

## 基于 VxBus 的高速数据采集卡驱动程序开发 \*

任申真<sup>1,2</sup> 谭亮<sup>1,2</sup> 李一平<sup>1,2</sup> 封锡盛<sup>1</sup>

(<sup>1</sup>中国科学院沈阳自动化研究所 机器人学国家重点实验室 沈阳 110016)

(<sup>2</sup>中国科学院 研究生院 北京 100049)

Driver development based on VxBus for data acquisition card

REN Shen-zhen<sup>1,2</sup>, TAN Liang<sup>1,2</sup>, LI Yi-ping<sup>1,2</sup>, FENG Xi-sheng<sup>1</sup>

(<sup>1</sup>State key laboratory of robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China)

(<sup>2</sup>Graduate School of Chinese Academy of Sciences, Beijing 100049, China)

**【摘要】**介绍了嵌入式操作系统 VxWorks 6 最新设备驱动模式 VxBus 的原理,分析了在该驱动模式下数据采集卡 ADT650 的驱动程序开发过程,ADT650 数据采集卡是基于 PC104 总线的扩展板。通过 PC104 总线可将其与 PC104 嵌入式系统构成一个高性能的数据采集与控制系统,适用于结构紧凑,高可靠地嵌入式应用。设计并实现了数字量、模拟量信号采集的驱动程序和应用编程接口。在 VxWorks 嵌入式系统下实现了 ADT650 高速数据采集卡的驱动程序,对系统的扩展应用带来了极大方便。

**关键词:** VxWorks 数据采集卡 VxBus PC104 应用编程接口

**【Abstract】** The principle of drive mode VxBus for latest equipment in the embedded OS VxWorks is introduced, then the driver development for the Data acquisition card in the mode of VxBus driver is analyzed. AD650 data acquisition card is an expansion board based on PC104 bus which is integrated with PC104 embedded system to form a data acquisition and control system with high performance through PC104 bus to be applied in embedded system with compact structure and high reliability. Moreover the driver and application programming interface of acquisition multi-channel digital quantity and analog quantity is designed and realized. The realization of driver for ADT650 high speed data acquisition card under VxWorks embedded system will provide convenience for the expansion of the system.

**Key words:** VxWorks Data acquisition card VxBus PC104 API

中图分类号: TH16, TP24 文献标识码: A

### 1 前言

采用 PC104 主板运行 VxWorks 嵌入式操作系统的方式构建了水下机器人控制系统。VxWorks 是一个分布式、多用户、多任务的嵌入式实时操作系统,它具有极高的安全性、可靠性、稳定性。从设计之初,它被广泛应用于通信、国防、航空等重要领域。水下机器人控制系统需要跟与各种传感器设备交互实现设备的控制和系统监测,主要包括大量数字信号和模拟信号。PC104 主板自身的外围接口难以满足机器人控制系统的需求。利用 PC104 总线扩展一块 ADT650 高速数据采集卡,采集卡具有的丰富接口可以满足系统需要。为了 ADT650 数据采集卡的通用性,基于 VxWorks 系统开发了采集卡的驱动程序和应用编程接口。

### 2 VxWorks 驱动模式

#### 2.1 设备驱动程序

设备驱动程序提供连接到计算机硬件的软件接口,它是操作系统的信任部分。设备驱动程序就是操作系统内核和机器硬件之间的接口,它为应用程序屏蔽了硬件的细节,在应用程序看来,硬件设备只是一个设备文件,应用程序可以像操作普通文件一样对硬件设备进行操作。

#### 2.2 VxBus 驱动模式介绍

VxWorks 操作系统自从 6.2 版本以来,它对驱动模式做了很

大改进,采用全新的 VxBus 模式进行驱动程序的设计,对高速数据采集卡 ADT650 的驱动程序开发就是在此模式下实现的。

VxWorks 传统的驱动模式是基于单一设备的,而 VxBus 模式是基于虚拟总线的。VxWorks 传统驱动模式缺乏通用性。驱动程序和操作系统之间没有标准的接口,无法自己寻找已经存在的设备。由于驱动程序有一些代码保存在 BSP 中,当驱动设备增加很多时,会导致 BSP 复杂度增加。VxBus 驱动模式为操作系统调用驱动程序提供了标准的 API 函数。根据总线接口的不同,采取不同的策略去寻找安装到系统的硬件设备,比如 PLB (Processor Local Bus) 总线根据设备列表去安装新设备,而 PCI (Peripheral Component Interconnect) 根据设备地址自动去安装新设备。VxBus 在 BSP 中有统一的调用驱动接口函数,而驱动程序都在驱动程序文件中,增加新的设备对 BSP 的维护开销很小。

VxBus 驱动模式具有两层含义。首先, VxBus 指在最小 BSP 系统下支持设备驱动的一种特殊结构。它具有使设备驱动与硬件设备相匹配的功能,使驱动程序能够控制相应的硬件设备。操作系统的应用程序部分可以调用驱动程序来使硬件设备实现相应功能。另外, VxBus 有时指 VxWorks 操作系统可以在 Workbench 开发环境中调用的一套组件。VxBus 函数都以组件形式存放在操作系统中,每一个 VxWorks 的 VxBus 驱动函数都封装成一个相应的组件,并且 VxBus 驱动程序调用相关的模块也封装成独立的组件。

每一个组件都可以在 workbench 开发环境中独立加载或卸载。

设备、驱动和实例这三个术语对理解 VxBus 驱动模式很重要。设备就是提供相应功能的硬件设备。驱动指的是可执行代码段和配置信息,用来使操作系统控制硬件设备正常工作。每一个驱动程序可以挂接多个硬件设备。设备和驱动在一起构成了一个实例,如图 1 所示。

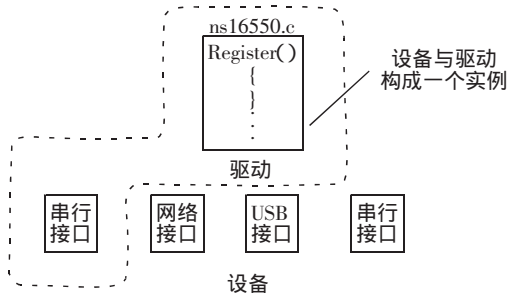


图 1 VxBus 实例构成

驱动方法形成一种使操作系统可以调用实例的机制,利用该机制操作系统的功能模块可以很方便的调用硬件设备。当使用驱动方法时,功能模块通过查询所有的实例来决定哪个实例可以满足功能模块的要求,并确定所要执行的操作。最终功能模块和满足要求的实例进行匹配。

## 3 ADT650 驱动开发

### 3.1 ADT650 数据采集卡

ADT650 数据采集卡是基于 PC104 总线的扩展板。通过 PC104 总线可将其与 PC104 嵌入式系统构成一个高性能的数据采集与控制系统,适用于结构紧凑,高可靠地嵌入式应用。

ADT650 数据采集卡具有 24 路可以软件配置输入输出方向的数字量 IO 口,16 路单端或 8 路差分输入的模拟输入量接口,以及 4 通道模拟输出接口,并且可以使能外部中断。它具有的丰富系统资源,可以满足水下机器人系统的数据交互要求。

水下机器人系统中使用 ADT650 数据采集卡一般有两种使用方式。一种是直接在应用程序中对硬件地址进行操作,通过读写寄存器对接口数据进行处理。这种方式实现起来比较简单,但是不利于程序的模块化编程,对以后程序的扩展及其移植都带来更多困难。另一种是比较常用的为扩展卡编写底层驱动程序的方式。驱动程序完成与底层硬件相关的操作,并封装成一个驱动程序模块,为用户提供相应的应用编程接口。需要使用 ADT650 采集卡时在加载驱动模块。应用程序需要对设备进行操作时只要调用驱动程序提供的 API 函数接口。这种方式层次分明,并且应用程序具有通用性,为扩展和移植带来了方便。就是采用编写驱动程序的方式。

由于采用的数据采集卡是 PC104 总线接口,VxBus 驱动模式下把该类型的接口定义为 PLB 类型的设备。对于 PLB 类型的设备,VxWorks 系统通过读取 BSP 提供的设备列表来确定安装的设备及设备的功能。首先,在硬件上通过设置数据采集卡跳线设定采集卡在系统中的物理地址为 0x300,然后在软件上通过 pc.h 中设定采集卡的基地址为 0x300。这样,VxWorks 系统便可以识别数据采集卡。

### 3.2 数字量 IO 口驱动程序

数字量 IO 口既可以作为数字量输入口也可以作为数字量输

出口。作为哪种功能可以通过软件设定。数字量有 24 路,每 8 路为一组,进行输入输出设定。通过驱动的封装函数 SetIODir(int group, int dir)来设定,参数 group 确定要设定 IO 口是 A B C 哪组,参数 dir 设定 IO 口的方向。IO 口作为输入口使用时,驱动程序以固定的采样周期不停地扫描输入 IO 口,然后将 IO 口状态存储在数据区。驱动程序采取两种方式将 IO 口状态传输给应用程序:查询方式和中断方式。应用程序可以通过函数 SelectMode(char mode)来设定应用程序使用何种方法来获得 IO 口状态。查询方式即应用程序需要知道 IO 口状态时,便通过查询函数 CheckState(char state)来获得 IO 口状态。驱动程序采集到 IO 口数据后只是保存在数据区,只有当收到查询命令后才将数据区内容传送给应用程序。

中断方式采用是驱动程序将采集到的数据与上次采集到的数据进行对比,当发现 IO 口数据发生变化后采用软件中断方式将数据发送给应用程序。为了减小外界电磁干扰的影响,一般以连续 3 次采集到一致数据作为判断标准。这种方式下,应用程序需要重新开辟一个线程来等待驱动程序发送的数据。这两种方式有各自的优势。查询方式,应用程序编程简单,占用系统资源少。而中断方式能够使系统可以及时的获得 IO 口的状态。用户可以根据需求选择使用哪种方式。IO 口作为输出方式使用时,应用程序只要通过函数 SetOutIO(char IO, char outData)设定输出值。驱动程序以 1 秒为周期更新输出口状态。应用程序调用数字量 IO 口驱动程序的方法,如图 2 所示。

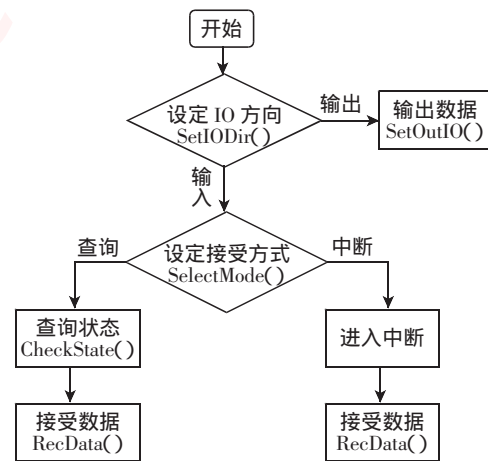


图 2 调用 IO 口驱动方法

### 3.3 模拟量驱动程序

ADT650 采用 12 位的模/数变换器,采样精度满足了水下机器人系统的要求。并且可以根据需要设定单极性或双极性的输入和单端或差分输入。这些参数与输入量程都是通过板卡的跳线来设定的,为了驱动程序的通用性,应用程序可以通过函数 SetParam(char polarity, char inmode, int range)来设定。驱动程序采用条件编译的方式针对不同的参数来选择相应的算法。

模拟量转换步骤,如图 3 所示。水下机器人系统需要采集多路来自传感器及相关设备的 A/D 转换数据。在选择输入通道时,采取连续的一组通道按顺序进行模数转换。在需要转换时,驱动程序对相应的通道进行设定。在当前通道转换完成后,程序会自动递增到选择范围内的下一个通道进行转换。使用户不用每次转换前都写入指定的通道。通道选择好后,向相应的寄存器写入数据开始 A/

D 转换。转换器大约在 10ms 内可以完成一次转换 用户可以查询转换状态寄存器来确定转换是否完成。伪码如下所示：

```
forever
{
Status=inp(base+8)&0x01;
If Status = 0
Then conversion is complete ,
Else A/D converter is busy.
}
```

A/D 转换完成后 需要分一个 8 位字节和一个 4 位字节读出

转换结果 步骤如下：

```
LSB=inportb(base); //Get low 8 bits
MSB=inportb(base+1) & 0x0f; //Get high 4 bits
Data=MSB * 256+LSB ;
//Combine the 2 bytes into a 12-bit value
```

得到 A/D 变换值后 需要将其转换成有意义的值。驱动程序根据应用程序之前的设定参数将该值转换成实际测量的电压值，然后将电压值发送给应用程序。

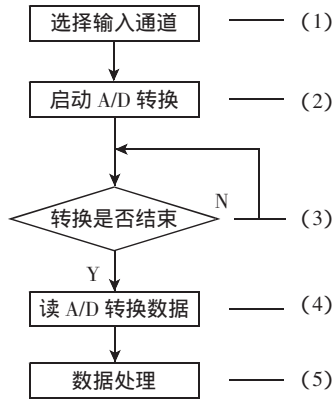


图 3 A/D 转换步骤

应用程序调用模拟量驱动程序的方法 如图 4 所示。应用程序接受 A/D 转换数据的方式与数字量 IO 口的驱动程序相似，也采取查询模式和中断模式。

### 3.4 驱动加载进工程

VxBus 驱动模式下，用户编写的驱动程序存放在第三方驱动程序目录下。需要在系统 image 文件的 void usrAppInit(void)函

数中加载 ADT650 采集卡的驱动程序。当 VxWorks 系统启动后，系统会自动加载采集卡的驱动程序。应用程序便可以使用驱动程序封装的 API 函数。

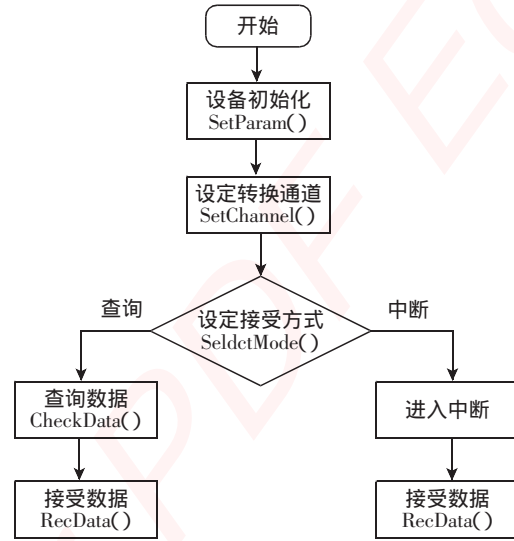


图 4 调用模拟量驱动程序

## 4 总结

在 VxWorks 嵌入式系统下实现了 ADT650 高速数据采集卡的驱动程序 对系统的扩展应用带来了极大方便。该方法不仅可以满足了水下机器人控制系统对大量数字量及模拟量数据采集的要求，还可以方便的扩展到其它使用 VxWorks 操作系统的应用当中 具有很高的实用价值。

### 参考文献

- [1] 孔祥营. 嵌入式实时操作系统 VxWorks 机器开发环境 Tornado[M]. 北京: 中国电力出版社, 2002(1).
- [2] VxWorks programmer's guide version 3.0 [Z]. Wind River System Inc., 2007.
- [3] Technological manual data acquisition card of ADT650 [Z]. SenBo Corporation, 2005.
- [4] 余芳, 卜正良, 舒乃秋, 章涛. 基于 PC104 总线的 28 通道同步数据采集卡[J]. 电测与仪表, 2003(12).
- [5] 唐明明, 邵根富, 高健. 多通道同步 A/D 采集卡的设计与实现[J]. 中国水运(学术版), 2007(7).
- [6] 姚松, 杨兆选. 数据采集卡的研制和应用[J]. 自动化与仪表, 2001(1).

# 嵌入式资源免费下载

## 总线协议:

1. [基于 PCIe 驱动程序的数据传输卡 DMA 传输](#)
2. [基于 PCIe 总线协议的设备驱动开发](#)
3. [CANopen 协议介绍](#)
4. [基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计](#)
5. [FPGA 实现 PCIe 总线 DMA 设计](#)
6. [PCI Express 协议实现与验证](#)
7. [VPX 总线技术及其实现](#)
8. [基于 Xilinx FPGA 的 PCIE 接口实现](#)
9. [基于 PCI 总线的 GPS 授时卡设计](#)
10. [基于 CPCI 标准的 6U 信号处理平台的设计](#)
11. [USB30 电路保护](#)
12. [USB30 协议分析与框架设计](#)
13. [USB 30 中的 CRC 校验原理及实现](#)
14. [基于 CPLD 的 UART 设计](#)
15. [IPMI 在 VPX 系统中的应用与设计](#)
16. [基于 CPCI 总线的 PMC 载板设计](#)
17. [基于 VPX 总线的工件台运动控制系统研究与开发](#)
18. [PCI Express 流控机制的研究与实现](#)
19. [UART16C554 的设计](#)
20. [基于 VPX 的高性能计算机设计](#)
21. [基于 CAN 总线技术的嵌入式网关设计](#)
22. [Visual C 串行通讯控件使用方法与技巧的研究](#)
23. [IEEE1588 精密时钟同步关键技术研究](#)
24. [GPS 信号发生器射频模块的一种实现方案](#)
25. [基于 CPCI 接口的视频采集卡的设计](#)
26. [基于 VPX 的 3U 信号处理平台的设计](#)
27. [基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计](#)
28. [AT89C52 单片机与 ARINC429 航空总线接口设计](#)
29. [基于 CPCI 总线多 DSP 系统的高速主机接口设计](#)

## VxWorks:

1. [基于 VxWorks 的多任务程序设计](#)
2. [基于 VxWorks 的数据采集存储装置设计](#)
3. [Flash 文件系统分析及其在 VxWorks 中的实现](#)
4. [VxWorks 多任务编程中的异常研究](#)
5. [VxWorks 应用技巧两例](#)
6. [一种基于 VxWorks 的飞行仿真实时管理系统](#)
7. [在 VxWorks 系统中使用 TrueType 字库](#)
8. [基于 FreeType 的 VxWorks 中文显示方案](#)
9. [基于 Tilcon 的 VxWorks 简单动画开发](#)
10. [基于 Tilcon 的某武器显控系统界面设计](#)
11. [基于 Tilcon 的综合导航信息处理装置界面设计](#)
12. [VxWorks 的内存配置和管理](#)
13. [基于 VxWorks 系统的 PCI 配置与应用](#)
14. [基于 MPC8270 的 VxWorks BSP 的移植](#)
15. [Bootrom 功能改进经验谈](#)
16. [基于 VxWorks 嵌入式系统的中文平台研究与实现](#)
17. [VxBus 的 A429 接口驱动](#)
18. [基于 VxBus 和 MPC8569E 千兆网驱动开发和实现](#)
19. [一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法](#)
20. [基于 VxBus 的设备驱动开发](#)
21. [基于 VxBus 的驱动程序架构分析](#)

## Linux:

1. [Linux 程序设计第三版及源代码](#)
2. [NAND FLASH 文件系统的设计与实现](#)
3. [多通道串行通信设备的 Linux 驱动程序实现](#)
4. [Zsh 开发指南-数组](#)
5. [常用 GDB 命令中文速览](#)
6. [嵌入式 C 进阶之道](#)
7. [Linux 串口编程实例](#)
8. [基于 Yocto Project 的嵌入式应用设计](#)
9. [Android 应用的反编译](#)
10. [基于 Android 行为的加密应用系统研究](#)
11. [嵌入式 Linux 系统移植步步通](#)
12. [嵌入式 C++ 语言精华文章集锦](#)
13. [基于 Linux 的高性能服务器端的设计与研究](#)
14. [S3C6410 移植 Android 内核](#)
15. [Android 开发指南中文版](#)

16. [图解 Linux 操作系统架构设计与实现原理（第二版）](#)
17. [如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核](#)
18. [Android 简单 mp3 播放器源码](#)
19. [嵌入式 Linux 系统实时性的研究](#)
20. [Android 嵌入式系统架构及内核浅析](#)
21. [基于嵌入式 Linux 操作系统内核实时性的改进方法研究](#)
22. [Linux TCP IP 协议详解](#)
23. [Linux 桌面环境下内存去重技术的研究与实现](#)

## Windows CE:

1. [Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计](#)
2. [Windows CE 的 CAN 总线驱动程序设计](#)
3. [基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究](#)
4. [基于 Windows CE.NET 平台的串行通信实现](#)
5. [基于 Windows CE.NET 下的 GPRS 模块的研究与开发](#)
6. [win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码](#)
7. [Windows 下的 USB 设备驱动程序开发](#)
8. [WinCE 的大容量程控数据传输解决方案设计](#)
9. [WinCE6.0 安装开发详解](#)
10. [DOS 下仿 Windows 的自带计算器程序 C 源码](#)
11. [G726 局域网语音通话程序和源代码](#)
12. [WinCE 主板加载第三方驱动程序的方法](#)
13. [WinCE 下的注册表编辑程序和源代码](#)
14. [WinCE 串口通信源代码](#)
15. [WINCE 的 SD 卡程序\[可实现读写的源码\]](#)
16. [基于 WinCE 的 BootLoader 研究](#)

## PowerPC:

1. [Freescale MPC8536 开发板原理图](#)
2. [基于 MPC8548E 的固件设计](#)
3. [基于 MPC8548E 的嵌入式数据处理系统设计](#)
4. [基于 PowerPC 嵌入式网络通信平台的实现](#)
5. [PowerPC 在车辆显控系统中的应用](#)
6. [基于 PowerPC 的单板计算机的设计](#)

7. [用 PowerPC860 实现 FPGA 配置](#)
8. [基于 MPC8247 嵌入式电力交换系统的设计与实现](#)
9. [基于设备树的 MPC8247 嵌入式 Linux 系统开发](#)
10. [基于 MPC8313E 嵌入式系统 UBoot 的移植](#)

## ARM:

1. [基于 DiskOnChip 2000 的驱动程序设计及应用](#)
2. [基于 ARM 体系的 PC-104 总线设计](#)
3. [基于 ARM 的嵌入式系统中断处理机制研究](#)
4. [设计 ARM 的中断处理](#)
5. [基于 ARM 的数据采集系统并行总线的驱动设计](#)
6. [S3C2410 下的 TFT LCD 驱动源码](#)
7. [STM32 SD 卡移植 FATFS 文件系统源码](#)
8. [STM32 ADC 多通道源码](#)
9. [ARM Linux 在 EP7312 上的移植](#)
10. [ARM 经典 300 问](#)
11. [基于 S5PV210 的频谱监测设备嵌入式系统设计与实现](#)
12. [Uboot 中 start.S 源码的指令级的详尽解析](#)
13. [基于 ARM9 的嵌入式 Zigbee 网关设计与实现](#)
14. [基于 S3C6410 处理器的嵌入式 Linux 系统移植](#)

## Hardware:

1. [DSP 电源的典型设计](#)
2. [高频脉冲电源设计](#)
3. [电源的综合保护设计](#)
4. [任意波形电源的设计](#)
5. [高速 PCB 信号完整性分析及应用](#)
6. [DM642 高速图像采集系统的电磁干扰设计](#)
7. [使用 COMExpress Nano 工控板实现 IP 调度设备](#)
8. [基于 COM Express 架构的数据记录仪的设计与实现](#)
9. [基于 COM Express 的信号系统逻辑运算单元设计](#)
10. [基于 COM Express 的回波预处理模块设计](#)
11. [基于 X86 平台的简单多任务内核的分析与实现](#)
12. [基于 UEFI Shell 的 PreOS Application 的开发与研究](#)

