

# VxWorks 下现场总线 CAN 驱动实现与改进

周 宾, 潘 俊, 蒋金松

(中国电子科技集团公司第二十八研究所, 江苏 南京 210007)

**摘要:** 由于车辆计算机集成制造系统的日益发展和控制系统的渐渐复杂化, 要实现通过车内电脑终端对每个设备运行状态进行监视和控制, 就需要建立上层信息网络与底层控制网络的纵向集成。集散式控制系统已变得不能满足发展的需求, 而现场总线技术为解决这个问题提供了途径。文中介绍了在 VxWorks 下 CAN 驱动的实现, 并通过实验对数据传输稳定性进行测试, 通过采用中断-轮询方式对数据接收进行改进, 提高了车辆车内控制数据传输的稳定性。

**关键词:** CAN; 现场总线; 驱动; 改进

中图分类号: TP31

文献标识码: A

文章编号: 1673-629X(2014)05-0188-03

doi: 10.3969/j.issn.1673-629X.2014.05.046

## Realization and Improvement of CAN Field-bus's Driver for VxWorks

ZHOU Bin, PAN Jun, JIANG Jin-song

(The 28th Institute of China Electronics Technology Group Corporation, Nanjing 210007, China)

**Abstract:** With the development of vehicle computer integration manufacture system and the control system complicated gradually, to achieve the monitoring and control through the in-car computer for each equipment running status, need to establish the vertical integration between information network and bottom layer control network. The concentrate control system is not able to meet the requirement of modern vehicle, however, the filed-bus control system has provided a solution to this difficulty. To ensure the stability of control data in the vehicle's severe environment, introduce the realization of CAN field-bus driver in VxWorks embedded real-time OS, and through testing data transmission, improve the way of data reception by interrupt-polling.

**Key words:** CAN; field bus; driver; improvement

## 0 引言

由于车辆系统内部集成众多设备, 位于车辆不同位置的设备通过现场总线互连起来, 构成了底层的网络<sup>[1]</sup>。如何通过信息网与控制网集成, 实现车辆设备高效有序的运行, 成为车辆控制系统的重中之重。在该嵌入式系统中, 所采用的操作系统是 VxWorks, 它由 WindRiver 公司开发, 在实时操作系统(Real Time Operation System, RTOS)领域具有领导地位<sup>[2]</sup>, 是一个具有可裁剪、可伸缩和高可靠性的实时操作系统; 该系统选用的 CAN 控制器 MCP2510, 它内含二个接收缓冲器、三个发送缓冲器, 具有灵活的中断管理能力<sup>[3]</sup>。

## 1 VxWorks 下 CAN 驱动设计

### 1.1 VxWorks 设备驱动分析

VxWorks 操作系统实现驱动方式有两种:

① VxWorks 的 I/O 系统把设备驱动程序作为内核过程实现, 该方式可以大大减少系统开销, 提高系统的可靠性和实时性。

② 直接在顶层任务中实现硬件操作, 把设备驱动程序作为独立任务来实现, 完成特有的专用驱动程序<sup>[4]</sup>。

该系统采用通过 I/O 系统存取的标准设备驱动机制。

### 1.2 CAN 驱动的实现

CAN 设备驱动的基本实现例程主要包括设备驱动的安装、设备的创建、设备的打开、报文收发、中断服务和 I/O 控制等<sup>[5]</sup>。

(1) 驱动安装函数。

驱动的安装函数需要实现的内容为: 通过调用 iosDrvInstall() 函数将驱动程序的入口位置添加到系统的驱动程序表中; 完成一些必要的初始化。

将驱动程序添加到驱动程序描述表中,部分伪代码如下:

```
STATUS CanDrv()
{
    .....
    if((CanDriverNum = iosDrvInstall(Can_Open, NULL, Can_Open, Can_Close, Can_Read, Can_Write, Can_Ioctl) == ERROR))
    {
        .....
    }
}
```

(2) 设备创建函数。

在驱动程序安装完后,下一步就是创建设备,创建过程为:

- ①判断是否已经安装驱动程序,如未安装立即返回错误;
- ②为设备描述符分配空间并将其初始化;
- ③调用 iosDevAdd() 函数将设备添加到系统设备描述表中;
- ④最后执行一些设备相关的初始化<sup>[6]</sup>。

实现过程部分代码如下:

```
STATUS Can_Dev_Create(char * dev_Name,int arg... )
{
    Can_DEV * p_CanDev; /* 设备描述符 */
    if((p_CanDev=(Can_DEV *) malloc( sizeof( Can_DEV)))
    ==NULL)
    return(ERROR);
    .....
    /* 将设备添加到设备列表 */
    if( iosDevAdd( &p_CanDev->devHdr, devName, CanDrv-Num) == ERROR)
    {
        .....
    }
}
```

(3) 打开设备 Can\_Open()。

设备打开操作通常和设备创建操作使用相同的函数。打开操作会执行一些设备的初始化或者对设备参数进行设置。如果执行成功,返回设备的标志,这个标志通常是指向设备描述符的指针 p\_DevHdr;如果操作失败则会返回 ERROR。

(4) 设备读(Can\_Read)/写(Can\_Write)函数。

读写的数据流层次关系如图 1 所示。

应用层调用 ioLib 中的 Read 和 Write 函数,这些函数从输入的 fd 中找到对应的设备描述符,然后找到 DRV\_TABLE 表,调用 DRV\_TABLE 中对应的函数。DRV\_TABLE 表中的函数则是第二层的 CAN 驱动例程(Can\_Read、Can\_Write),通过 Can\_Read\Can\_Write 和真正的 CAN 设备打交道。通过中断回调函数(CanIRx、CanITx)来实现第二层和底层的通信<sup>[7]</sup>。具体地说,中断服务程序负责数据的收发。

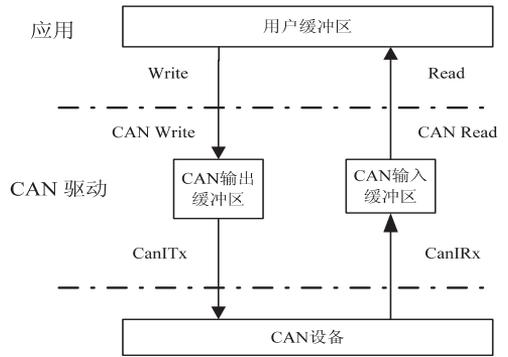


图 1 数据流关系图

(5) 中断服务 Can\_Intr()。

中断服务通过 Can\_Intr() 读 CAN 控制器的中断寄存器识别中断源,然后根据不同中断源分别进入不同工作流程。中断源是由控制器复位时中断使能寄存器的值决定的。该系统只打开了发送、接收和数据溢出三种中断<sup>[8]</sup>。

(6) I/O 控制 Can\_Ioctl()。

CAN 设备通过 I/O 请求控制代码的类型来完成 CAN 设备的控制,主要包括读取和写入缓冲器现有的数据帧数量;清空 CAN 设备的读或写缓冲器;按请求的大小重新分配 CAN 设备的读\写缓冲器;设置 CAN 设备总线波特率。

## 2 CAN 驱动改进

在驱动的实际应用过程中,发现当短时间(毫秒级)接收大量数据时,丢包现象比较严重,于是对驱动程序的数据接收处理模块进行研究改进。改进前的数据接收采用的是中断查询方式,它的工作原理:每当接收寄存器接收到数据,中断寄存器的接收中断使能,并产生硬件中断信号通知 CPU 有数据已经接收到,进入数据接收读取状态,在数据读取后,清空中断标识位,等待下一次的接收中断<sup>[9]</sup>。其处理流程如图 2 所示。

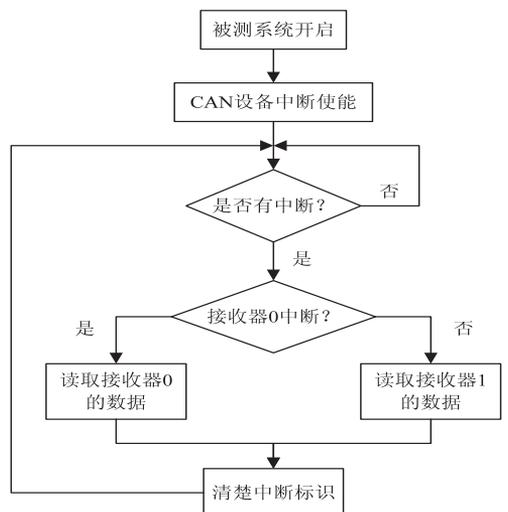


图 2 中断处理流程

由于改进前的中断每次只能处理一个接收寄存器的中断响应,而采用的 CAN 控制器 MCP2510,它内含二个接收缓冲器<sup>[10]</sup>,当 2 个接收缓冲区都有数据时,传统中断方式只能读取完接收器 0 数据后,将中断标识位清空,再等待接收器 1 的中断。由于中断处理优先级高,该系统改进的 CAN 驱动以中断处理方式为主,结合轮询方式,在处理接收器中断响应后,对另一个接收器的状态进行轮询处理,这样大大提高了 CAN 控制器的数据接收处理能力,其处理流程如图 3 所示。

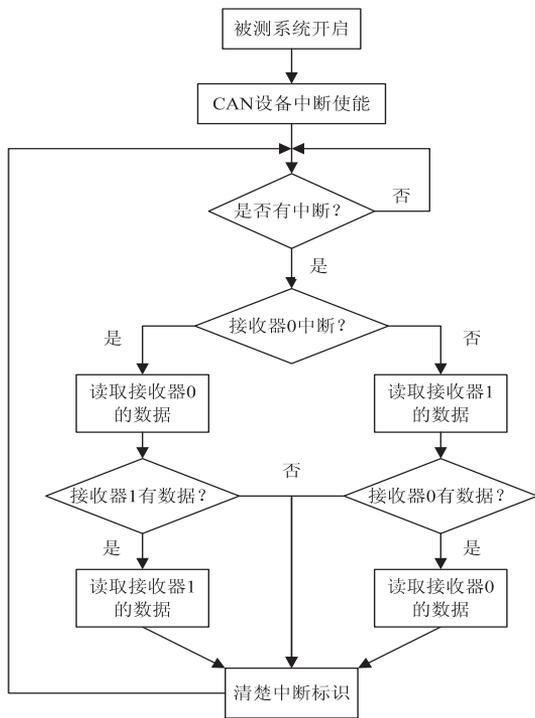


图 3 中断-轮询方式处理流程

### 3 测试环境搭建及结果分析

本节通过实验对比 CAN 分别以传统的中断方式接收和文中改进后的中断-轮询方式接收的运行性能。测试环境如下:测试硬件环境由 CAN 数据报文发送器和被测机器组成<sup>[11]</sup>。其中,被测机器的实际硬件由车载计算机、一块 MCP2510-CAN 接收芯片设计的电路板和一条 RS232 串口线组成;数据报文发生器采用周立功的 USB-CAN-II 型设备,它是一款功能强大的 CAN 网络开发测试工具<sup>[12]</sup>。测试系统发送数据的操作由 USB-CAN-II 完成,它按一定的频率向被测机器发送数据包。在测试实验中,报文发送器 USB-CAN-II 分别以 10 B/ms、20 B/ms、40 B/ms、80 B/ms、200 B/ms、400 B/ms、600 B/ms、800 B/ms、1 000 B/ms 的速率向被测平台发送数据。被测平台将接收到的数据个数通过串口打印到 PC 机上。其接收情况的曲线图如图 4 所示。

图中,横坐标表示 USB-CAN-II 发送的数据包数

目,纵坐标显示接收到的数据帧占总发出帧的百分比。由数据曲线可以分析出,使用传统的中断接收方式,在接收快速短帧 CAN 数据包时,如果每毫秒发送数据包个数达到 1 000,丢包率达 40% 左右;而采用中断-轮询的接收方式,较大地降低了丢包率。造成上述结果的主要原因是:如果 CAN 设备采用单次中断接收时,处理器要在应用程序和中断程序之间频繁切换,在切换过程中容易造成丢包。而采用中断-轮询机制后,切换次数减少,所以丢失的数据包变少了。

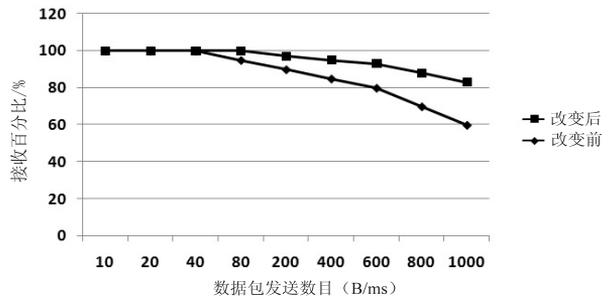


图 4 数据接收对比图

### 4 结束语

文中根据实际项目出发,介绍了 VxWorks 操作系统在 x86 平台基础上 CAN 设备驱动的实现,给其他在该架构下编写 VxWorks 设备驱动提供了基本的思路。同时,对 MCP2510 的驱动程序进行深入研究,通过搭建实验环境,对改进前后的驱动进行高强度的测试,采用改进的中断-轮询方式,有效地降低数据丢包率,对车内现场总线数据传输稳定性有直接的帮助。

#### 参考文献:

- [1] 张 民. 浅谈现场总线及发展前景[J]. 青海科技, 2005 (1):54-57.
- [2] 张敏燕. 基于嵌入式实时操作系统 VxWorks 平台的分析与研究[D]. 南京:南京理工大学,2007.
- [3] Microchip. MCP2510 stand-alone CAN controller with SPI-interface[M]. [s. l.]:Microchip,2004.
- [4] 周启平,张 杨. VxWorks 下设备驱动程序及 BSP 开发指南[M]. 北京:中国电力出版社,2004.
- [5] River W. VxWorks BSP 开发人员指南[M]. 北京:清华大学出版社,2003.
- [6] 李延伟,陈欠根,朱 毅. 嵌入式操作系统 VxWorks 下 CAN 驱动程序开发[J]. 工业控制计算机,2006,19(10):30-32.
- [7] 沈连丰. 嵌入式系统及其开发应用[M]. 北京:电子工业出版社,2005.
- [8] Hanzalek Z, Schmieder E, Wenzel P. Creation of the laboratory for fieldbus-based automation systems[C]//Proc of second IEEE real-time education workshop. [s. l.]:[s. n.],1997.

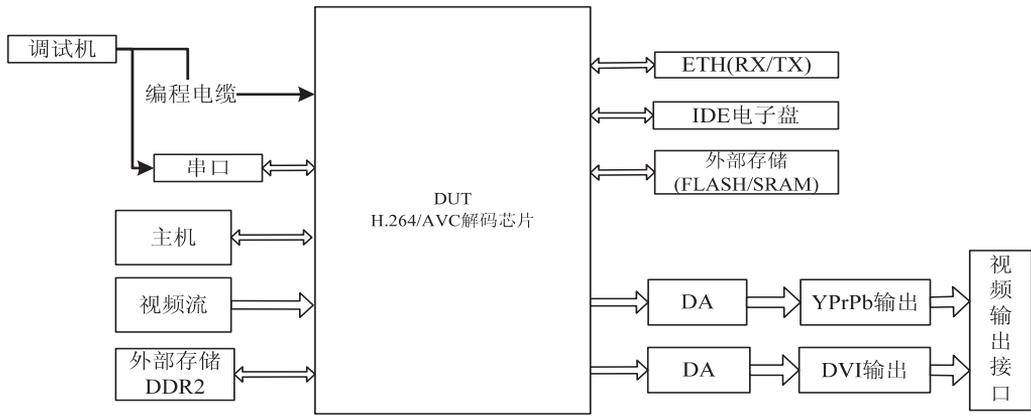


图 3 FPGA 原型验证平台

有的验证项,表 2 给出了部分验证项的策划结果。

表 2 部分验证项的策划结果

功能点	输入码流	是否在虚拟平台验证	是否在 FPGA 原型平台验证	是否在后仿真验证
支持帧模式解码	H264Dec001. 264	是	是	是
支持 CAVLC	H264Dec002. 264	是	是	是
支持 P 帧解码	H264Dec003. 264	是	是	是
码流输入接口输入码流	H264Dec004. 264	是	否	是
以太网终端输入码流	H264Dec005. 264	是	是	否
IDE 记录接口输入码流	H264Dec006. 264	是	是	否
视频流分辨率为 352×288	H264Dec007. 264	是	否	否

## 5 结束语

H. 264/AVC 压缩算法运算量大,对应功能点复杂,文中的解码器涉及三种数据通路,指出八种分辨率的视频源,给验证工作带来了很大挑战。文中在充分了解解码器芯片功能点的基础上,对其功能点进行分级、划分,结合多种验证手段,在解码器设计的不同阶段对其进行充分验证。在节约验证资源的同时,提高了验证效率,缩短了芯片的设计周期。

## 参考文献:

[1] 毕厚杰. 新一代视频压缩编码标准-H. 264/AVC[M]. 北京:人民邮电出版社,2005.  
 [2] 郭鹏飞. SoC 设计中的软硬件协同设计[J]. 今日电子, 2004(6):73-75.  
 [3] Rashinkar P. System-on-a-chip verification methodology and techniques[M]. [s. l.]:Kluwer Academic Publishers,2001.

[4] Vcodex. H. 264/MEPG-4 part 10 white paper[EB/OL]. 2004-07-04. <http://www.vcodex.com>.  
 [5] Chen T C,Chien S Y,Huang Y W, et al. Analysis and architecture design of an HDTV720p 30 frames/s H. 264/AVC encoder[J]. IEEE Trans on Circuits and Systems for Video Technology,2006,16(6):673-688.  
 [6] Sheng Bin,Gao Wen,Wu Di. An implemented architecture of deblocking filter for H. 264/AVC[C]//Proc of 2004 international conference on image processing. Singapore:IEEE Press, 2004:665-668.  
 [7] Masselos K,Merakos P,Stouraitis T, et al. A novel algorithm for low-power image video coding[J]. IEEE Trans on Circuits and System for Video Technology,1998,8(3):258-263.  
 [8] Joint Video Team. ITU-T Rec. H. 264|ISO/IEC 14496-10, draft ITU-T recommendation and final draft international standard of joint video specification[S]. 2003.  
 [9] Tourapis A M. H. 264/MPEG-4 AVC reference software manual[M]. [s. l.]:Dolby Laboratories Inc,Fraunhofer-Institute HHI,Microsoft Corporation,2007.  
 [10] 张力航,林涛,周开伦. 软硬件协同设计技术在 H. 264 解码器设计中的应用[J]. 电子设计应用,2006(9):89-90.  
 [11] 田靖,田泽. AFDX-ES SoC 虚拟仿真平台的构建与应用[J]. 计算机技术与发展,2010,20(8):192-194.  
 [12] 桑圣锋,张德学,于国苹. AEMB 软核处理器的 SoC 系统验证平台[J]. 单片机与嵌入式系统应用,2010(4):43-45.  
 [13] 虞致国,魏敬和. 基于 FPGA 的 ARM SoC 原型验证平台设计[J]. 电子与封装,2007(5):25-28.  
 [14] 李建昌,赵博生,鲍胜荣,等. 基于 ARM7TDMI 的 SoC 芯片的 FPGA 验证平台设计[J]. 电子工程师,2004,30(9):6-9.

(上接第 190 页)

[9] Yang Le,Yang Peng,Tian Aiyong, et al. Research on multi-parameter physiological monitor based on CAN bus[J]. IF-MBE Proceedings,2008,19:417-419.  
 [10] 李貌,秦霆镐,闫世晓. MCP2515 在 CAN 总线系统智能节点的应用[J]. 微计算机信息,2005,21(06Z):37-39.  
 [11] Hasnaoui S,Kallel O,Kbaier R. An implementation of a pro-

posed modification of CAN protocol on CAN fieldbus controller component for supporting a dynamic priority policy[C]// Proc of industry application. [s. l.]:IEEE Press,2003:23-31.  
 [12] 王继国,孙新亚. CAN 控制器芯片 MCP2510 在远程监测系统中的应用[J]. 电子技术应用,2004(4):77-80.

## VxWorks下现场总线CAN驱动实现与改进

作者: 周宾, 潘俊, 蒋金松, ZHOU Bin, PAN Jun, JIANG Jin-song  
作者单位: 中国电子科技集团公司第二十八研究所, 江苏 南京, 210007  
刊名: 计算机技术与发展   
英文刊名: Computer Technology and Development  
年, 卷(期): 2014(5)

本文链接: [http://d.wanfangdata.com.cn/Periodical\\_wjfz201405046.aspx](http://d.wanfangdata.com.cn/Periodical_wjfz201405046.aspx)