第33卷 第21期 计 算 机 工 程 2007年11月
Vol.33 No.21 Computer Engineering November 2007
·工程应用技术与实现· 文章编号：1000—3428(2007)21—0240—03 文献标识码：A 中图分类号：TP36

一种嵌入式USB主机功能模块设计与实现

朱晓锦，庞海珑，王 健，邵 勇
(上海大学机电工程与自动化学院上海市电站自动化技术重点实验室，上海 200072)

摘 要：基于LPC2292(ARM7)为CPU、Nucleus PLUS为实时操作系统的测控系统平台，分析了以ISP1161A1 USB主机控制芯片构建USB主机模块的设计与实现方法，包括USB主机功能模块设计方案、嵌入式USB主机的硬件设计思路、嵌入式USB主机驱动程序的设计方法以及设计方案的实现方式和过程。针对主机驱动程序的主要相关功能、面向通用USB设备驱动程序接口和主机控制器驱动程序的下层数据管理等关键问题，给出了解决方案和实现方法，提供了关键数据结构和简明注释。

关键词：嵌入式USB主机；USB主机控制器；驱动程序

Design and Realization of Embedded USB Host Function Module

ZHU Xiao-jin, PANG Hai-long, WANG Jian, SHAO Yong
(Shanghai Key Laboratory of Power Station Automation Technology, School of Electromechanical Engineering & Automation, Shanghai University, Shanghai 200072)

【Abstract】 Based on a test and control platform with CPU LPC2292(ARM7)and RTOS with Nucleus PLUS, this paper analyses and expatiates design and realization method of a USB host module based on USB host controlling chip ISP1161A1, including the design scheme of USB host function module, the frames of USB host hardware circuit, the design method of embedded USB host driver procedure, and the realization mode and operation process of this scheme. According to the primary and correlated function of host driver procedure, aiming at the key problems of general USB device driver procedure interface and lower layer data management of host controller driver procedure, the settlement scheme key data structure and concise remarks are given.

【Key words】 embedded USB host; USB host controller; deriver

1 概述

所谓嵌入式USB主机系统，就是在嵌入式系统中实现USB主机功能，使嵌入式系统具有与USB设备进行通信和数据交互的功能。嵌入式USB主机技术作为USB总线的推广，将USB技术拓展到与PC机无直接联系的嵌入式领域，该项技术在国内外正处于积极研究和发展阶段，目前已有部分半导体公司推出了USB主机接口芯片，如Philips，Cypress、Atmel等；同时市场上也逐步出现了具有部分功能的嵌入式USB主机产品，尤其是某些消费类电子产品，如数码相机直接连接到打印机等。但目前嵌入式USB主机技术和产品还存在许多不足，突出不足之处就在于功能单一，仅能支持一种具体的USB设备，不能为其他的USB设备提供主机服务，从而无法体现出USB技术通用性的特点和优势，这也是嵌入式USB主机系统无法在工业控制领域中获得充分应用的主要原因[1~2]。

本文基于上海市科委“基于混合传输方式的测控系统研究”科研项目，针对USB嵌入式系统在工业控制领域应用的现状，提出了在LPC2292(ARM7)为CPU、Nucleus PLUS为实时操作系统的测控系统平台上，以ISP1161A1 USB主机控制芯片为核心构建USB主机模块的设计思想与实现方法。该模块实现了USB主机控制器功能，尤其是基于软件策略为测控系统平台上加载不同的USB设备驱动程序提供了通用接口，最终在该测控系统中实现了多种USB设备的数据传输与通信，同时针对主机驱动程序的主要相关功能，面向通用USB设备驱动程序接口和主机控制器驱动程序的下层数据管理等关键问题，给出解决方案和实现方法。实际调试和运行表明，只要加载某一类USB设备驱动程序，就能实现该测控平台同时与相应的USB设备进行通信。

2 总体设计方案

总体方案构建中，系统测控平台的CPU选用的是基于ARM7内核的中央处理器LPC2292，实时操作系统则基于Nucleus Plus系统处理内核。由于系统的混合传输模式除了外接USB模块外，还有图形用户系统模块、ZigBee无线发射系统模块和TCP/IP网络接口模块，因此该类功能模块均以扩展板卡的形式进行设计。

方案中嵌入式USB主机模块的设计主要分为硬件设计和软件设计两个部分。由于测控平台提供了CPU的地址数据总线接口，因此硬件设计部分主要考虑USB主机控制器芯片的外围电路设计；而嵌入式USB主机软件设计是关键技术和主要难点，原因在于针对嵌入式USB主机特定功能，关键是要求实现嵌入式USB主机系统和测控平台必备的通用属性，即能够为USB设备驱动程序提供通用接口，从而使测控系统平

基金项目：上海市科委科技攻关计划基金资助项目(041111038)；上海市教委“曙光计划”基金资助项目(04SG41)；上海市重点学科建设资助基金项目(T0103)
作者简介：朱晓锦(1965－)，男，博士、副教授，主研方向：测控技术与信号处理；庞海珑，硕士研究生；王 健，教授；邵 勇，讲师
收稿日期：2006-11-13 **E-mail：**mgzhuxj@staff.shu.edu.cn

台具备同时针对不同USB设备提供主机服务的功能[3]。

PC 机上的 USB 主机软件系统结构通常被分为 3 层，即通用串行总线驱动程序(USBD)、主机控制器驱动程序(HCD)和客户端软件。图 1 所示为这 3 层软件的结构和主要功能。本文所述的主机端软件结构也借鉴了这种软件体系。

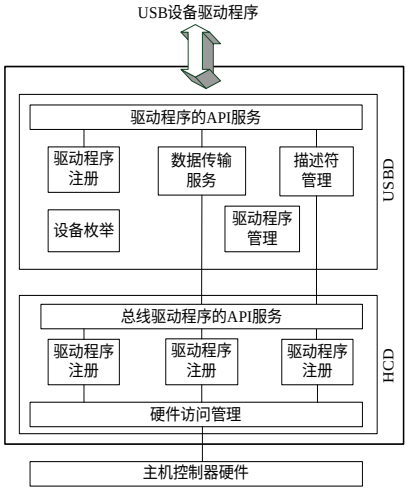

图 1　USB 主机软件系统结构

分析USB主机的软件结构，可知USBD要为上层的客户端驱动程序提供USB设备的抽象，并为USB设备驱动程序和所驱动的设备之间提供数据传输的抽象，这些抽象的传输要通过USBD为USB设备驱动程序提供通用接口进行传输；而HCD则为USBD提供USB总线抽象以及数据在总线上传输的抽象，这些抽象也要通过USBD为HCD提供接口来传输。上述抽象与接口的实现，就可以使该嵌入式USB主机系统平台具有通用属性，从而为多种USB设备提供USB主机的服务支持和功能[4]。

3　USB 主机模块硬件设计

USB主机控制核心芯片选用的是Philips公司ISP1161A1，该芯片涵盖通用串行总线(USB)、主机控制器(HC)和设备控制器(DC)功能，其主机控制器符合通用串行总线 2.0 规范，支持全速(12Mb/s)和低速(1.5Mb/s)的数据传输，其设备控制器同样符合通用串行总线 2.0 规范，并支持全速(12Mb/s)的数据传输[1]。另外，HC和DC共用一个微处理器总线接口，具有相同的数据总线，但I/O地址不同，同时也有各自的中断请求输出管脚。ISP1161A1 为USB HC 提供两个下行端口，为USB DC 提供一个上行端口，其中HC的下行端口可与任意一个符合USB2.0 规范并含有USB上行端口的USB 器件和USB集线器相连。类似地，DC的上行端口可与任意符合USB2.0 规范并含有USB下行端口的USB主机和集线器相连。

由于测控系统平台为外扩模块所提供的接口为 P2 口，考虑到 ISP1161A1 中定义了一个通用 PIO 接口，以增加速度及易用性，并可与大多数微控器直接接口，因此本文规定 ISP1161A1 采用 PIO 模式。对于 ARM7 内核中央处理器 LPC2292，USB 主机控制芯片 ISP1161A1 类似于一个具有 16 位数据总线的存储器设备，且只使用 A1 和 A0 两条地址线访问内部控制寄存器及 FIFO 缓冲区 RAM，因此，ISP1161A1 仅占 LPC2292 的 4 个存储单元。在可编程的 I/O(PIO)操作模式下，外部微处理器可读/写 ISP1161A1 的内部控制寄存器及 FIFO 缓冲区 RAM。图 2 所示为 ISP1161A1 与微处理器之间的硬件接口连接原理，这里应该注意的是，ISP1161A1 的外部复位端口占一个 GPIO 口资源。

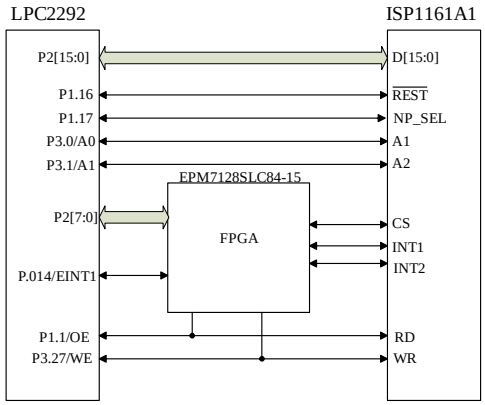

图 2　ISP1161A1 与处理器硬件原理

4　USB 主机模块软件设计

4.1　主机驱动程序 USBD 的实现

从协议本身和实际应用的角度来说，在本嵌入式 USB 主机系统中，USBD 作为主机驱动程序的中间层要提供 4 类功能：USB 总线管理，USB 设备的配置和管理，协议控制命令集和数据传输的管理。对于以上几个功能的管理，本设计分别定义了抽象数据结构：usb_bus, usb_device 和 UTB。通过对这 3 个数据结构的组织和管理实现了相应功能。对于协议控制命令集，本设计是在对 USB2.0 协议的分析了解的基础上，根据协议中所定义的命令，给出相应的命令函数来实现的。

由于篇幅的限制，在本文中仅列数据传输管理抽象数据结构的实现。为了实现数据传输，在 USBD 建立了一个数据结构 UTB。UTB 中具有 USB 数据传输的所有信息，包括传输类型、传输方向、数据缓存区、数据传输的设备、端点、返回信息及指向传输完成的处理函数指针等。UTB 的数据结构定义如下：

```
struct UTB{
struct list_head UTB_list;//保存 UTB 链头
struct UTB *next;//用于连接下一个 UTB
struct usb_device *dev;
//指向 UTB 所要访问 USB 设备的相关信息
uint32 status;//保存 UTB 的状态信息
uint32 transfer_flags;//保存 UTB 的数据传输标志
uint32 *transfer_buffer;//指向 UTB 的数据传输缓冲区
uint32 transfer_buffer_length;//保存 UTB 数据传输缓冲区大小
uint32 usb_data_length;//保存 UTB 的真实数据长度
uint32 bandwidth;//保存 UTB 所需的带宽
uint32 start_frame;//保存起始帧
uint32 number_of_packets;//保存 UTB 相关联的数据包数据
uint32 error_count;//用于 UTB 数据传输错误计数
uint32 timeout;}//用于控制 UTB 数据传输的输入参数
```

4.2　USBD 与设备驱动程序接口的实现

当 ISP1161A1 集线器检测到有设备接入时，发出相应的中断信号；操作系统则调用相应的中断程序，通过 USBD 对设备进行交互，这里交互主要包括获取设备的表述符和对设备进行配置。配置完成后，系统通过获取的描述符和设备号，在 USB 设备驱动程序中搜索相应驱动程序，找到后调用 usb_dr_install()函数装载该驱动程序，并在装载完成后将设备管理权交给驱动程序。

驱动程序与 USBD 的接口分为命令接口和功能接口。命

令接口包括：建立 UTB 函数 usb_alloc_utb()，释放 UTB 函数 usb_free_utb()，向 USBD 提交 UTB 的传输函数 usb_submit_utb()，以及取消正在传输的 UTB 函数 usb_unlink_utb()；功能接口主要实现 USB2.0 协议规范中所定义的 4 种传输：控制传输 usb_control_msg()，批量传输 usb_bulk_msg()，中断传输 usb_int_msg()，以及实时传输 usb_rt_msg()，这 4 个传输函数实现了 USB 中 4 种传输方式的数据组织。在数据组织完成后，再调用命令接口的 usb_submit_utb()，将组织好的数据提交给 USBD，然后 USBD 通过 HCD 并由 ISP1161A1 对 USB 设备进行通信，从而完成整个数据交互。

4.3 USBD 与 HCD 接口的实现

在 HCD 层，提供一个 USBD_HCD_interface 结构，USBD 将通过该结构中的接口函数来访问相应的主机控制器。USBD_HCD_interface 结构主要由资源管理接口和 USB 数据传输接口两部分组成，定义如下：

```
struct USBD_HCD_interface{
uint32 (*allocate)(struct usb_device *);//HCD 资源分配函数
uint32 (*deallocate)(struct usb_device*);//HCD 资源释放函数
uint32 (*get_frame_number)(struct usb_device *usb_dev);
//读取主机控制器帧数目
uint32 (*submit_UTB)(struct UTB *pUTB);
//提交一个 UTB 给 HCD
uint32 (*unlink)(struct UTB *pUTB); }
//从 HCD 上取消一个已提交的 UTB
```

4.4 主机控制器驱动程序 HCD 的实现

HCD的主要功能是完成对ISP1161A1 的驱动，并与ISP1161A1 合作完成USB各种事务处理。它根据一定的规则调度所有将被广播发送到USB上的事务处理，将调度好的USB事务发送到ISP1161A1 的缓存中；ISP1161A1 处理完成USB事务时，将返回的结果仍存到缓存中；CD则将这些结果交给USBD层，此外HCD还完成对ISP1161A1 的配置和驱动等操作[5]。

HCD 的主要技术难点是事务的调度问题，本文主要参考 OHCI(open host controller interface，开放主机控制接口)规范的事务调度方法进行设计。

首先是要定义一个 OHCI 的数据结构，通过这个数据结构来对各种传输类型的数据进行管理。该结构体具体如下：

```
typedef struct ohci {
struct root_hub rh; //根集线器的结构体
uint8 active; //ISP1161A1 工作状态
struct ctrl_ptd ctrllist[CTRL_NUMBER]; //控制传输数据包链表
struct bulk_ptd bulklist[BULK_NUMBER];
//批量传输数据包链表
struct done_ptd * done_list_head; //完成链表
struct iso_ptd *isolist; //周期性数据链表
struct usb_bus * bus; //USB 总线
uint32 frame_number; //帧数
}ohci_t ;
```

结构中定义了 4 个链表：控制传输数据链表，批量传输数据链表，完成数据链表和周期性数据链表，其中除了完成数据链表是一维链表外，其他的链表都是二维链表(如图 3 所示)，主要由数据结构 PTD(Philips transfer descriptor)和 ED(endpoint descriptor)组成。每个链由一个可变的全局变量指针指定，每个 ED 描述 USB 设备一个端点(endpoint)的所有数据传输，所有的 ED 被连接在一起，而 PTD 表述才是最终要在 USB 总线上传输的数据包，同时属于同一个 USB 设备端点的 PTD 被链接在一起，并挂在相应的 ED 上。

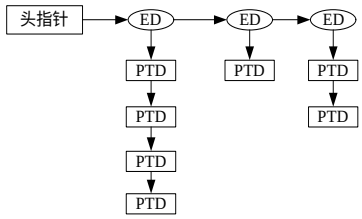

图 3 链表结构

HCD 就是通过对这些链表操作来完成对各种传输类型的数据进行调度。当 USBD 传下来一个数据时，HCD 将该数据转为 PTD 数据格式后，按照其传输类型挂到相应的链表上。当 ISP1161A1 的帧中断到来后，HCD 首先把 ISP1161A1 上 ITL 和 ATL 缓存区中的上次传输完成返回的结果取出，并放到完成数据链表上，然后按照带宽的管理规则，将周期性链表上的中断和实时数据摘下，放到 ISP1161A1 的 ITL 缓存区中，再把控制传输与批量传输链表上的 PTD 包摘下，放到 ATL 缓存区中，而剩下 USB 总线上的数据传输就可由 ISP1161A1 完成，此时 HCD 可以将返回的数据放入到相应的 UTB 中，由 USBD 向上层传输。

4.5 客户端驱动程序

USB 客户端驱动程序的作用相当于普通设备驱动程序，前述部分已对 USB 系统软件设计进行了分析，不仅简化了驱动程序软件通信协议的复杂度，又使它与硬件的操作隔离开来，大大简化了与设备通信所需的工作，从而使客户端驱动程序的编写变得更加简化和易于开发。因此，不再详细介绍 USB 的客户端驱动程序的设计过程，仅提及所要涉及到的注意要点和必要方面。

在 USB 设备驱动程序设计时，首先要考虑到设备类规范。USB-IF 除了定制 USB 基本规范外，还制定了一系列设备类规范，且每一个标准 USB 设备都有相应的设备类规范。如鼠标、键盘对应的是人机交互设备类(HID)，数码相机、扫描仪对应的是影像设备类，打印机对应的是打印设备类。因此，必须遵照相应的设备类规范进行设计，而且客户端驱动程序与下层通信就是按照这些设备类规范来实现的。客户端驱动程序与操作系统接口则是通过为操作系统提供 API 函数来实现的，当系统要求与设备进行通信时就要调用这些 API 函数，并与客户端驱动程序进行交互；然后客户端驱动程序再根据交互内容按照相应设备类规范进行与 USBD 的通信，并且 USBD 再调用 HCD 将通信内容发往 USB 设备，最后 HCD 与 USB 设备通信的最终结果再向上逐层提交，直至操作系统得到该结果。

5 结束语

本文所实现的嵌入式 USB 主机系统平台，在硬件上提供了两个信号传输稳定的主机端口，在软件上提供了通用的 USB 设备驱动程序接口，成功地解决了目前嵌入式 USB 主机无法在工业控制领域中进行多个 USB 通信的难题，并在嵌入式 USB 主机处理系统平台的构建方法和开发技术上，获得了良好的研究结果，掌握了关键的技术方法，进一步拓展了嵌入式 USB 主机的应用领域，为 USB 通信技术在嵌入式工业控制系统的积极应用提供了关键技术储备。

（下转第 248 页）