**Hdu模块走读文档**

|  |  |
| --- | --- |
| **作者** | **吴智华** |
| 安全 | MM |
| 版本 | V1.0 |

目录

[1. 部署图 3](#_Toc385435945)

[2. 组件图 3](#_Toc385435946)

[3. 模块分解 4](#_Toc385435947)

[4. 接口定义 4](#_Toc385435948)

[4.1. 和HDU外设交互消息 4](#_Toc385435949)

[4.2. 和应用服务器交互消息 5](#_Toc385435950)

[4.3. 和HduClient交互消息 6](#_Toc385435951)

[5. 业务流程 7](#_Toc385435952)

[5.1. 初始化 7](#_Toc385435953)

[5.2. 接受Hdu连接 8](#_Toc385435954)

[5.2.1. 外设掉线 9](#_Toc385435955)

[5.3. 接受应用服务器连接 10](#_Toc385435956)

[5.3.1. 应用服务器掉线 11](#_Toc385435957)

[5.3.2. 应用服务器请求状态 12](#_Toc385435958)

[5.3.3. 应用服务器请求调整音量 13](#_Toc385435959)

[5.3.4. 应用服务器重启外设 13](#_Toc385435960)

[5.4. 接受HduClient连接 14](#_Toc385435961)

[5.4.1. HduClient掉线 15](#_Toc385435962)

[5.4.2. 开启等业务通用操作流程 16](#_Toc385435963)

[6. 类图 17](#_Toc385435964)

[7. 遗留问题 17](#_Toc385435965)

# 部署图



HduPool主要和图中三个模块通过OSP进行消息交互。

# 组件图



HduPool包含了三个模块：

1. kdvsys.lib：读取配置
2. kdvlog.lib：日志处理

3.　Osp.lib：　消息通信

# 模块分解



HduPool包含了以下几个模块：

1. CBrdSsn：负责Hdu单板接入,将单板信息报给HduPool主控
2. CEqpSsn：负责Hdu业务通信(不负责断链检测)
3. CAcsSsn：负责接受应用服务器的接入，将业务操作的消息转给HduPool主控，并将主控的操作结果或状态通告反馈给应用服务器
4. CClientSsn：负责Hdu媒体客户端的接入，将业务操作的消息转给HduPool主控，并将主控的操作结果或状态通告反馈给Hdu客户端
5. CHduPool：负责消息转发,负责所有外设数据[单板配置/通道使用状态]的保存.负责单板EqpId的分配

# 接口定义

说明:以下消息名称在HDU内部已经重新定义,但是消息ID还是和原来一样的.

## HDUPOOL和HDU外设交互消息

|  |  |  |
| --- | --- | --- |
| **消息号** | **功能** | **消息结构** |
| BOARD\_MPC\_REG | HDU外设向资源池上报单板配置 | TBrdPos + dwEqpIp + byEthChoice + byOsType+byGUidLen + abyGUid |
| MPC\_BOARD\_REG\_ACK | HDU单板注册成功 | byMpcActive + TBrdRegAck |
| MPC\_BOARD\_REG\_NACK | 单板注册失败 | NULL |

消息里有用的内容是TBrdPos和dwEqpIp，将会保留到本地(HduPoolApp)，其他的内容不保存.byEqpId已经由BrdMgr分好,带在消息的头部.

应答内容里只有TBrdRegAck有用，这是告诉对端另一块HduPool的IP/Port,通知其建双链.

|  |  |  |
| --- | --- | --- |
| BOARD\_MPC\_GET\_CONFIG | HDU外设向资源池请求单板配置 | TBrdPos |
| MPC\_BOARD\_GET\_CONFIG\_ACK | 返回单板配置请求 | TEqpHduEntry |
| MPC\_BOARD\_GET\_CONFIG\_ACK | 获取配置失败 | NULL |

给单板回的消息里只有EqpId是有用的(填成INVALID\_ID),其他都是使用的默认构造.

|  |  |  |
| --- | --- | --- |
| HDUPOOL\_HDUDISCONNECTED\_NOTIF | HDU外设掉线通知,由CBrdMgrApp通知HduPool主控 | dwNodeId |

|  |  |  |
| --- | --- | --- |
| HDU\_MCU\_REG\_REQ | HDU外设向资源池上报业务配置 | TPeriEqpRegReq |
| MCU\_HDU\_REG\_ACK | 业务注册成功 | TPeriEqpRegAck + TPrsTimeSpan + TEqpHduCfgInfo |
| MCU\_HDU\_REG\_NACK | 业务注册失败 | TPeriEqpRegReq |

资源池收到注册请求时,将TPeriEqpRegReq里的Port/通道数/外设别名,保存下来(保存到CHduPoolApp).目的是当HduClient和Acs来连接时,给它们返回这些信息.

返回的TPeriEqpRegAck里只有m\_dwMsSSrc是需要设置的(此值由main初始化时设进内存),其他的参数填0;TPrsTimeSpan外部不感知,使用默认构造的值; TEqpHduCfgInfo不需要,全填0

|  |  |  |
| --- | --- | --- |
| HDU\_MCU\_CHNNLSTATUS\_NOTIF | Hdu通知资源池通道状态属性 | THduChnStatus+ byFirstNotifTmp + byHduSubChnId |

THduVolumeInfo里m\_byEqpId/m\_byChnlIdx/m\_byVolume/m\_byIsMute是需要正确填写,其他的都填0即可.

THduChnStatus中只需要填写m\_byChnIdx, m\_byVolume, m\_byMute即可,其他都填0

## 和应用服务器交互消息

|  |  |  |
| --- | --- | --- |
| **消息号** | **消息功能** | **消息结构** |
| ACS\_HDUPOOL\_ CONN\_REQ | 应用服务器来连接 | NULL |
| HDUPOOL\_ACS\_ CONN\_ACK | 响应连接请求，并上报应用服务器外设配置 | NULL |
| HDUPOOL\_ACS\_ CONN\_NACK | 连接失败 | dwErrCode |
| ACS\_HDUPOOL\_GETSTATUS\_REQ | 应用服务器请求所有外设状态 | NULL |
| HDUPOOL\_ACS\_GETSTATUS\_ACK | 状态请求成功 | NULL |
| HDUPOOL\_ACS\_GETSTATUS\_NACK | 状态请求失败 | dwErrCode |
| HDUPOOL\_ACS\_GETSTATUS\_NOTIF | 外设状态通知  场景:   1. ACS来请求,只回给对应ACS 2. 外设配置全部上报,此时广播给所有ACS | byNum + N\*THduInfo |

|  |  |  |
| --- | --- | --- |
| ACS\_HDUPOOL\_REBOOT\_CMD | 应用服务器请求某单板重启 | NULL |

|  |  |  |
| --- | --- | --- |
| ACS\_HDUPOOL\_CHANGEHDUVOLUME\_CMD | 应用服务器前来修改通道音量 | THduVolumeInfo |
| HDUPOOL\_ACS\_CHANGEHDUVOLUME\_NOTIF | 修改成功(ACS需要检验消息长度) | THduVolumeInfo |

|  |  |  |
| --- | --- | --- |
| HDUPOOL\_ACSDISCONNECTED\_NOTIF | 应用服务器掉线 | dwNodeId |
| HDUPOOL\_CLIENTDISCONNECTED\_NOTIF | Hdu媒体客户端掉线 | dwNodeId |

# 业务流程

## Mcu(hduclientlib)与hdupool的建链



## Mcu(hduclientlib)与hdupool的断链



## Css与hdupool的建链



### Css与hdupool的断链



Hdupool只清除acsId标志位，及ospNode缓存

### Hdu与hdupool 的建链



### Hdupool与hdu的断链



### Hdu简介

公司现用的hdu只有hdu2，一块hdu2板卡含有一个com口负责配置hdu；一个网口负责接收网络数据包；两个音频输出和两个视频输出口，能够同时输出两路音视频码流。所有的hdu2板卡插在一块集成板卡上（其实就是一个交换机）。hdu2上电启动就运行hdu2程序。由hdu2程序控制该hdu的全部工作。

一个hdu两个通道，每个通道外接一个电视。将电视组合起来就是一个电视墙。通过会馆的界面可以对hdu进行操作。最普通的就是将一个终端拖入hdu通道中，则该hdu的该通道将会播放终端发送的码流。另外hdu还有四个子通道。每个子通道也能播放一路视频码流。就是所谓的四分屏。未开启四分屏时，使用的都是该hdu的0子通道。除此之外会馆的主席跟随，发言会场跟随，内容共享跟随都是先预占一个hdu通道，然后按照当前跟随的是哪个终端就将哪个终端的码流打向hdu。所有会馆的这些操作基本上都是会馆通过MCU发出命令，设置声音是会馆通过CSS发出命令。

## Hdu通道的正常播放流程

