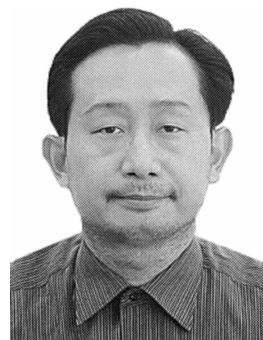


研
究
开
发

机车网络中 USB 设备的开发与应用

易伟民, 肖家博

(株洲南车时代电气股份有限公司 技术中心, 湖南 株洲 412001)



摘 要: 分析了基于 coldfire CPU 与 VxWorks 实时操作系统的 USB 驱动程序的体系架构与设计实现, 介绍了 USB 设备在机车通信网络中的多种应用。实际应用表明, 该设备在机车网络应用中稳定可靠、方便易用, 该设计可推广应用于多个领域。

关键词: 车辆控制模块; 列车通信网络; VxWorks; USB

中图分类号: U26; TP393 **文献标识码:** A **文章编号:** 1000-128X(2011)02-0012-03

作者简介: 易伟民 (1973-), 男, 硕士, 工程师, 从事机车网络基础开发与工业网络集成应用。

Development and Application of USB Device in Locomotive TCN

YI Wei-min, XIAO Jia-bo

(Technology Center, Zhuzhou CSR Times Electric Co., Ltd., Zhuzhou, Hunan 412001, China)

Abstract: This paper analyzed the structure and the design of USB device, which was based on a coldfire CPU and the VxWorks platform. This paper introduced the multi-application of USB device in TCN. In practice, this USB device has good stability and reliability and is easy to use. The design can be applied to many types of equipment in locomotive.

Key words: vehicle control module(VCM); TCN; VxWorks; USB

0 引言

随着计算机技术的高速发展, 外围设备日益增多, 系统中数据量在不断增大, 数据传输速度越来越高, USB 接口的应用随之越来越广泛。USB 使用方便, 成本低, 支持热插拔和即插即用, 接口可对外提供电源, 数据传输速度最高可达到 480 Mbps, 最多支持 127 个设备, 这一系列的优点与特点使它在铁路机车车辆领域也有广阔的应用空间。列车通信网络 (TCN) 中一些设备可用 USB 接口实现大容量数据交互, 例如程序更新、软件版权授权、故障环境信息下载等等, 其基础平台均可统一化。本文以车辆控制模块 (VCM) 为例, 介绍一种 USB 设备的设计与实现方法。VCM 作为基于 IEC61375 的分布式列车控制系统 DTECS 中的核心功能单元, 主要用于车辆的过程控制、故障诊断和存储、车辆级通信管理等功能。为了方便 VCM 中大容量故障数据的分析、故障数据下载、程序的上传下载以及一些控制操

作等, 应用 USB 是一个很好的解决方案。

1 硬件设计

在嵌入式系统中, 常用的 USB 硬件设计有 2 种: 一种是微处理器 CPU 通过外部总线扩展 USB 控制器件; 另一种是选用内嵌 USB 控制器的 CPU。二者都需要配合 USB 通信的物理层接口电路。飞思卡尔公司 coldfire 产品系列提供了多种内置 USB 控制器的 CPU, 包括低速/全速和高速 USB 解决方案。这些器件易于集成、可靠性高、软件和硬件支持包众多, 便于开发人员优化设计流程, 加快产品开发进程。本设计以内置 USB 控制器的 coldfire CPU 为例, 使用其他系列 CPU 时只需做简单的移植即可。

USB 控制器信号连接的通信接口物理层电路如图 1 所示, 可提供 500 W 浪涌防护能力, 可对外提供 500 mA 供电电流。

2 VxWorks 下 USB 驱动协议栈结构

USB 支持几种数据传输方式: 控制、块、中断、同

收稿日期: 2010-05-25; 收修改稿日期: 2010-11-23

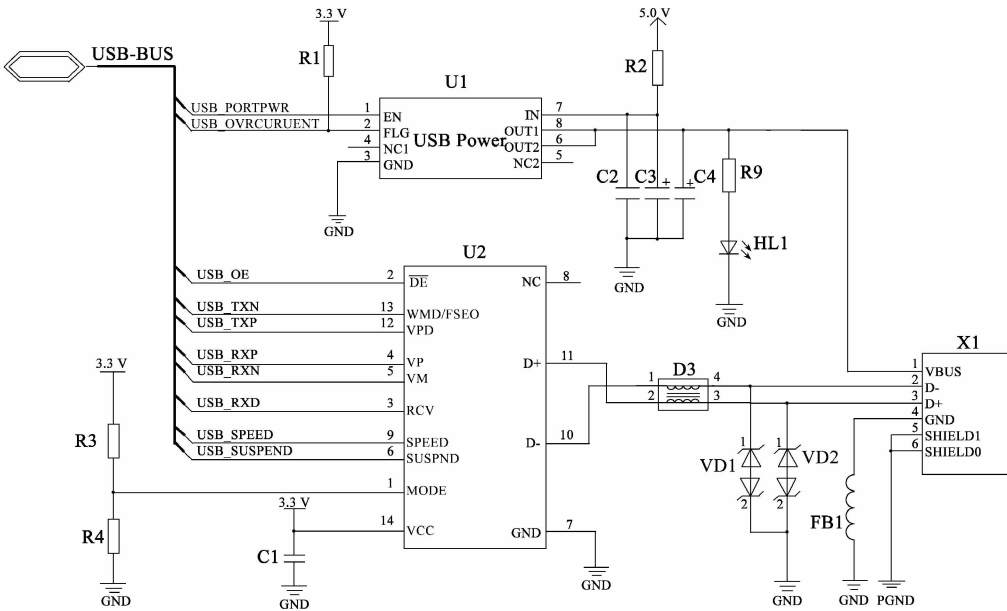


图 1 USB 接口物理层

更高层提供可与 USB 设备通信的路径，同时还负责 USB 电源处理、USB 带宽管理、USB 热插拔。USB 支持 USB 2.0 规范，并且向下兼容 USB 1.0。VxWorks 下 USB 完成的工作主要包括 USB 初始化、USB 注册接口、设备管理、USB 请求接口。USB 初始化是首先调用 USB 的入口函数 `usbInit()`，初始化 USB 用到的数据结构，然后调用函数 `usbHubInit` 初始化集线器驱动的数据结构，最后使用函数 `usbHCDInit()` 挂接相应的主机驱动器。注册接口主要是因为 USB 设备类驱动与 USB 共享有 `USBHST_DEVICE_DRIVER` 数据结构。为了实现与 USB 设备通信，必须在 USB 中通过函数 `usbHstDriverRegister()` 注册这个数据结构。USB 为每一个设备类维护着一个设备类结构。设备管理主要是指当 USB 设备插入或拔出 USB 系统时，集线器驱动程序将 USB 设备插入或拔出的信息通知 USB，USB 根据得到的设备信息（例如设备类号，子类号，协议号等）查找设备驱动程序列表，查找到该设备的驱动程序后调用 USB 设备的相应函数进行设备的相应操作。当一个设备类不在系统中使用时，通过撤销函数来撤销注册。该操作的实质是删除相应设备类的设备类结构。USB 请求接口主要是规范中定义的一系列标准请求接口，这些接口必须被所有的 USB 设备支持。

步。USB 主协议栈驱动在 VxWorks 实时操作系统中从下至上协议栈分为：USB 主控制器、USB 主控制器驱动、USB 驱动、转换层、USB 类设备驱动。VxWorks 下 USB 主协议栈的结构如图 2 所示。

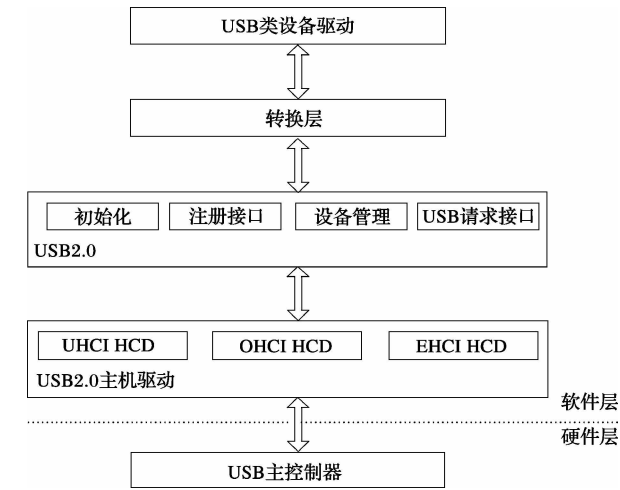


图 2 USB 主协议栈结构

① USB 主控制器位于协议栈的最底部，这是主系统中控制 USB 设备的硬件部分。在车辆控制模块 VCM 中是指 CPU 自带的 USB 控制器。

② USB 控制器驱动主要指操作系统中负责 USB 控制器读写操作的功能代码。能够使用 USB 的硬件系统一般包括 1 个或多个 USB 主控制器 (USB host controllers)，各个制造厂家生产的 USB 主控制器基本上都符合 3 种主要的规范中的 1 种，它们分别是：EHCI、OHCI、UHCI。本设计在 VxWorks 操作系统下为符合 3 种规范的主控制器开发了与硬件独立的 USB HCD 驱动 (USB HOST CONTROLLER DRIVER)。

③ USB 与底层硬件无关，其通过 HCD 与底层控制器通信。USB 管理每一个与主机相连的 USB 设备，向

④ 转换层的功能主要是保持向下兼容 VxWorks 支持的设备驱动（键盘、鼠标等）。例如 USB 1.0 协议栈更新到 USB 2.0，已经存在的基于 USB 1.0 主协议栈的 USB 设备类驱动接口通过 USB 2.0 转换层就能用于 USB 2.0 环境中。

⑤ USB 类设备驱动主要包括集线器类设备驱动、键盘类设备驱动、鼠标类设备驱动、打印机类设备驱动、扬声器类设备驱动、大容量的存储器类驱动、通信类设备驱动等。

VxWorks 下 USB 主协议栈启动流程为：初始化 USB 主控制器 (USB)；加载 USB 主控制器驱动 (HCD)；加载 USB 设备驱动程序。

在 VxWorks 集成开发环境 WORKSBENCH 下进行配置，其步骤如下：

① 包含 USB 主协议栈 USB Host Stack 组件，通过在工程中包含 USB 主协议栈完成对 USB 的支持。系统启动时对 USB 主控制器驱动进行初始化，需要包含

USB Host StackInit 组件。

②包含主控制器(Host Controller), 根据具体设计选用相应的控制器。一般在嵌入式模块的手册中都有其模块控制器的类别, 在系统启动时对主控制器进行初始化需要包含相应控制器的初始化组件。

③包含设备驱动, 根据设计的需求选择相应的设备驱动。

通过上述配置生成的 VxWorks 操作系统中带有 USB 主协议栈, 可以完成 USB2.0 主站功能。

3 VCM 中 USB 的实现

VCM 中 USB 初始化及配置流程图如图 3 所示。

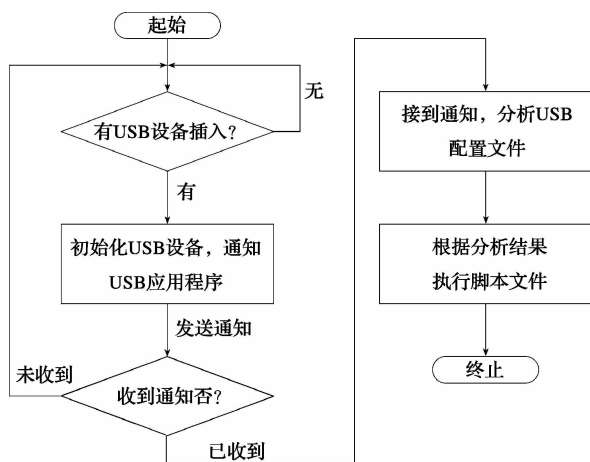


图 3 USB 初始化及配置流程

在 USB 主协议栈初始化完成后, 一旦有 USB 设备接入系统, 就会进行 USB 设备的初始化操作, 同时发送一个信号量通知 USB 应用处理程序。当 USB 应用处理程序收到该信号量后, 立即分析 USB 中的操作配置文件, 然后执行配置文件脚本, 同时 VCM 面板上灯不停闪烁, 并把相关执行脚本日志写进 USB, 以便分析。脚本执行完后, 灯恢复正常状态, USB 操作完毕。

VCM 中 USB 的配置文件解析与执行过程是, 车辆控制模块 VCM 中 USB 的配置文件存放在 USB 盘的 teg 目录下, 当 USB 盘识别后, 就会读取 USB 盘中的 USB 配置文件 usb_confm.dat, 解析用户要执行什么操作。下面是 VCM 使用 USB 来更新底层程序 isa.out、VxWorks, 并更新 VxWorks BSP 的一个配置文件:

```
rm /tffs/isa.out
rm /tffs/VxWorks
copy /usb0/teg/isa.out /tffs/isa.out
copy /usb0/teg/VxWorks /tffs/VxWorks
refreshFlash /usb0/bootrom.bin
```

脚本首先删除/tffs/目录下的文件isa.out、VxWorks, 然后执行拷贝命令把/usb0/teg/目录下的文件isa.out、VxWorks拷贝到文件系统/tffs/目录下, 最后执行bootrom的更新命令。更新需要的文件在/usb0/目录下。

USB 操作完以后, 会把脚本相关的执行结果写到

文件usb_log.dat。usb_log.dat 文件用来记载执行usb_confm.dat中脚本的结果。如下所示, 首先记录日志开始标志, 把脚本的执行时间记录下来, 然后是相关的脚本执行操作。

```
*****UPDATE LOG START*****
```

```
Now we begin to update program....
```

```
START DATE:09-07-06 START TIME:13:47:52
```

```
Delete File /tffs/ isa.out Successful
```

```
Delete File /tffs/ VxWorks Successful
```

```
Copy File /usb0/teg/isa.out TO /tffs/isa.out Successful
```

```
Copy File /usb0/teg/VxWorks TO /tffs/VxWorks Successful
```

```
refreshFlash bsp from /usb0/bootrom.bin Successful
```

```
Now we stop to update program!!
```

```
*****UPDATE LOG START*****
```

当实现特定功能时, 首先进行相关协议栈的初始化, 创建一个信号量 USB_FIND_SEM=semBCreate(SEM_Q_FIFO, SEM_EMPTY), 并执行等待 USB 设备插入的函数 semTake(USB_FIND_SEM, WAIT_FOREVER)。如果发现了 USB 设备, 首先解析 USB 设备中配置文件, 并执行相关操作 usbop_syssoft()。根据配置文件, 通过发送不同的信号量, 程序可跳转进入响应的功能模块, 例如 VCM 故障记录数据下载 fault_down()、程序更新 program_update()、便携式调试接口 ptd_debug()、软件版本授权管理 ver_manager() 等。当执行相关功能操作的时候, VCM 面板上的 ST 灯按相应规律闪烁。这样, 通过一个共同的底层程序接口, 可将该 USB 设备应用于不同需求, 使得该方案在列车通信网络中可以应用于多个功能模块。

4 结语

USB 接口在列车通信网 TCN 中, 可实现故障下载、程序更新、调试接口、软件授权管理等众多功能, 在技术开发时选好一款合适的 CPU 作为整个网络的通用 CPU, 那么该 USB 接口可适用于网络中众多的功能模块, 其他功能接口也可按这种思路进行设计。本设计已经应用于多个项目, 为调试人员、服务人员提供了便利的人机接口。实际应用表明, 该设备稳定可靠、使用方便, 不仅可推广应用于铁路车辆领域, 还可应用于地面信息系统以及其他领域。

参考文献:

- [1] Jan Axelsson, 陈逸. USB 大全 [M]. 北京: 中国电力出版社, 2003.
- [2] IEC61375-1:1999, 铁道电气设备 列车总线 第一部分: 列车通讯网络 [S].
- [3] 曾嵘, 路向阳. 首列国产交流传动地铁列车控制及网络系统 [J]. 机车电传动, 2004(3): 35-38.
- [4] 曾嵘, 杨卫峰, 刘军. 列车分布式网络通信与控制系统 [J]. 机车电传动, 2009(3): 17-19.