复用模块说明书

版本	日期	作者	说明
1.0	2016-7-11		1、multiplex

1. 主要功能

模块主要功能：

1、 SPTS 模式 Ts 流复用成 MPTS 模式的 Ts 流输出（暂时未支持）

2、 SPTS 模式 Ts 流直通输出

2. 限制条件

1、 暂时只支持 Ts Over Udp 输入。

2、 暂时只支持 SPTS 模式输出

3、 此模块暂时未支持

3. 输入

1、 协议支持： Ts Over Udp

2、 IP 输入模式：SPTS

3、 输入通道：暂时支持 4 路 SPTS 输入

4. 输出

1、 协议支持：Ts Over Udp

2、 IP 输出模式：SPTS

3、 输出通道：暂时支持 4 路 SPTS 输出

5. 目录结构

Objs 目录：存放编译时产生的.o 等文件，是一个临时文件

.c/.h 文件：模块的源文件

Makefile 文件：编译所需文件

Build.sh 文件：一键编译文件，产生的可执行文件 multiplex_x64 将被放到上级目录的 build 文件中

依赖库文件： libmxml_x64.a（XML 库文件）、libmsg_x64.a（消息机制库文件）

6. 流程

一、代码流程图

```
pChannel = &pHandle->m_channel;
//解释参数
Mp_Core_Parse_Argv(pHandle,argc,argv);

//加载默认参数信息
Mp_Core_Def_Paramter_Load(pHandle);

//加载配置参数信息
Mp_Xml_Paramter_Load(pHandle,MP_XML_DIR,MP_CHANNEL_XML);

//参数合法性进行检测
Mp_Core_Check_Paramter(pHandle);

//参数回写输出
Mp_Xml_Paramter_Save(pHandle,MP_XML_DIR,MP_CHANNEL_XML);
Mp_Xml_Item_Save(pHandle,MP_XML_DIR);

#ifdef TS_LIB
if(0 == TS_Init())    //初始化模块
{
    TS_SearchCallBack_Register(Mp_Core_SearchFinish);      //搜索结束回调
    TS_TimerOutCallBack_Register(3000,Mp_Core_TsTimerOut);//节目超时回调
}
#endif

//Mt_Init(Mp_Core_TsInfo_Handler);

//通道线程
for(idx = 0;idx < pChannel->m_num; idx++){
    #ifdef TS_LIB
    TS_Channel_Register(pChannel->m_inchn[idx].m_chno,1); //注册节目搜索
    #endif
    Mp_Pthread(&pChannel->m_inchn[idx].m_pid,Mp_Core_InChn_Main,&pChannel->m_inchn[idx]);
}
Mp_Pthread(&pChannel->m_outchn.m_pid,Mp_Core_OutChn_Main,pHandle);

//启用通信
pHandle->mq_recv = Msg_Recv_Create(MQ_MP_RECV);
if(pHandle->mq_recv)
{
    pHandle->mq_send = Msg_Send_Create(MQ_MP_SEND);
    if(pHandle->mq_send)
    {
        pHandle->m_ready = TRUE;
```

图 1

二、代码详解（按执行顺序）

1、 首先为进程的全局 handle 分配空间

pHandle = (Mp_Handle *)calloc(1,sizeof(Mp_Handle));

2、 解释参数

Mp_Core_Parse_Argv（）;

3、 加载默认参数信息

Mp_Xml_Paramter_Load（）;

4、 参数合法性进行检测

Mp_Core_Check_Paramter（）；

5、 参数回写输出

Mp_Xml_Paramter_Save（）；

Mp_Xml_Item_Save（）；

6、 通道线程

//注册节目搜索

TS_Channel_Register（）；

//输入处理线程

Mp_Core_InChn_Main（）；

具体处理流程：

1、 从 IP 端口中接收 UDP 包数据。

2、 输入通道数据处理程序，调用：

Mw_Core_InChnRecv_Handler（）。

//按 SPTS 或者 MPTS 发送音视频数据

Mp_Core_Send_Handle（）；

//输出处理线程

Mp_Core_OutChn_Main（）；

具体处理流程：

定时发送 PAT、PMT、SDT 表

SPTS：

TS_Spts_Pat_Get（）；

TS_Spts_Pmt_Get（）；

TS_Spts_Sdt_Get（）；

MPTS：

TS_Mpts_Pat_Get（）；

TS_Spts_Pmt_Get（）；

TS_Mpts_Sdt_Get（）；

7、 创建本进程消息队列

pHandle->mq_recv = Msg_Recv_Create(MQ_MP_RECV);

pHandle->mq_send = Msg_Send_Create(MQ_MP_SEND);

8、 阻塞等待接收消息，并且判断消息类型

Msg_Recv(pHandle->mq_recv,&rmsg);

主要流程如下，包括 POST 和 GET 方法：

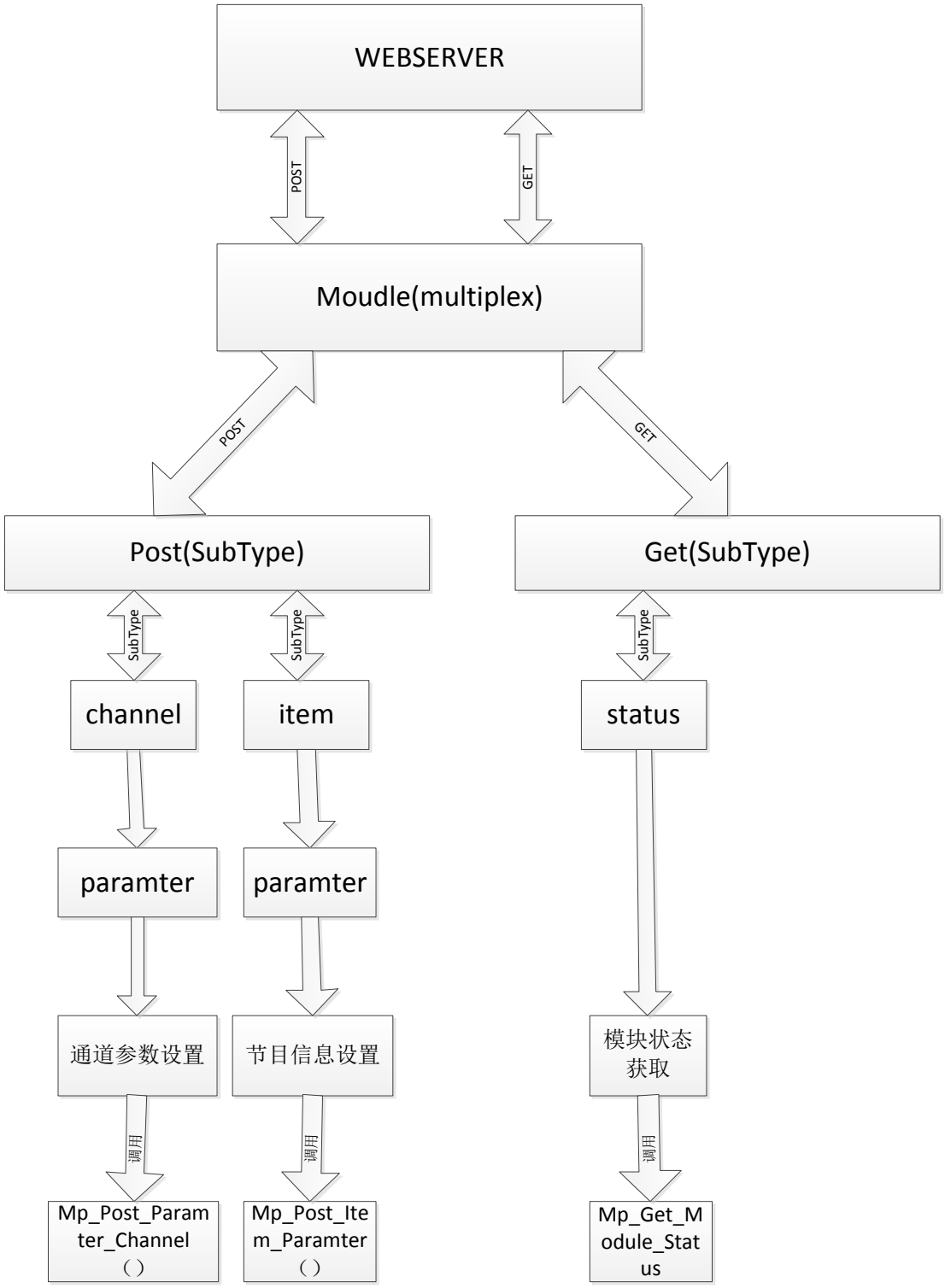

图 2　WEB 和本进程交互过程

三、关联 WEB 界面

1、SPTS 输出相关设置

图 3 SPTS 相关设置

3、 MPTS 相关设置

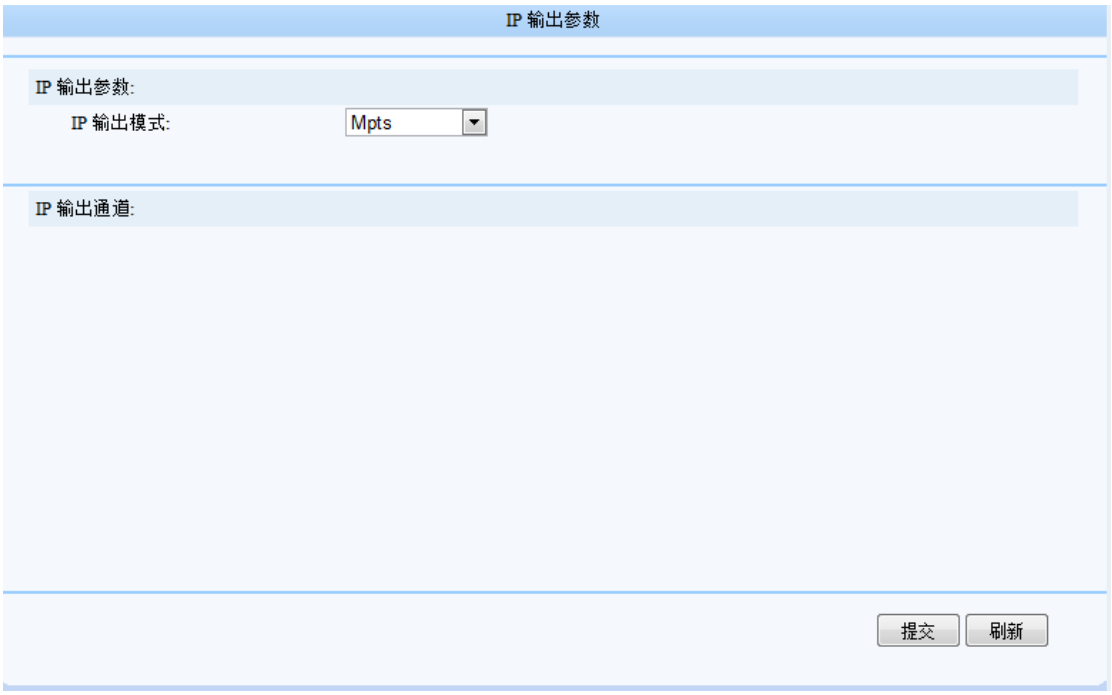

图 4 MPTS 相关设置

7. 鸣谢

感谢之前对这个模块有过贡献的同事们。