信令通道协议

# 1：头文件

XDefineSignalingChannelStruct.h

Json 使用的jsoncpp

a\_signaling\_channel\_json\_reader\_builder.h

# 2：tcp传输的数据头定义

enum signaling\_channel\_cmd\_def

{

SINGNALING\_CHANNEL\_CMD\_REDIRECTION\_DEVICE = 0,

SINGNALING\_CHANNEL\_CMD\_REDIRECTION\_USER,

SINGNALING\_CHANNEL\_CMD\_REDIRECTION\_CLASSIC\_GET\_DEVICES,

SINGNALING\_CHANNEL\_CMD\_LOGIN = 20, //登陆

SINGNALING\_CHANNEL\_CMD\_LOGIN\_NEED\_SESSION, //key s -> c

SINGNALING\_CHANNEL\_CMD\_HEARTBEAT, //心跳

SINGNALING\_CHANNEL\_CMD\_LOGOUT, //注销

//

SINGNALING\_CHANNEL\_CMD\_IOT\_DEVICE, //IOT的添加修改删除

SINGNALING\_CHANNEL\_CMD\_IOT\_DEVICE\_SENSORS, //IOT的传感器

SINGNALING\_CHANNEL\_CMD\_TRANS, //透明传输

SINGNALING\_CHANNEL\_CMD\_GPS,

SINGNALING\_CHANNEL\_CMD\_ALARM,

};

enum signaling\_channel\_cmd\_result

{

SINGNALING\_CHANNEL\_RESULT\_OK = 0,

SINGNALING\_CHANNEL\_RESULT\_ERROR

};

struct signaling\_channel\_head\_t

{

uint32\_t cmd; //signaling\_channel\_cmd\_def

uint32\_t result; //signaling\_channel\_cmd\_result

uint32\_t session; //

uint32\_t datalen; //数据长度

};

# 3：json定义

“header”节点：必须，必须的关键字：command，result，deviceuuid，msguuid

"command" : 命令：已定义的命令有"getdevices"，"adddevice"，"modifydevice"，"deletedevice"，"ptz"，"setparam"，"getparams"，"getsensors"，

"result" : 操作结果

"deviceuuid" : 设备id

"msguuid" : 消息ID

"info"节点，可选，"info"内的项目可选

例如：

{

"header" :

{

"command" : "cmd",

"result" : 0,

"deviceuuid" : "WWDSDJ9982HH88",

"msguuid" : "777777"

},

"info": {

"pwd": "222",

"user": "111",

}

}

# 4：信令组成

## 1.1 Tcp链路使用 数据头+json数据的方式

signaling\_channel\_head\_t + json\_data

## 1.2 Websocket使用json数据的方式

服务器会校验

# 5 ：TCP设备与服务器交互

交互时序图

请求session

登陆

心跳

# 6：TCP客户端与服务器交互

交互时序图

请求session

登陆

心跳

# 7：WEBSOCKET客户端与服务器交互

经典服务器：

请注意：请人工确保该账号下的设备都在同一个服务器上，否则请不要提供该服务给用户。

物联网服务器

交互时序图

登陆

心跳