datamul 模块说明书

版本	日期	作者	说明
1.0	2016-7-13		1、data_mul

1. 主要功能

模块主要功能：

1、 传输采集模块采集的数据

2、 对数据进行复制，以期实现一对多编码

2. 限制条件

1、 暂时最大支持到 64 路音视频输入

2、 暂时最大支持 16 路音视频输出

3、 同一台服务器中 Magewell 采集卡的拨码开关设置的通道不能重复

3. 输入

1、 64 路音视频，通过内存共享

4. 输出

1、 16 路音视频，通过内存共享

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 data_mul_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）、libmem_share.a（内存共享库文件）

6. 流程

一、代码流程图

```
g_Capture_pHandle = pHandle;

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

//加载参数
Capture_Xml_ChannelParamter_Load(pHandle,CAPTURE_XML_DIR,CAPTURE_CONFIG_XML);

//保存参数
Capture_Xml_ChannelParamter_Save(pHandle,CAPTURE_XML_DIR,CAPTURE_CONFIG_XML);

mssleep(100);

shmkey_init();
shm_enc_init();

mssleep(100);

pthread_create(&pHandle->m_pid,NULL,capture_data_mul,NULL);

//启用通信
pHandle->mq_recv = Msg_Recv_Create(MQ_CAPTURE_RECV);
pHandle->mq_send = Msg_Send_Create(MQ_CAPTURE_SEND);

if(pHandle->mq_recv == NULL || pHandle->mq_send == NULL)
{
    char * mesg = strerror(24);
    printf("Mesg:%s\n",mesg);

    return 0;
}

pHandle->m_ready = 1;
pHandle->m_ready = 1;
while(pHandle->m_ready)
{
    memset(&rmsg, 0 ,sizeof(Msg_Param));

    lRet = Msg_Recv(pHandle->mq_recv,&rmsg);
```

图 1 代码流程图

二、流程框图

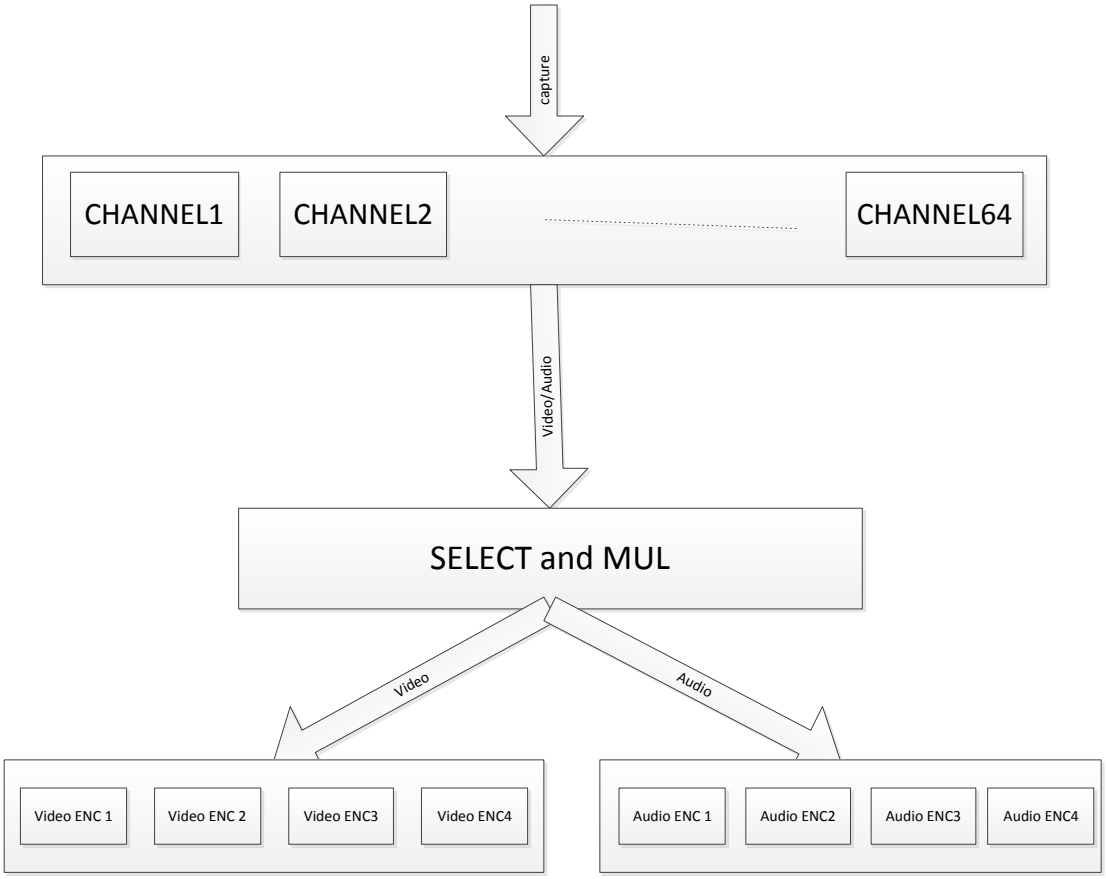

图 2 流程框图

三、代码详解（按执行顺序）

1、 首先为进程的全局 handle 分配空间

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

2、 加载默认参数信息

Capture_Core_Def_Paramter_Load(pHandle);

3、 加载 XML 参数

Capture_Xml_ChannelParamter_Load（）;

4、 参数回写输出

Capture_Xml_ChannelParamter_Save（）;

5、 初始化共享内存 ID

shmkey_init();

6、 初始编码音视频化共享内存

shm_enc_init();

7、 创建数据复制线程

pthread_create(&pHandle->m_pid,NULL,capture_data_mul,NULL);

分析 capture_data_mul：

1、共享内存设置成 PULL 模式

shm_init(SHM_PULL_MODEL);

2、判断采集通道的共享内存是否被编码进程使用，若使用返回共享内存描述符

share_fd_tmp[share_num] = shm_chn_attach(share_num);

3、PULL 数据

handle= shm_pull(share_num, r_tmp);

4、共享内存设置成 PUSH 模式

shm_init(SHM_PUSH_MODEL);

5、PUSH 数据，复制数据

shared_mem_release(r_tmp, share_num, handle_capture);

6、释放 PULL 产生的 handle

shm_release(handle);

8、 创建本进程消息队列

pHandle->mq_recv = Msg_Recv_Create(MQ_CAPTURE_RECV);

pHandle->mq_send = Msg_Send_Create(MQ_CAPTURE_SEND);

9、 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：

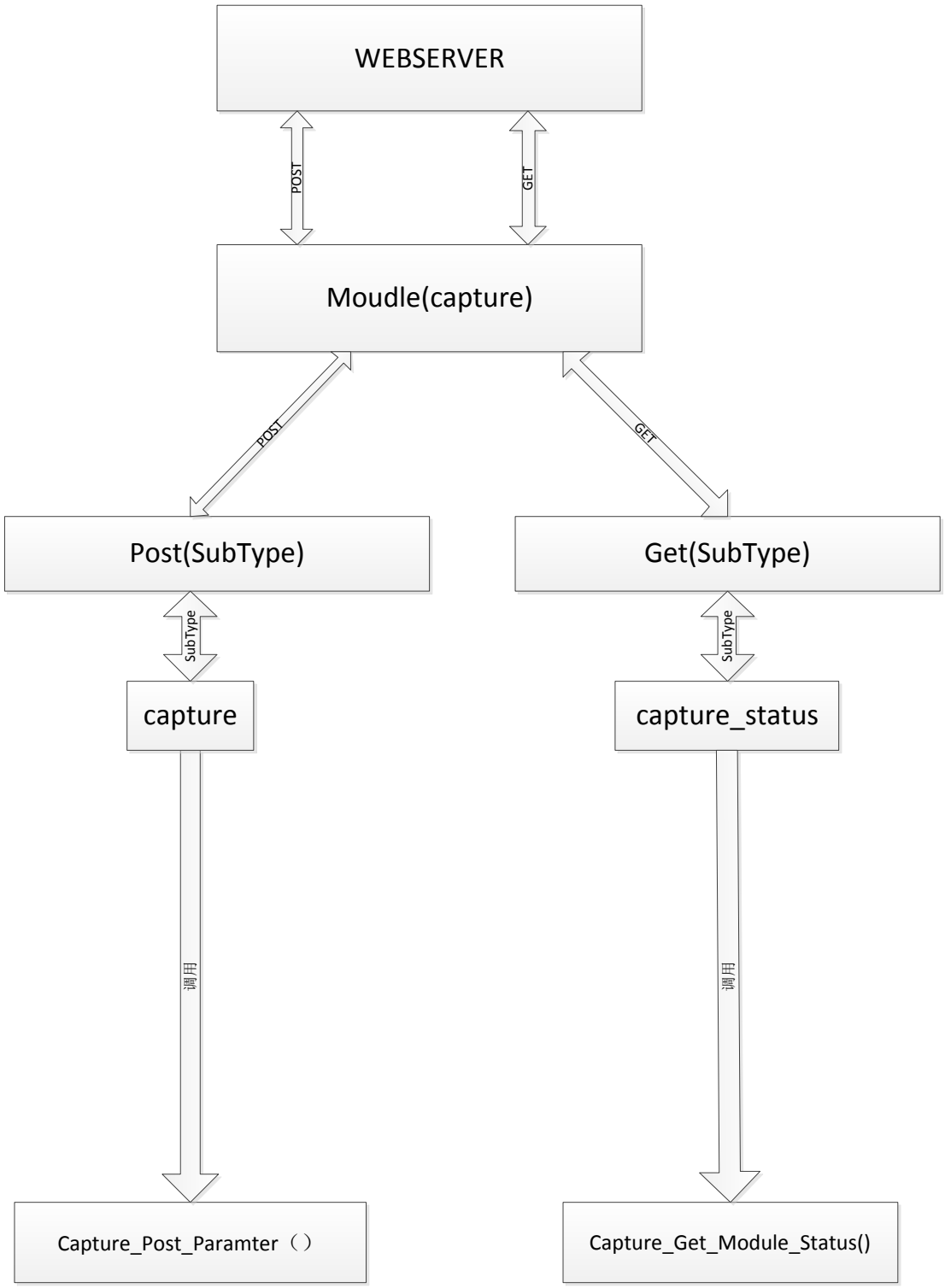

图 2　WEB 和本进程交互过程

7. 鸣谢

感谢之前对这个模块有过贡献的同事们。