

doi: 10.3969/j.issn.1006-1576.2011.04.029

# PPC4XX 平台下 VxWorks 嵌入式 USB 驱动程序开发

梅勇, 官琴

(中国兵器工业第 58 研究所 军用电子产品事业部, 四川 绵阳 621000)

**摘要:** 基于 VxWorks 操作系统的高性能、高可靠性、高实时性等特点, 开发 PPC4XX 平台下 Vxworks 嵌入式 USB 驱动程序。在详细介绍 VxWorks USB 的基础上, 给出开发流程的关键步骤。结果表明, 该方法便于操作, 可为相关人员提供参考。

**关键词:** VxWorks; USB; USB 接口

**中图分类号:** TP311.11 **文献标志码:** A

## PPC4XX Platform Embedded VxWorks USB Driver Development

Mei Yong, Guan Qin

(Dept. of Military Electronic, No. 58 Research Institute of China Ordnance Industries, Mianyang 621000, China)

**Abstract:** Based on VxWorks operating systems high performance, high reliability, and real-time characteristics, develop ppc4xx platform VxWorks embedded USB driver. On the basis of introducing VxWorks USB, put forward key steps in the development processes. The results show that the method is easy to operate, and provides a reference for the related personnel.

**Keywords:** VxWorks; universal serial bus; USB interface

### 0 引言

VxWorks 操作系统是美国 windRiver (风河) 公司推出的一种嵌入式实时操作系统, 以其高性能、高可靠性、高实时性等特点, 成为实时操作系统中最具特色的系统之一。通用串行总线 (universal serial bus, USB) 是一种计算机外围串行通信接口标准, 它克服了传统计算机串、并口的缺陷, 具有热插拔、即插即用、数据传输可靠、扩展方便、低成本等优势, 已成为当前计算机系统必备的接口之一, 也被广泛应用于嵌入式操作系统中。

笔者开发 PPC4XX 平台下 VxWorks 嵌入式 USB 驱动程序。采用的软件平台是 VxWorks 操作系统, 编译软件为 Tornado Version2.2, 硬件使用 PowerPC 构架, CPU 芯片型号为 PPC460ex (简称 PPC460 主板)。PPC460 主板的主频为 1 000 M, 其功能接口为 2 个 KM 网口, 1 个 USB 接口, 2 个串口和 1 个 VGA (video graphics array) 接口。

### 1 VxWorks USB 主机驱动栈结构

VxWorks 下 USB 主机驱动栈框架结构如图 1。

图 1 中, 在栈的最底层是 USB HC, 是主系统中控制每一个 USB 的硬件部分。目前, 市场上主要有 2 大类 USB 主控制器: 一种是支持由 intel 公司

最先提出的通用主控制器接口 (universal host controller interface, UHCI); 另一种是支持由微软、康柏和国家半导体公司联合设计提出的开放主控制器接口 (open host controller interface, OHCI)。笔者采用的是开放主控制器接口, 对这一类型的主控制器, 有一个与硬件独立的 USB 主控制器驱动 (host controller driver, HCD), 主要代码段在 usbHcdOhciLib.c 中。USB 主驱动 (USB host driver, USBD) 和 HCD 之间的接口允许一个或超过一个的底层控制器。

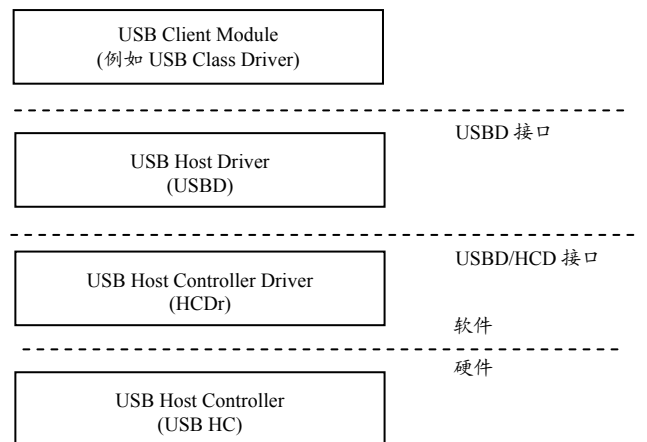


图 1 USB 主机驱动栈框架结构

U 盘是基于 USB 总线, 符合 USB 大容量存储

收稿日期: 2010-12-12; 修回日期: 2011-01-14

作者简介: 梅勇 (1979—), 男, 湖北人, 本科, 从事嵌入式系统开发研究。

类协议的移动存储设备。U 盘的数据传输是按 USB 大容量存储设备类规范的子规范 CBI (control/bulk/interrupt) 传输协议或 bulk-only 传输协议进行的。批量传输用于大容量的数据传输, 没有固定的传输速率, 不占用固定带宽, 因此, 笔者使用的大容量存储设备 U 盘采用 bulk-only 传输协议, 在 VxWorks 下提供 usbBulkDevLib.c, 实现 Bulk-only 类设备驱动。

## 2 程序流程

在 VxWorks 操作系统中, U 盘是当作块设备来使用的。VxWorks 系统中 USB 块设备的结构如图 2。

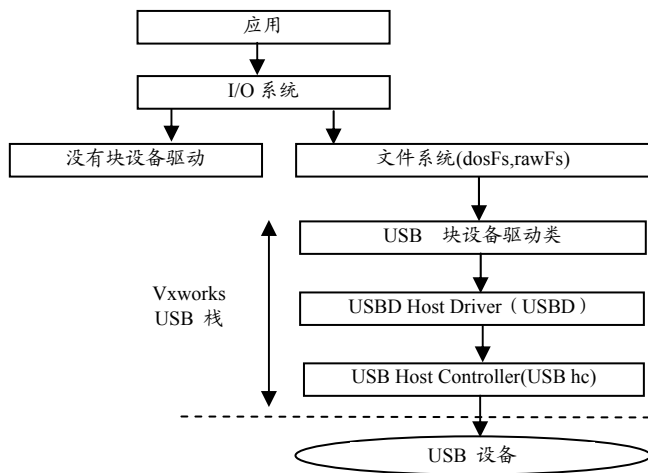


图 2 Vxworks 系统 USB 块设备结构

### 2.1 驱动基本设置及修改

由于 VxWorks 下 USB 驱动是基于 PCI 总线的, 会使用到 PCI 总线驱动库, 而在 PPC460ex 处理器上, USB 控制器是基于本地总线的, 为了借用 VxWorks 下的 USB 驱动程序, 需要修改 PPC460ex bsp 包中 usbPciStub.c 文件和硬件的接口部分。本驱动是 ohci 的接口, 有 2 个函数需要修改:

```
1)  VOID usbPciConfigHeaderGet
(
  UINT8 busNo,      /* Bus number of device
*/
  UINT8 deviceNo, /* Device number of device */
  UINT8 funcNo,   /* Function number of device
*/
  pPCI_CFG_HEADER pCfgHdr /* Buffer
provided by caller */
)
```

这个函数的主要功能获取 pci 结构设备的头信息。如果 USB 是 pci 接口的则直接使用这个函数。

如果不是基于 pci 接口的可以直接给 pCfgHdr 赋值。最主要的就是 USB 设备的基地址和中断号一定要赋值正确。否则设备无法操作。应用中修改如下:

```
VOID usbPciConfigHeaderGet
(
  UINT8 busNo,      /* Bus number of device
*/
  UINT8 deviceNo, /* Device number of
device */
  UINT8 funcNo,   /* Function number of
device */
  pPCI_CFG_HEADER pCfgHdr
  /* Buffer provided by caller */
)
{
  /* Do nothing if CfgHdr is NULL */
  int i;
  if(busNo==0&& deviceNo==1&& funcNo==0)
  {
    if(pCfgHdr == NULL)
      return;
    /* Initialize the buffer to zeros. */
    memset(pCfgHdr, 0, sizeof(*pCfgHdr));
    /* Read and store each field in the PCI
configuration header. */
    pCfgHdr->vendorId   = 0x0011;
    pCfgHdr->deviceId   = 0x0022;
    pCfgHdr->command    =
0xff;//usbPciWordGet (busNo, deviceNo, funcNo,
PCI_CFG_COMMAND);
    pCfgHdr->status     = 0xff;
    pCfgHdr->revisionId = 0x00;
    pCfgHdr->cacheLineSize = 0x00;
    pCfgHdr->latencyTimer = 0;
    pCfgHdr->headerType = 0x00;
    pCfgHdr->bist       = 0x00;
    pCfgHdr->pgmIf     = devPgmIf;
    pCfgHdr->subClass  = devSubClass;
    pCfgHdr->pciClass  = devClass;
    for (i = 1; i < PCI_CFG_NUM_BASE_REG;
i++)
    {
      pCfgHdr->baseReg [i] = 0;
    }
    pCfgHdr->baseReg [0] = 0xe20d0000;
    pCfgHdr->romBase   = 0x00;
    pCfgHdr->intLine   = 1;
    pCfgHdr->intPin    = 30;
  }
}
```

```

    pCfgHdr->minGrant = 0x00;
    pCfgHdr->maxLatency = 0x00;
}
else
    return;
}
2) STATUS usbPciIntConnect
(
    VOIDFUNCPTR func, /* new interrupt handler
*/
    VOID * param, /* parameter for int handler
*/
    UINT16 intNo /* interrupt vector number
*/
)

```

该函数的主要功能是将中断号连接到中断向量表。由于它是通过 PCI 总线库获取对应的中断号,因此其中的中断号也需要按照硬件手册的说明直接赋值而不是通过 PCI 库直接获取,否则中断无法使用。

## 2.2 VxWorks 组件的选择

选择如下组件:

- 1) hardware->buses->USB Hosts->OHCI
- 2) hardware->buses->USBHosts->USBHost Init->OHCI Init 和 USB Host Stack Init
- 3) hardware->peripherals->USBDevices->Mass Storage-Bulk
- 4) hardware->peripherals->USBDevices->USB DeviceInit->Bulk Mass Storage Init
- 5) operation system componenets->IO system components(libc)->File System and Disk Utilities
- 6) operation system components->IO system components->dosFs FileSystem Components

其中,组件 1)、2)主要是 HCD 和 usbd 的初始化。HCD 具有唯一的执行入口点,对 HCD 发出的所有请求都是通过把数据结构主机请求块(host request blocks, hrb)传递给 HCD 执行入口点,使相关请求得到处理。HCD 要能对 USB 根 hub 进行枚举,即能解释到达根 hub 的传输请求并对它做出回应。HCD 负责识别主机发送的各种请求并对他们做出回应。Usbd 主要功能是完成 USB 主机栈对 ohci 型 usb 主控制器的挂载。Ohci 型主控制器被挂载时,HCD 创建的任务“tOhciInt”监视主机控制器事件的发生。这个任务处于休眠状态,不占用 CPU 时间。

当 U 盘插拔时触发中断信号报告给“tOhciInt”,HCD 即可得知设备的状态。

3)、4)、5)、6)是大容量储存设备类驱动的组件。大容量储存设备是一种典型的块设备,其驱动程序直接与文件系统交互,用户应用程序通过 I/O 系统标准接口函数访问文件系统。U 盘设备的创建是调用 usbBulkBlkDevCreate() 完成,描述了 usb\_bulk\_dev 设备的一个逻辑分区。从 Vxworks5.5 开始,文件系统升级到 dosfs2.0 版本,引入了 CBD (cached block I/O) 的概念,文件系统创建在 CBD 上。cbioLib 封装块设备 blk\_dev 得到基本的 CBD 设备,它提供的缓存机制很简单,为了在基本的 CBD 设备的基础上实现更高的缓存,vxworks 提供了 dcacheLib,可调用 dcacheDevCreate()在块设备 blk\_dev 基础上创建缓存 CBD 设备,在利用 dpartDevCreate() 对 CBD 设备进行分区。dpartPartGet()获得某个分区后,将此分区的 CBD 设备挂装到 DOS 文件系统(dosFsDevCreate()实现),I/O 系统便可通过文件系统来访问 U 盘设备了。

大容量存储设备可以在任意时刻插入系统或从系统拔出,USB 类设备驱动通过回调机制支持 USB 设备的热插拔特性。usbBulkDynamicAttachRegister()将用户注册到大容量设备类驱动,当 U 盘插入系统或从系统拔出时,HCD 能知道状态的变化,然后 USB 调用用户提供的回调函数,将 U 盘设备的 USB\_NODE\_ID 和标识符(USB\_BULK\_ATTACH 或 USB\_BULK\_REMOVE,显示设备连接或卸下)传递给回调函数。若要取消大容量设备类驱动对用户的注册,可通过 usbBulkDynamicAttachUnregister()实现。

## 3 结束语

通过上述关键步骤,再编译下载到主板上,就可以对插入的 U 盘进行读写操作、格式化操作等。该主板 USB 接口可以操作、存储上百兆的大文件,对 U 盘的识别没有限制,并能动态插拔 U 盘。结果表明,该方法操作方便,可为相关人员提供参考。

## 参考文献:

- [1] USB Developer's Kit Programmer's Guide1.1.2.
- [2] 李霞,刘宏立,童调生. VxWorks 下嵌入式 USB 主机驱动的实现[J]. 航天控制, 2007(12): 20-45.
- [3] 康涛. 在 CVI 中直接编写 USB 驱动通信软件的方法[J]. 兵工自动化, 2009, 28(12): 94-96.