

一种飞艇半物理仿真系统 ARINC429 驱动的设计与实现

李冠林¹, 乔晋龙^{1,2}

(1. 中国特种飞行器研究所, 湖北 荆门 448035; 2. 高速水动力航空科技重点实验室, 湖北 荆门 448035)

摘要: 为了满足飞艇实际飞行过程对通信可靠性和实时性的需求, 半物理仿真系统的飞艇模型计算机采用以 PC 为平台的 Xpc 嵌入式实时系统, 飞控计算机采用以 PowerPC 为平台的 VxWork 嵌入式实时系统, 通过 ARINC429 协议完成数据通讯; 硬件上, PowerPC 通过桥接芯片访问并控制 PCI 设备; 软件上, 按照 Xpc 系统和 vxWork 系统 PCI 设备驱动程序的编写流程, 分析了 xPC 内存驱动框架和 vxBus 驱动框架关键技术点, 设计了飞艇模型计算机 Xpc 和飞控计算机 Vxwork 的 ARINC429 通讯卡驱动程序; 通过实际的应用, 证实所开发的 ARINC429 驱动程序满足飞艇半物理仿真平台数据通讯的可靠性和实时性。

关键词: 半物理仿真系统; 嵌入式实时系统; ARINC429 协议; 驱动程序

Design and Implementation of a ARINC429 Driver for Semi— physical Simulation System

Li Guanlin¹, Qiao Jinlong^{1,2}

(1. China Special Vehicle Research Institute, Jingmen 448035, China;

2. Key Aviation Scientific and Technological Laboratory of High—Speed Hydrodynamic, Jingmen 448035, China)

Abstract: In order to meet the demand of communication reliability and real—time for airship in actual flight process, the airship model computer of semi—physical simulation system uses an embedded real—time system Xpc which uses PC as a platform, flight control computer uses an embedded real—time system VxWork which uses PowerPC as a platform, through the ARINC429 protocol for data communications. In hardware, PowerPC accesses and controls PCI devices through the bridge chip; in software, in accordance with written procedures of PCI device drivers in Xpc system and vxWork system, it analyzed the key technical points of the xPC memory drive frame and the vxBus drive frame, designed the communication card driver ARINC429 of airship computer model Xpc and flight control computer Vxwork. Through practical application, it confirmed the development of the ARINC429 driver meet the reliability and real—time of data communications for airship semi—physical simulation platform.

Keywords: semi—physical simulation system; embedded real—time system; ARINC429 protocol; drivers

0 引言

飞艇的半物理仿真实验是检验飞艇飞控性能的有效途径。飞艇半物理仿真系统中不同的操作系统环境, 同一硬件设备原理和流程虽然一致, 但具体驱动方法通常因操作系统的不同而有所差异^[1]。因此, 为了保证飞艇半物理仿真系统正常的 ARINC429 通讯, 开发出满足 Xpc 系统和 vxWork 系统的 ARINC429 通讯卡驱动程序是十分必要的。

1 系统概述

飞艇半物理仿真主要由飞艇仿真平台和飞控计算机组成, 硬件组成如图 1 所示。

飞艇仿真平台建立在 xPC 系统环境下, 利用飞艇模型模拟真实飞艇的飞行, 飞控计算机建立在 xWork 环境下, 负责飞艇的飞行控制和导航, 飞控计算机和飞艇仿真平台通过 ARINC429 总线完成二者之间的通讯。EFIS 负责显示飞艇飞行过程中的各种信息参数。

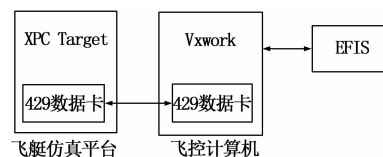


图 1 飞艇仿真平台硬件组成

2 ARINC429

ARINC429 总线是广泛应用在航空电子系统中的一种通讯总线。它采用差分的工作方式, 其结构简单、可靠性高和抗干扰性强, 被广泛应用于飞机航电设备数据传输和控制命令传输。ARINC429 传输字的格式如图 2 所示。

位号	32	31-30	29-11	10-9	8-1
功能	Odd	SSM	Data	SDI	Label号

图 2 ARINC429 传输字的格式

ARINC429 传输字格式主要由奇偶校验位 (Odd)、符号/状态位 (SSM)、数据区域 (Data)、源/目标标识 (SDI) 和标号位 (Label) 组成。在通讯过程中, 发送顺序依次为标号位、源/目标标识、数据区域、符号/状态位和奇偶校验位, 其中标

收稿日期: 2015-09-15; 修回日期: 2015-11-04。

作者简介: 李冠林 (1987—), 男, 山西朔州人, 硕士研究生, 工程师, 主要从事嵌入式方向的研究。

号位发送顺序为由高到低 (8—1), 而传输字的发送顺序都是由低到高 (9—32)。当接收方接收到 429 传输字时, 需要将 Label 位的进行逆序处理。

3 关键技术

3.1 飞艇仿真系统部分

xPC 是基于 RTW (Real-Time Workshop) 的实时仿真和开发平台, 其采用“双机”模式实现产品原型开发和系统的测试。宿主机运行 simulink 完成仿真系统模型开发, 通过网线将模型加载到目标机上; 目标机启动时加载实时内核, 该实时内核保证仿真系统高效的运行^[2]。

xPC 目标机支持 ISA 总线和 PCI 总线的 I/O 设备, 飞艇仿真模块采用 PCI 总线的 ARINC429 通讯卡。xPC 驱动程序文件即为由 C 语言编写函数组成的 S-functions 文件, 主要由初始化模块、执行模块和结束模块组成, 如图 3 所示。

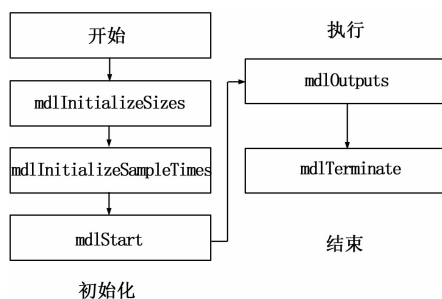


图 3 xPC 目标驱动程序组成

初始化模块包括由 mdlInitializeSizes 函数, mdlInitializeSampleTimes 函数和 mdlStart 函数组成, mdlInitializeSizes 函数和 mdlInitializeSampleTimes 函数主要负责输入输出参数初始化、状态量初始化和系统采样时间的初始化, mdlStart 函数负责 I/O 设备初始化; 执行部分根据仿真模型, 完成相应的任务, 执行部分功能集中在 mdlOutputs 函数, 该函数在负责 PCI 设备的读写操作; 结束模块功能函数为 mdlTerminate, 在运行结束后, 负责释放相关资源, 使 I/O 设备安全退出。

xPC Target 启动时, PCI 总线驱动被加载, BIOS 扫描挂在 PCI 总线上的 I/O 设备, 并为 PCI 设备分配合理的 I/O 空间地址、物理地址和中断号等资源, 无需用户在驱动程序中为 PCI 设备申请相关的资源, 驱动程序通过厂商标识 (Vendor_ID) 和设备标识 (Device_ID) 来获取 PCI 设备的配置空间信息, 代码如下:

```

r132eGetPCIInfo (Vendor_ID, Device_ID, &pciinfo);
pciinfo 是一个结构体变量, 结构如下:
{
    unsigned long BaseAddress[6];
    unsigned short AddressSpaceIndicator[6];
    ...
    unsigned short InterruptLine;
}

```

BIOS 为每个 PCI 设备提供了 6 个基址, 其中有部分基址是无效的, 通过 r132eShowPCIInfo 函数确定有效基址。PCI 设备映射分为 I/O 映射和内存映射, 驱动程序可以通过 pciinfo 中的 AddressSpaceIndicator 属性值来确定 PCI 映射类型。若 PCI 设备为内存映射, AddressSpaceIndicator 属性值 0; 若 PCI

设备为 I/O 映射 AddressSpaceIndicator 属性值 1。ARINC 429 通讯卡为内存映射, pciinfo 结构中的基址 BaseAddress 必需通过函数 r132eGetDevicePtr 转化成虚拟地址, 然后用 C 语言的指针访问相应的虚拟地址, 代码如下:

```

bar2Base = (char *) r132eGetDevicePtr ((char *) (pciinfo.
BaseAddress[2]),
4096, RT_PG_USERREADWRITE);
* (bar2Base) = 0x12;

```

编译器往往会优化内存, 将内存变量缓存到寄存器。当驱动程序进行写操作, volatile 关键词提醒编译器它后面所定义的变量随时都有可能改变, 因此编译后的程序每次需要存储或读取这个变量的时候, 都会直接从变量地址中读取数据。如果没有 volatile 关键字, 则编译器可能优化读取和存储, 可能暂时使用寄存器中的值, 如果这个变量有别的程序更新了的话, 将出现不一致的现象, 该驱动将寄存器地址值都用 volatile 关键词进行修饰, 以避免寄存器内值不能及时更新的问题。

```

typedef struct
{
    volatile uint32_T * txFifo_0; //pci429 发送通道 fifo
    ...
    volatile uint16_T * bps_sel; //dei1016 波特率控制
    volatile uint8_T * recEn; //8 路接收使能
} Regs_429;

```

3.2 飞控计算机系统部分

飞控计算机处理器板与 ARINC429 有关的硬件如图 4 所示。

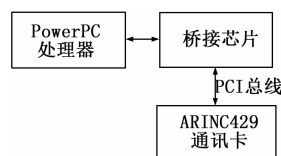


图 4 飞控计算机处理器板示意图

桥接芯片是为 PowerPC 专门设计的桥芯片/存储器控制器, 主要实现 PowerPC 到 PCI 的桥接功能, 同时管理内存 VxWorks 是 WindRiver 公司推出的一个高效的嵌入式实时操作系统, 内核可根据实际需求进行裁减, 拥有良好的可靠性和卓越的实时性。由于在不同 CPU 架构上提供了统一的接口和相同的运行特性, VxWork 的应用程序可以在不同的硬件上进行移植, 被广泛地应用在通信、军事、航空、航天等领域中^[3-4]。

WindRiver 公司从 VxWork6. X 开始引入 VxBus 驱动模型, 该模型定义了一些标准的接口来操作系统和硬件设备, VxBus 驱动模型涉及设备、驱动和实例。每一个驱动可以与多个设备关联, 但是每个实例只能对应其中的一种关联^[5], VxBus 驱动实例模型如图 5 所示。

VxBus 架构下 arinc429 通讯卡的初始化主要包括硬件查找阶段、驱动注册阶段、驱动初始化阶段 1、驱动初始化阶段 2 和驱动初始化阶段 3 五个部分。在硬件查找阶段, vxworks 系统调用 sysHwInit 函数激活设备, 在 sysHwInit 函数中, 通过调用 hardWareInterFaceInit 函数和 hardWareInterFaceBusInit 函数来初始化和激活 arinc429 通讯卡。在 hardWareInterFaceBusInit 函数, 通过调用 vxbDevRegister 函数来完成 arinc429 通讯卡驱动的注册。

(下转第 180 页)

大于其余 3 个频点的现象。该微波转换开关可以通过调节螺钉,使滤波器中心谐振频率在 20 MHz 范围内进行调整,可实现两分波道同频或异频,以满足不同场合的需求。

4 结束语

本文设计了一种可调谐的 S 波段分波道微波转换开关, S 波段输入微波信号在控制信号的作用下可选择两路不同输出通道,并将两路腔体滤波器对输入微波信号进行选频、滤波,两分波道输出频率同频或异频都可,自由度较高,调谐频率范围可达 20 MHz。该微波转换开关具有大功率、窄通带、低损耗、高隔离度等优点,性能稳定,易于实现,满足某型机载数据链测试要求,且适应于环境恶劣的场所,具有较好的经济效益和军事效益。

~~~~~

(上接第 177 页)

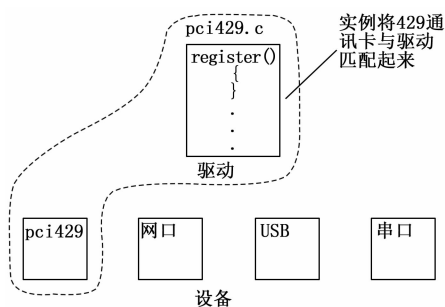


图 5 Vxbus 驱动实例模型

驱动程序注册主要包括 vxbDevRegInfo 结构体初始化和调用 vxbDevRegister 函数两部分, vxbDevRegInfo 结构体提供了驱动注册所需的相关信息,初始化代码如下:

```
LOCAL struct vxbDevRegInfo pci429DevRegistration =
{
...
VXB_DEVICEID_DEVICE, /* devID */
VXB_BUSID_PCI, /* busID = PLB */
"arinc429", /* drvName */
&pci429Funcs, /* pDrvBusFuncs */
pci429DriverCtrl, /* pMethods */
...
};
```

驱动初始化 3 个阶段主要完成内存空间、I/O 空间和中断等资源的申请以及实例的建立,使上层应用程序可以正常使用驱动程序。在 pci429DevRegistration 结构中 pci429Funcs 提供了设备驱动初始化 3 个阶段的入口地址,定义如下:

```
LOCAL struct drvBusFuncs pci429Funcs =
{
pci429InstInit, /* devInstanceInit */
pci429InstInit2, /* devInstanceInit2 */
pci429InstConnect /* devConnect */
};
```

在 pci429InstInit2 函数中主要负责获取硬件设备相关的资源, vxbRegMap 函数进行板卡基地址映射,从而调用相关的读写函数对设备卡进行操作。

代码如下:

```
LOCAL void wrsampleInstInit2(VXB_DEVICE_ID pInst)
{
```

## 参考文献:

- [1] 杨健. W 波段单刀单掷开关和单刀双掷开关 [D]. 成都: 电子科技大学, 2005.
- [2] 刘呈斌, 徐军, 罗慎独. 高隔离微带 PIN 单刀单掷开关 [A]. 2003 年全国微波毫米波会议 [C]. 2003: 307-310.
- [3] 袁婷婷, 陈晓娟, 陈中子, 等. 隔离度大于 95dB 的 Ku 波段微带型开关 [J]. 半导体学报, 2008, 29 (10): 2034-2037.
- [4] 张华, 孙健, 陶若燕. C 波段窄带腔体滤波器设计 [A]. 2007 年全国微波毫米波会议论文集 [C]. 2007, 10: 1135-1138.
- [5] 张晓齐. 圆杆梳状及交指带通滤波器的计算机辅助设计 [J]. 微波, 1985, 4: 16-26.
- [6] 马慧瑾, 刘忠健等. 一种可调谐的窄带圆杆梳状腔体滤波器的设计 [J]. 计算机测量与控制, 2013, 21 (11): 3083-3086.

...

```
VXB_PCI_BUS_CFG_READ (pDev, PCI_CFG_VENDOR_ID, 2,
&venId);
VXB_PCI_BUS_CFG_READ (pDev, PCI_CFG_DEVICE_ID, 2,
&devId);
/* find the memory mapped window for the device registers */
for (i = 0; i < VXB_MAXBARS; i++)
{
if (pInst->regBaseFlags[i] == VXB_REG_MEM)
break;
}
pDrvCtrl->pci429Bar = pInst->pRegBase[i]; /* store the
base address */
vxbRegMap (pInst, i, &pDrvCtrl->pci429Handle); /* map the
window */
...
}
```

在 pci429InstConnect 函数中完成硬件启动时相关资源的配置,包括中断部分、发送部分部分、接收部分以及状态命令字段。

## 4 试验结果与分析

EFIS 实时准确的更新飞艇模型发送的飞行状态信息,并且飞艇模型随着飞控计算机操作命令的不同,实时准确的变换着飞行姿态。实验证明, ARINC429 通讯卡驱动程序保证了飞艇仿真系统稳定、可靠和实时性的通信。

## 5 结论

飞艇仿真系统模型计算机和飞控计算机分别采用 xpc 系统和 vxworks 系统, ARINC429 驱动程序保证仿真系统在不同操作系统之间的实时通信,同时 ARINC429 驱动程序为上层应用程序访问 ARINC429 通信卡提供基础。

## 参考文献:

- [1] 王蒙. 基于嵌入式系统的 SpaceWire 总线驱动技术研究 [J]. 计算机测量与控制, 2014, 22 (2): 483-485.
- [2] 宋炜, 祖家奎. 基于 x P C 目标的实时 I/O 接口技术研究 [J]. 计算机应用, 2008, 27 (1): 62-65.
- [3] 李焱. VxWorks 下 82C55A 并口驱动系统设计及测试 [J]. 计算机测量与控制, 2010, 18 (2): 261-263.
- [4] 曹桂平. Vxworks 设备驱动开发详解 [M]. 北京: 电子工业出版社, 2011.
- [5] Wind River System Inc.. VxWorks programmer's guide [Z]. Wind River System Inc., 1999.