

## A7 DevMng

从xml配置文件加载驱动配置  
pDeviceMng->loadCfgFile(CFGXML\_FILE\_PATH);

从ini配置文件加载消息键值配置  
pDeviceMng->loadShareParam(CFGINI\_FILE\_PATH);

检查配置文件参数是否有效  
pDeviceMng->CheckParamValidity();

根据键值ini文件中的参数初始化消息队列  
pMsgMng->Init(pDeviceMng->GetDeviceMngKey(),  
pDeviceMng->GetDeviceMngKey() + 1, pDeviceMng->  
GetDeviceMngKey() + 2, pDeviceMng->  
GetDeviceMngKey() + 3)

创建消息对象  
pRDriverMsg->create\_object()  
pRAppMsg->create\_object()

pthread\_create(&DriverMsg\_id, NULL, DriverMsg\_task, NULL); // 驱动消息接收任务87654332  
pthread\_create(&AppMsg\_id, NULL, AppMsg\_task, NULL); // 应用消息接收任务87654330

运行驱动程序  
pDeviceMng->SetupDriver(pMsgMng->pSDriverMsgKey)

pdriver = new driver(CfgList.at(i).script, cfg.id, cfg.name, msgid); // 创建带有键值的驱动程序对象

初始化驱动消息对象，创建消息标识  
pdriver->InitMsg() //87654333

初始化驱动  
pdriver->Init() 获取驱动信息，超时时间5s（该功能实际无用）  
Msg\_GetInfo()

根据指定的消息键值和驱动程序ID运行驱动程序  
ShellSetupDriver(cfg.script, cfg.name,  
cfg.id, SysMinKey + 2, SysMinKey + 3)

周期性（间隔3s）向驱动发送心跳  
pDeviceMng->SendHeartToDriver();

向驱动发送心跳  
Msg\_SendHeart()

2s后检测驱动程序的运行情况，并根据情况发送状态报告消息  
pDeviceMng->DriverHeartMng();

(pids.size() > 1) // 如果进程ID列表的大小大于1，表示进程存在多个实例  
system(cmd.toStdString().c\_str()); // 终止所有驱动程序的进程

通信异常或异常恢复时向应用广播消息  
pMsgMng->BroadcastToApp(MSG\_TYPE\_AppReportDriverComNormal, &data,  
1);

驱动异常，杀死所有该名称驱动，调用启动脚本重启  
ShellSetupDriver(item.value()->GetDriverScript(), item.value()->GetDriverName(), item.value()->GetDriverId(), SysMinKey + 2,  
SysMinKey + 3);