飞机参数服务仿真模块说明文档V2.0

2015.11.16

# 前言

依据《飞机参数服务通讯协议》与《航电接口数据通信协议》，参考现有仿真软件APSSv1.0进行修改移植，使之能运行于UTEST平台上。

本文档提到的飞机参数仿真软件，以下简称APSS。

# 总体设计

为减少软件复杂度，APSS将划分为两个子模块：对外参数服务与航电数据更新。

航电数据更新模块负责向AIDCP注册订阅周期性消息，并将原始消息解包、更新到共享数据变量中；

对外参数服务模块则根据成员系统的请求，从共享数据区读取数据，组包发送给成员系统。



# 模块说明

*全部的函数说明，最好能使用工具自动生成。*

## ApsXmlOp

负责飞机参数的初始化，从配置文件/tmp/aps\_config.xml中获取信息。

**入口函数**：

bool getDataFromXMLFile(char\* url) ;

**输出变量**：

static int paramnum;

static APS\_CONFIG \*g\_apsconfig;

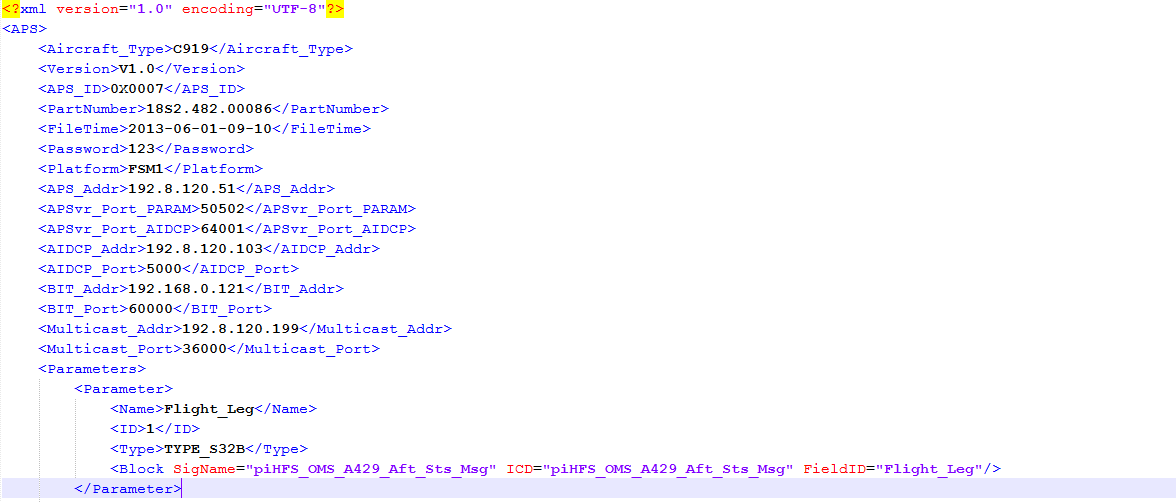
static CParamData \*g\_hparameter;

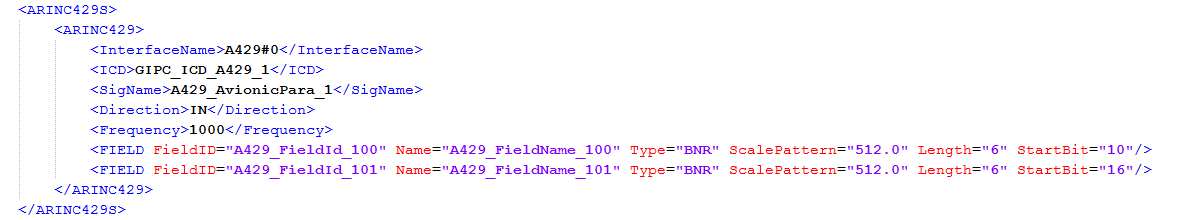
static std::list<int> flightModes;

其中CParamData参数定义比较复杂，基本上是引用原有结构定义。

一个参数包括最多2个信号，每个信号可包括多个数据域（field），但目前从使用情况上看，由于参数定义会具体到ICD信号的具体一个域，所以他们都是1对1的关系。参数解码后的值存放于CAPSParaVal。

配置文件已定义的ICD节点包括A429和AFDX。







## ApsHDSvr

航电数据更新模块负责向AIDCP注册订阅周期性消息，并将原始消息解包、更新到共享数据变量中。

**入口函数**：

bool apsHd\_start(bool isDebug);

bool apsHd\_stop();

**输出变量**：

static CParamData \*g\_hparameter;





updateParams负责更新数据，即进行解码，目前处理的数据类型包括以下11种：

TYPE\_F64B=0,

TYPE\_S64B,

TYPE\_F32B,

TYPE\_U32B,

TYPE\_S32B,

TYPE\_BOOL,

TYPE\_STRING,

TYPE\_DATE,

TYPE\_TIME,

TYPE\_OPAQUE,

TYPE\_ENUM,

特别要注意的是，A429和AFDX数据的处理方式是不一样的。

A429采用get429Value进行标准数值的获取，不处理字符串；

AFDX支持各种数据类型，特别是对2字节、4字节和8字节整数进行了网络字节序转换。

此模块可在后期进行修改，使得ICD定义的数据类型，能够映射到APSS通信协议定义的数据类型上。

## ApsParaSvr

对外参数服务模块则根据成员系统的请求，从共享数据区读取数据，组包发送给成员系统。

**入口函数**：

bool apsHd\_start(bool isDebug);

bool apsHd\_stop();

**输出变量**：

static CParamData \*g\_hparameter;



## HDMessageHandle

定义数据类型，处理通信协议xml格式的组包、解包。

这个类与AISS的相似，但有APSS自有的内容，包括函数和数据类型定义。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*for APSS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

static void GetPublishAvionicsDataPesponse(ST\_DATA\_RESPONSE\* \_data,

char\* buffer, int bufferSize);

static void GetSubscribeAvionicsDataPesponse(ST\_DATA\_RESPONSE\* \_data,

char\* buffer, int bufferSize);

static void GetUnSubscribePesponse(ST\_DATA\_RESPONSE\* \_data, char\* buffer,

int bufferSize);

static void GetHeartBeatResponse(ST\_HEARTBEAT\_RESPONSE\* \_data, char\* buffer,

int bufferSize);

static void GetPeriodicMessage(ST\_DATA\_RESPONSE\* \_data, char\* buffer,

int bufferSize);

static void GetEventMessage(ST\_DATA\_RESPONSE\* \_data, char\* buffer,

int bufferSize);

static void SetSubscribeAvionicsData(ST\_DATA\_REQUEST data, char\* buffer);

static void SetUnSubscribe(ST\_DATA\_REQUEST data, char\* buffer);

static void SetHeartBeat(ST\_HEARTBEAT\_REQUEST data, char\* buffer);

static void SetRegisterData(ST\_APP\_REGISTER\_DATA \_data, char\* buffer);

static void GetRegisterResponseData(ST\_APP\_REGISTER\_RESPONSE\_DATA\* \_data,

char\* buffer, int bufferSize);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*for APSS client \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

static APS\_MESSAGE\_ENUM GetApsMessageName(char\* buffer, int bufferSize);

static void GetApsRequestMsg(APS\_DATA\_REQUEST\* \_data, char\* buffer,

int bufferSize);

static void SetApsResponseMsg(APS\_DATA\_RESPONSE data, char\* buffer);

static void CreateGenericProperty(double num, xmlNodePtr node,

xmlChar\* propNm, char\* fmt);

static void CreateLongLongProperty(unsigned long long num, xmlNodePtr node,xmlChar\* propNm);

static void CreateFloatProperty(float num, xmlNodePtr node, xmlChar\* propNm);

## CommonUtility

定义公用函数。

只有HDMessageHandle在用了。

## APSS

本模块由UTEST的user code模板生成，主要负责调用APSS模块功能。

调用APSS时，最好能配置成周期性运行，方便接口数据的反馈。

**采集开始时候，启动APSS服务：**

eResult APSS::start() {

char filename[] = "/tmp/aps\_config.xml";

ApsXmlOp::getDataFromXMLFile(filename);

strcpy(ApsXmlOp::g\_apsconfig->APS\_Name, "APS");

ApsXmlOp::g\_apsconfig->ProcessId = getpid();

ApsHDSvr::g\_apsconfig = ApsXmlOp::g\_apsconfig;

ApsHDSvr::g\_hparameter = ApsXmlOp::g\_hparameter;

ApsHDSvr::paramnum = ApsXmlOp::paramnum;

ApsParaSvr::g\_hparameter = ApsXmlOp::g\_hparameter;

ApsParaSvr::g\_apsconfig = ApsXmlOp::g\_apsconfig;

ApsParaSvr::paramnum = ApsXmlOp::paramnum;

ApsParaSvr::flightModes = ApsXmlOp::flightModes;

ApsHDSvr::apsHd\_start(false);

ApsParaSvr::aps\_start(false);

return eResult\_OK;

}

**每个周期内，将接口数据更新：**

eResult APSS::step() {

ApsParaSvr::debug = (\_ucDef->isDebug == 1);

ApsHDSvr::debug = (\_ucDef->isDebug == 1);

\_ucDef->isHdRegist = ApsHDSvr::isRegist;

\_ucDef->isParaRun = ApsParaSvr::isRun;

return eResult\_OK;

}

**UTEST取消部署的时候调用APSS停止函数：**

eResult APSS::stop() {

ApsParaSvr::aps\_stop();

ApsHDSvr::apsHd\_stop();

return eResult\_OK;

}

# 文件清单

|  |  |  |
| --- | --- | --- |
| 编号 | 文件名称 | 说明 |
| 1 | APSS.cc | 飞参调用接口文件 |
| 2 | APSS.h |  |
| 3 | ApsHDSvr.cpp | 飞参与航电接口服务通讯模块 |
| 4 | ApsHDSvr.h |  |
| 5 | ApsParaSvr.cpp | 飞参对外提供参数服务模块 |
| 6 | ApsParaSvr.h |  |
| 7 | ApsXmlOp.cpp | 读取飞参配置参数模块 |
| 8 | ApsXmlOp.h |  |
| 9 | HDMessageHandle.cpp | 处理通信协议xml格式的组包、解包。 |
| 10 | HDMessageHandle.h |  |
| 11 | CommonUtility.cpp | 一些公用函数，主要是HDMessageHandle使用 |
| 12 | CommonUtility.h |  |
| 13 | /tmp/aps\_config.xml | 飞参服务的配置文件，固定位置/tmp/ |

# 使用说明

APSS加载到UTEST平台后，包含4个参数

IsParaRun和isHdRegist分别代表飞参服务是否已准备好对外提供参数服务、是否已经与AIDCP建立连接并通讯，当采集开始时，如果没有错误发生，其值应为1（代表成功）；

IsDebug表示是否在控制台打印调试信息，1为打印，0为不打印；

startCmd暂作保留。

调用APSS时，最好能配置成周期性运行，方便接口数据的反馈。

# 开发FRQ

UTEST运行或者停止时出现Segment fault错误，主要原因如下：

1. C/C++地址、指针分配使用错误；
2. 应该使用地址&的地方，遗漏&，但编译不报错；
3. 数组使用越界未保护，同时导致结构体内的数据被篡改，比较难发现错误，如果已超出本结构体数据区，导致异常，直接退出；
4. 已打开的socket连接未正常关闭；
5. 线程中sleeep时间过长，无法及时退出；
6. 已运行线程未正常结束；
7. 调用printf时候，注意参数的正确性及其打印格式。