

文章编号:1001-893X(2012)06-1027-04

# 适用于民机 IMA 的通用机载软件开发平台\*

郑 澜,王运盛

(中电科航空电子有限公司,成都 611731)

**摘 要:**针对目前国内民用飞机在综合模块化航电系统(IMA)集成应用技术的发展,参考国外民用飞机航电系统的 IMA 体系结构和 IMA 集成应用技术,结合分区操作系统的特点,给出了在分区操作系统下开发 BSP 和 AFDX 驱动移植的方法。利用这些方法可以构建一种满足民用飞机的通用机载软件开发平台,为民用飞机中 IMA 上的驻留功能软件开发提供基础实验环境和验证平台。

**关键词:**民用飞机;综合模块化航电系统;AFDX 驱动;分区操作系统;软件开发平台

**中图分类号:**TP399;V243 **文献标志码:**A **doi:**10.3969/j.issn.1001-893X.2012.06.040

## A General Airborne Software Development Platform for Civil Aircraft IMA

ZHENG Lan, WANG Yun-sheng

(China Electronics Technology Avionics Co., Ltd., Chengdu 611731, China)

**Abstract:** According to the development of domestic civil aircrafts in IMA(Integrated Modular Avionics) integrated application technology, with reference to foreign civil aircraft avionics system architecture and integrated application technology of IMA, combined with characteristics of partition operating system, the methods of BSP development and AFDX drive transplantation in partition operating system are given. These methods can be used to build general airborne software development platform for civil aircrafts, and to provide basis experimental environment and verification platform for IMA hosted function software development.

**Key words:** civil aircraft; IMA; AFDX driver; partition OS; software development platform

## 1 引 言

从 20 世纪 90 年代起,技术的进步促使航电系统从联合式结构转向综合模块化航电系统(Integrated Modular Avionics, IMA),以外场更换模块(Lined Replaced Module, LRM)为基本系统单元,系统实现高度的物理综合和功能综合。

目前在民用飞机,特别是大型客机 A380 和 B7E7 上都采用综合模块化航空电子体系结构,都是以 ARINC 653 为基础的操作系统,将飞机上大量的功能处理放入模块化的输入/输出和通用处理硬件资源中。这种标准化的硬件和软件结构方法能够减

少采购成本,减轻资源压力,同时也保证了系统的可靠性和软件的安全性。

但是,目前国外供应商对基础开发平台的提供较为保守,短期内难以提供相应的开发平台,这就需要我们采用通用的硬件处理资源,研究 ARINC 653 分区操作系统的特点,自主构建适合于民用飞机的 IMA 综合模块化通用机载软件开发平台,从而在实践中不断提高自身的技术水平。

## 2 基于 IMA 的通用机载软件开发平台

通用机载软件开发平台的研究结合国外民用飞机航电系统的 IMA 体系结构,采用 PowerPC 系列处

\* 收稿日期:2011-12-08;修回日期:2012-02-28

理器,突破分区操作系统 BSP 开发、AFDX 驱动移植等关键技术,提供一套适合于民用飞机的通用机载软件开发平台,用于支撑 IMA 上的驻留功能软件的开发。

2.1 IMA 简介

为了降低民用飞机航电系统的成本,满足未来民用飞机对航空电子系统高灵活性、高可靠性、高安全性以及便于升级换代等要求,当前在民用飞机航电系统中采用综合模块化的航空电子体系架构,即 IMA 系统架构。IMA 是一种在软件控制下的高度综合化的航空电子体系,其目标之一是实现模块的标准化、重复使用和可互换性,通过严格定义和控制各模块的软硬件及接口,达到用少量几种模块支持当前和未来多种航空电子系统功能的目标。

民机的机载设备软件的研发必须满足 DO - 178B 标准规定的软件开发过程,这样才能确保研发的软件在功能上正确,在安全上可信,才能达到适航的要求。在 IMA 系统中,多个应用软件运行在同一个处理器上,每个应用具有不同的 DO - 178B 安全等级要求,这对航电系统开发提出了新的要求。这种不同安全等级的混合应用需要操作系统为每个应用程序提供独立的运行环境,包括时间上及空间上的隔离。

2.2 ARINC 653 简介

为适应 IMA 的应用需求,ARINC Inc. 于 20 世纪 90 年代中期发布了 ARINC 653 规范。ARINC 653 规定每个应用具有私有的处理器时间片,且不能被其他应用中中断;每个应用具有私有的运行空间,不能被其他应用访问。因此,对于 IMA 系统,ARINC 653 规范实现了不同应用软件之间的隔离,使具有不同安全等级的 DO - 178B 认证的应用程序可以运行在同一处理器上。从另一个角度来讲,符合 ARINC 653 规范,是 IMA 驻留软件通过 DO - 178B 认证的一个步骤。

VxWorks 653 是 Wind River 公司提供的符合 ARINC 653 规范并通过 DO - 178B A 级认证的操作系统。采用 VxWorks 653,开发人员可以灵活地将以不同标准编写的现有应用程序与其他不同安全和危急程度的应用程序结合在一起。这种灵活的应用程序在使用特性上极大地缩短了开发时间,并保留了这些应用程序原有的可靠性和健壮性。

2.3 系统组成

通用机载软件开发平台组成如图 1 所示,虚线

框内的部分即为通用机载软件开发平台。

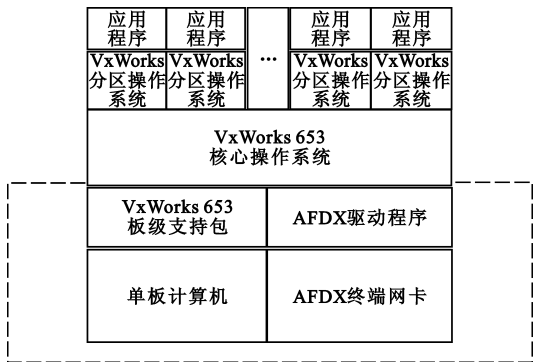


图 1 通用机载软件开发平台组成  
Fig.1 General airborne software development platform composition

为保证通用机载软件开发平台的硬件体系和架构模型的先进性和前瞻性,在平台的构建中采用通用的处理资源<sup>[1]</sup>,单板计算机采用 PowerPC 系列的 MPC 7447A 处理器作为 CPU 主板;AFDX 终端网卡采用一款高性能的 PMC 接口板卡,支持 ARINC664 规范<sup>[2]</sup>;板级支持包(BSP)和 AFDX 驱动程序支持 VxWorks 653 分区操作系统,构成了以 VxWorks 653 为主体的软件开发环境,支持时间和空间分区,为驻留的应用程序开发提供调试平台和运行环境,同时提供 AFDX 接口,应用程序可以通过 AFDX 接口与其他航电系统进行数据交互。

3 通用机载软件开发平台的集成

通用机载软件开发平台必须能够为各种应用提供稳定的运行环境,并且 IMA 中的操作系统直接关系到应用功能的开发和最终功能的实现。结合民用飞机航电系统的特点,考虑到民用飞机的安全性和机载软件适航的要求<sup>[3]</sup>,因此在通用机载软件开发平台中采用 VxWorks 653 分区操作系统,提供在分区操作系统下进行系统设计和软件开发的环境。由于在通用机载软件开发平台中采用的单板计算机和 AFDX 终端网卡只提供了 VxWorks5.5 下的 BSP 支持包和 AFDX 驱动程序,并没有提供 VxWorks 653 下的 BSP 支持包和 AFDX 驱动程序,所以需要将该单板计算机的 BSP 和 AFDX 驱动程序移植到 VxWorks 653 分区操作系统下。

3.1 VxWorks 653 BSP 开发

VxWorks 653 与 VxWorks 5.5 在驱动程序和编译环境上有很多不同,我们应充分利用 VxWorks 5.5 下

的 BSP 源代码,通过修改其中的文件结构和形式,包括修改部分源程序、添加部分文件,使其结构和形式符合 VxWorks 653 的编译环境,进一步利用编译器将修改过的 BSP 包进行编译,使编译出的结果能在单板计算机上运行。

VxWorks 653 BSP 的开发过程可分为以下 4 个阶段。

第一阶段为创建配置文件阶段。首先在 Vx-Works 653 BSP 目录下,创建 BSP 包目录,在这里可以名称为 xxx-bsp,并将原 VxWorks 5.5 下的 BSP 的内容拷贝至该目录下,同时修改 makefile 文件,创建配置文件 00region.sdf,00bootkernel.ddf,00kernel.ddf,00certkernel.ddf,00bsp.cdf,config.cdf,xxx\_bsp.xml,xxx\_bsp\_default.xml,并根据实际的资源定义每个文件的内容。

第二阶段是修改代码,并编译出 bootrom。这个阶段的目标是编译出 bootrom 文件,并使 bootrom 可以驱动串口。这个阶段过程比较繁琐,因为环境的变化,需要修改的文件可能会很多。实现的方法是用 VxWorks653 的编译命令来编译 BSP 包,具体的命令为 make RELEASE = prj\_bootApp,编译环境会在 \target \ proj 目录下创建 bootrom 工程,并编译工程。编译时,会出现错误提示,依照错误提示逐条修改。修改代码时,可只保留核心驱动和串口驱动程序,对于其他的驱动程序可暂不移植,以后再做处理,这样可简化移植过程。

第三阶段是修改代码,并编译 CoreOS。这个阶段的目标是编译出 CoreOS 文件,具体的命令为 make release,此阶段需将网络相关程序、PCI 总线驱动程序移植完成。

第四阶段为测试阶段。这个阶段的目标是编译完整的 VxWorks 653 工程,并进行网络、PCI 接口等的测试工作。

### 3.2 AFDX 驱动移植

根据 ARINC 653 规范<sup>[4]</sup>,当分区与外界其他模块进行通信时,应该通过虚端口(Pseudo-Ports)进行。

在 CoreOS 中的最底层,通过与 AFDX 硬件相关的驱动程序完成 AFDX 数据的收发控制;然后在上层实现适配于 Pseudo-Driver Layer 的 AFDX Pseudo-Driver,该驱动将完成虚端口上消息接收和发送的操作与 AFDX 数据收发的关联控制。

图 2 是适用于 ARINC 653 操作系统的 AFDX 驱

动程序运行模型。

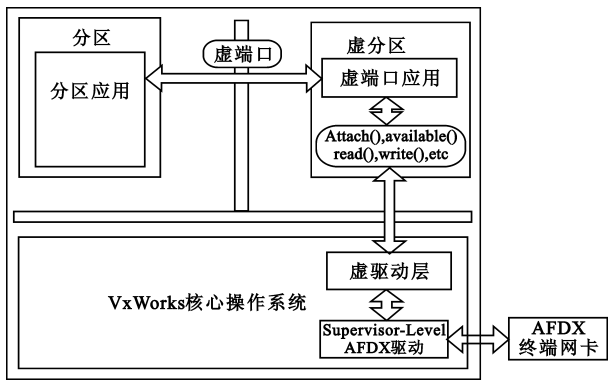


图 2 AFDX 驱动程序运行模型  
Fig.2 AFDX driver running model

位于不同模块上的两个分区之间进行通信的过程如图 3 所示。

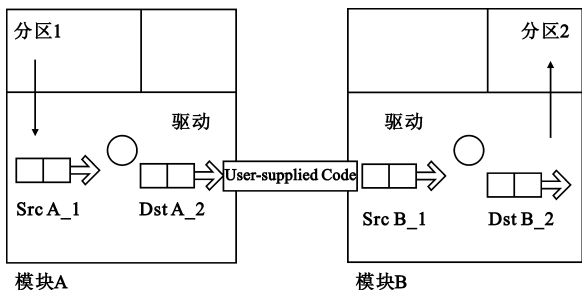


图 3 不同模块上的两个分区之间的通信  
Fig.3 Different modules on the communication between two partitions

结合图 3,解释分区 1 上的消息如何通过 AFDX 驱动程序发送至分区 2:

- (1)分区 1 上的应用通过 SEND\_QUEUE\_MESSAGE 服务,将消息发送至 Src A\_1 端口;
- (2)AFDX Pseudo-Driver 观察 Dst A\_2 端口是否可用;
- (3)如果 Dst A\_2 端口不可用,等到调度至该分区,再通过端口可用查询接口,检查该端口是否可用;如果该端口可用,则激活端口发送任务,通过调用与底层硬件相关的数据发送接口将数据发送出去;
- (4)系统在调度至分区 2 时将通过端口查询接口检查该 Src B\_1 端口是否有数据;
- (5)当有数据时,Dst B\_2 端口变为可用状态;
- (6)当分区 2 上的应用通过 RECEIVE\_QUEUE\_MESSAGE 服务接收数据时,系统将通过调用与底层硬件相关的数据接收接口获取数据,并将数据通过 Dst B\_2 端口交给应用。

在整个过程中,AFDX 终端网卡驱动对于 APEX 分区应用是不可见的。APEX 分区应用程序只需要调用标准的 ARINC 653 APEX 接口进行数据包的收发,实际过程完全由虚分区和 CoreOS 的驱动完成。

VxWorks 653 系统下 AFDX 移植过程至少需要完成如下工作:

(1)在相应的单板计算机的 BSP 包中需要实现 VxWorks 653 操作系统对 AFDX 网卡初始化的代码;

(2)在移植 AFDX 终端网卡驱动到 VxWorks 653 操作系统中时,需要考虑如何将 AFDX 终端网卡的 XML 格式配置表和 APEX 分区应用中创建的端口进行关联,以及如何加载此配置表到 VxWorks 653 操作系统中;

(3)CoreOS 层的驱动移植实现,需要充分考虑 VxWorks 653 系统结构,将 AFDX 终端网卡驱动设计融合到 CoreOS,需要实现 CoreOS 运行的数据收发任务以及和虚分区通信的任务;

(4)虚分区通信过程的实现,需要充分考虑虚分区与 APEX 分区应用如何进行通信,同时还需要考虑与 CoreOS 如何进行数据交互;

(5)APEX 应用分区例程如何完成 AFDX 数据收发,实现 APEX 分区调用通用数据通信接口收发 AFDX 数据帧。

## 4 结束语

综合模块化航电系统架构正在取代传统的联合式航电系统架构而成为今后民用飞机的主要架构方

式。采用本文给出的分区操作系统 BSP 开发和 AFDX 驱动移植的方法,能够构建一种适合于民机 IMA 的通用机载软件开发平台,从而支持 IMA 上的功能软件的开发,有助于验证民用飞机的 IMA 体系架构。在实验过程中积累的经验将为民用飞机航电系统 IMA 项目实施的规划、执行、合作提供重要参考,为后续工作的开展积累经验,从而为 IMA 的国产化奠定基础。

## 参考文献:

- [1] TSO - C153, Integrated Modular Avionics Hardware Elements[S].
- [2] ARINC 664p7, Avionics Full Duplex Switched Ethernet (AFDX) Network[S].
- [3] DO - 178B, Software Considerations in Airborne System and equipment Certification[S].
- [4] ARINC 653, Avionics Application Software Standard Interface[S].

## 作者简介:

郑 澜(1976—),男,四川成都人,2007 年获硕士学位,现为工程师,主要从事航空电子技术、机载软件的设计与研究;

ZHENG Lan was born in Chengdu, Sichuan Province, in 1976. He received the M. S. degree in 2007. He is now an engineer. His research concerns avionics technology, design and research of airborne software.

Email: xhl\_zl@163.com

王运盛(1979—),男,甘肃人,硕士,工程师,主要研究方向为综合模块化航空电子及 SCA 通信体系结构。

WANG Yun - sheng was born in Gansu Province, in 1979. He is now an engineer with the M. S. degree. His research interests include integrated modular avionics and SCA communication architecture.