_cu\_preview.c文件负责的取流控制,开始取流、停止取流、重新取流.

#### 2.3.2.1 取流结构体

typedef struct \_JXJ\_CU\_STREAM\_S\_

{

int chn; //ipc通道号

int streamNo; //码流号(0主码流, 1次码流)

int poolChn; //avpool通道号

int poolStreamNo; //码流号

long lRealPlayHandle; //rtsp句柄

long videoId; //avpool句柄ID

int refCount; //连接计数

void \*ipncHandle; //ipnc 句柄

pthread\_t pthId; //取流重连线程

}JXJ\_CU\_STREAM\_S;

#### 2.3.2.2 jxj\_cu\_start\_write\_frame开始写帧函数

**函数原型**：

int jxj\_cu\_start\_write\_frame(long ipncHandle, int chn, int nStream,

long \*pStreamHandle);

**功能说明**：

建立avpool通道与RTSP连接, 并将信息保存到流句柄中,且将流句柄保存的到IPNC句柄中, 同时创建取流重连线程.

**参数说明**：

**ipncHandle: 句柄来自**[**jxj\_cu\_ipnc\_connect函数**](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**chn: 通道号**

**nStream: 主次码流(0主码流, 1次码流)**

**pStreamHandle: 存放流句柄空间(***JXJ\_CU\_STREAM\_S结构体***)**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_start\_write\_frame函数源码与注释.

#### 2.3.2.3 jxj\_cu\_stop\_write\_frame停止写帧函数

**函数原型**：

int jxj\_cu\_stop\_write\_frame(long ipncHandle, long streamHandle);

**功能说明**：

清除avpool通道连接与RTSP连接, 并释放流句柄,且将IPNC句柄中流信息清除.

**参数说明**：

**ipncHandle: 句柄来自**[**jxj\_cu\_ipnc\_connect函数**](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**streamHandle: 流句柄**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_stop\_write\_frame函数源码与注释.

#### 2.3.2.4 jxj\_cu\_get\_request\_iframe请求I帧函数

**函数原型**：

int jxj\_cu\_get\_request\_iframe(long ipncHandle, long streamHandle);

**功能说明**：

最终调用[request\_iframe\_rtsp\_client函数](#_2.3.1.8_request_iframe_rtsp_client请)请求I帧.

**参数说明**：

**ipncHandle: 句柄来自**[**jxj\_cu\_ipnc\_connect函数**](#_2.1.1.1.2_jxj_cu_ipnc_connect连接函数)

**streamHandle: 流句柄**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_get\_request\_iframe函数源码与注释.

## 2.4 登录模块

登录模块包括**登录控制、参数控制**、**PTZ控制**.

### 2.4.1登录控制

**登录控制源码位于src目录下jxj\_cu\_logon.c文件.**

#### 2.4.1.1 登录结构体

typedef struct \_JXJ\_CU\_LOGON\_S\_

{

volatile int quit; //线程退出标记

int sock; //套接字

int sendPipe[2]; //发送管道

int recvPipe[2]; //接收管道

int port; //端口号

char userName[32]; //用户名

char passWord[32]; //密码

int keepAliveTime; //心跳包间隔

char pAddr[64]; //ip地址

pthread\_t pid; //线程ID

pthread\_mutex\_t lock; //操作锁

int alarm; //报警：0代表不报警，1代表报警

fAlarmCb cbAlarmFun; //报警回调函数

void \*pUserData; //用户数据

}JXJ\_CU\_LOGON\_S;

#### 2.4.1.2 jxj\_cu\_logon 登录函数

**函数原型**：

long jxj\_cu\_logon(const char\* pIpAddr, const char\* pUsr,

const char\* pPwd, const char\* pPort,

fAlarmCb cbAlarmFun, void\* pUserData);

**功能说明**：

连接IPC、保存与IPC通信的参数信息，并创建接收发送处理线程*jxj\_cu\_thread\_fun*.

**参数说明**：

**pIpAddr: ip地址**

**pUsr : 用户名**

**pPwd : 密码**

**pPort : 端口**

**cbAlarmFun: 报警回调函数**

**pUserData: 用户参数(*存放连接通道号*)**

**返回值**：

成功返回登录句柄(JXJ\_CU\_LOGON\_S地址);

失败返回0;

**代码解析**：详见jxj\_cu\_logon函数源码与注释.

#### 2.4.1.3 jxj\_cu\_logout 注销登录函数

**函数原型**：

int jxj\_cu\_logout(long longHandle);

**功能说明**：

退出登录IPC.

**参数说明**：

**longHandle: 登录句柄**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_logout函数源码与注释.

#### 2.4.1.4 jxj\_cu\_cfg\_get 获取参数函数

**函数原型**：

int jxj\_cu\_cfg\_get(long logonHandle, JParamId type,

void \* pCfg, int cfgSize);

**功能说明**：

获取IPC参数信息(*将参数信息封装成cu协议的xml数据格式,xml数据格式定义详见”* ***透传协议.doc****”文档.后续参数模块参数获取最终调用此接口*).

**参数说明**：

**longHandle: 登录句柄**

**type: 命令码(*详见****JParamId***)**

**pCfg: 存放响应参数配置**

**cfgSize: 配置参数大小**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_cfg\_get函数源码与注释.

#### 2.4.1.4 jxj\_cu\_cfg\_set 设置参数函数

**函数原型**：

int jxj\_cu\_cfg\_set(long logonHandle, JParamId type,

void \* pCfg, int cfgSize);

**功能说明**：

设置IPC参数信息(*将参数信息封装成cu协议的xml数据格式,xml数据格式定义详见”* ***透传协议.doc****”文档.后续参数模块参数设置最终调用此接口*).

**参数说明**：

**longHandle: 登录句柄**

**type: 命令码(*详见****JParamId***)**

**pCfg: 设置参数配置**

**cfgSize: 配置参数大小**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_cfg\_set函数源码与注释.

#### 2.4.1.5 jxj\_cu\_logon\_test 登录测试函数

**函数原型**：

int jxj\_cu\_logon\_test(const char\* pDstIP, const char\* pUsr,

const char\* pPwd, const char\* pPrt);

**功能说明**：

登录测试(*将参数信息封装成cu 登录数据xml格式, 根据响应xml中的resultCode值判断登录是否成功: 0-代表成功，其它值-代表错误码. 详见”* ***透传协议.doc****”文档.*).

**参数说明**：

pDstIP**: ip地址**

**pUsr : 用户名**

**pPwd : 密码**

pPrt**: 端口**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_logon\_test函数源码与注释.

### 2.4.2参数控制

**参数控制源码位于src目录下jxj\_cu\_param.c与jxj\_cu\_net\_param.c文件.**

参数获取与参数设置最终调用**jxj\_cu\_logon.c文件中**jxj\_cu\_cfg\_get与 jxj\_cu\_cfg\_set函数.

CU各参数的结构体信息及响应命令码详见**j\_sdk.h**头文件。

#### 2.4.1.1 jxj\_cu\_param.c文件解析

该文件提供了Cu协议各个参数获取设置接口. 所有参数获取设置成功返回0;

失败返回-1.

#### 2.4.1.1 jxj\_cu\_net\_param.c文件解析

该文件提供了Cu协议各个参数与IPNC通用结构体获取设置参数转换接口.

在**jxj\_cu\_net\_param.c**文件中全局量sCmdParam数组提供所有支持IPNC参数获取设置接口.

**参数获取：**

成功返回参数大小;失败返回-1.

**参数设置：**

成功返回0;失败返回-1.

IPNC各参数的结构体信息及响应命令码详见**ipnc\_lib.h**头文件。

### 2.4.3 PTZ控制

**PTZ控制源码位于src目录下jxj\_cu\_ptz.c文件.**

#### 2.4.3.1 jxj\_cu\_ptz\_set 发送ptz控制函数

**函数原型**：

int jxj\_cu\_ptz\_set(long logonHandle, JPTZControl \*cfg);

**功能说明**：

发送PTZ控制, 最终调用jxj\_cu\_cfg\_set函数.

**参数说明**：

**longHandle: 登录句柄**

**cfg: 命令码(***JPTZControl结构体数据***)**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_ptz\_set函数源码与注释.

#### 2.4.3.2 jxj\_cu\_client\_crtl\_ptz 控制ptz函数

**函数原型**：

int jxj\_cu\_client\_crtl\_ptz(long logonHandle,

JPTZAction action,

uint8\_t u8Speed);

**功能说明**：

PTZ操作控制, 控制码详见JPTZAction, 最终调用jxj\_cu\_ptz\_set函数.

**参数说明**：

**longHandle: 登录句柄**

**action: 控制命令码(*详见JPTZAction*)**

**u8Speed: 速度**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_client\_crtl\_ptz函数源码与注释.

#### 2.4.3.3 jxj\_cu\_client\_ ptz\_cruise 自动巡航设置函数

**函数原型**：

int jxj\_cu\_client\_ptz\_cruise(long logonHandle,

uint8\_t u8CruisePresetNo,

uint8\_t u8CruisePresetEnable,

uint8\_t u8CruisePresetSec,

uint8\_t u8CruisePresetSpeed);

**功能说明**：

自动巡航设置, 最终调用jxj\_cu\_ptz\_set函数.

**参数说明**：

**longHandle: 登录句柄**

**u8CruisePresetNo: 预置号**

**u8CruisePresetEnable: 使能与否**

**u8CruisePresetSec: 保留时间**

**u8CruisePresetSpeed: 速度**

**返回值**：

成功返回0;

失败返回-1;

**代码解析**：详见jxj\_cu\_client\_ ptz\_cruise函数源码与注释.