

文章编号:1004-5694(2001)03-0053-04

通用串行总线接口技术及其应用

夏险峰, 张伟, 张治中, 张云麟

(重庆邮电学院 通信网与信令系统重点实验室, 重庆 400065)

摘 要:通用串行总线(USB)是一种正在迅速发展的高速外设总线,它具有热插拔、智能化、易于使用等特点。简要介绍了USB协议,讨论了USB技术的原理及其工作流程和实现技术,并结合USBN9602芯片的结构及工作原理,阐述了USB接口电路的硬件设计过程和软件设计流程。

关键词:通用串行总线;宿主;外设;令牌

中图分类号:TN919.8 **文献标识码:**A

The Technology and Application of USB Interface

XIA Xian-feng, ZHANG Wei, ZHANG Zhi-zhong, ZHANG Yun lin

(Key Laboratory of Telecommunication Networks and
Signalling Systems, CUPT, Chongqing 400065, China)

Abstract: Universal Serial Bus (USB) is an emerging external bus employed in connecting peripherals with PC. The virtue of USB comes from its ability of being shared in essence. In USB system, the devices can be attached and detached without turning off the power supply. It is intelligent and expansive as well as apt to reduce the complexity of development of new peripherals. This paper presents USB protocol Revision 1.1, the principle and implementing method of this bus. And according to the construct and principle of the chip USBN9602, the process of how to design the hardware and software of the USB interface circuit is described here.

Key words: USB; host; device; token

通用串行总线(USB)是一种正在迅速发展的高速外设总线。它具有较高的数据传送速率,适应多媒体和通信需求;支持更多的外设(最多可达127个),减少接口和中断的数目,多个外设共享串行总线,减少成本,由宿主预先分配带宽,不出现总线竞争,支持即插即用,实现智能化接口,支持热插拔,易于使用等特点。鉴于USB的这些出众特性,康柏、IBM、Intel、微软、NEC、北方电子等大公司先后宣布支持USB协议,而国内联想等公司也在新机型上装备了

USB接口,而新近发布的USB协议2.0版本亦可支持速率高达480 Mbit/s的外设。

1 USB的工作原理及流程

1.1 USB的工作原理

USB的基本思想在于共享,即多个外设共享一条外设总线。实现共享的关键在于外设与主机的接口特征(电气特征和机械特征)即传输信息的标准化

• 收稿日期:2001-01-15

基金项目:重庆市重点科技资助项目(合同号:985033)

作者简介:夏险峰(1974-),男,四川万源人,硕士研究生,主要研究方向为通信网与信令系统。张云麟,教授。

与规范化,并恰当地把总线带宽分给各个管道。USB系统采用宿主集权模式,即宿主控制通信过程,外设遵从宿主控制,所有事务由宿主来启动。

1.1.1 USB逻辑设备与管道

USB系统由位于主机中的USB宿主(Host)、USB外设(Device)及连接它们的电缆组成。宿主和外设都是层次结构,如图1所示。

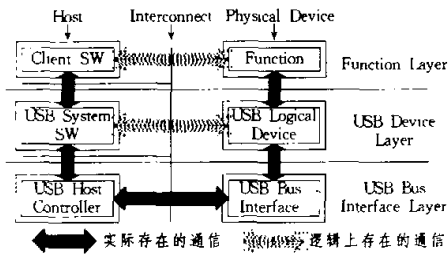


Fig. 1 USB system

宿主由客户程序(Client Software)、USB系统程序(USB System Software,由USB Driver和Host Controller Driver组成)、宿主控制器(Host Controller)组成。USB外设可分为2种,即一类称为集线器(Hub),为其他USB设备提供扩展接口并对外设有一定的控制功能;另一类为功能块(Function)。从客户应用程序来看每个功能块逻辑上由若干应用端点构成。

1.1.2 USB的包

事务是USB系统中信息传递的逻辑单位,其物理实现要依赖消息包机制。在连接宿主与外设的电缆上传递的信息流由消息包组成,包分为4类:令牌包、帧启动包、数据包、应答包,每个包均由域组成。这些域有:同步域、包标识域、地址域、帧号域、数据域、循环冗余码校验。

1.1.3 USB的传输事务

USB的传输事务有4种类型:批事务、控制建立事务、中断传输事务和同步传输事务。

1.1.4 USB的错误检测

USB的错误检测采用3种错误检测机制:比特填充破坏点、PID检测比特和CRCs。

1.1.5 USB的电气特征

在USB接口电路中,用于收发串行位流的收发器直接与串行接口引擎(SIE)相连,而位流在D+和

D-信号线上传送。当用于全速(12 Mbit/s)通信时,信号线电缆最长为5 m,并且用带屏蔽的双绞线;而用于低速通信时(1.5 Mbit/s),电缆最长3 m,用不带屏蔽的单线。发信号“1”时,D+上电压比D-上电压至少高200 mV,D+和D-上传送的是NRZI码。

1.2 USB工作流程

1.2.1 集线器、外设与系统互联

USB系统的连接方式为分层结构。根集线器与宿主相连,其他外设/集线器都与上一层的集线器相连,整个USB系统连接构成一棵树,普通外设是这棵树的叶子,集线器/复合型外设为非叶子接点(复合型外设是集线器和外设的一种综合形式)。

集线器在USB系统中具有重要作用,它使更多的外设可与宿主相连。一个外设可能处于6种状态之一:连接状态(Attached)、上电状态(Powered)、默认状态(Default)、地址状态(Address)、配置状态(Configured)及挂起状态(Suspended)。外设操作分5种:外设的动态接入与撤消、对外设赋地址、对外设进行配置、传输数据、外设的电源管理。宿主通过在与外设之间相连的默认通道上发出服务请求来控制外设。外设服务请求都属于控制类型,且分为3种类型:标准服务请求、外设类服务请求、自定义服务请求。外设服务请求的过程可描述为:建立连接包—对外设下达的命令+应答包。如外设处于可用状态并成功地收到服务请求后,就接收宿主发来的数据,并将成功地通知宿主。当一个外设连到USB上,集线器会向宿主报告,宿主将调用总线识别程序(Bus Enumeration)来识别和管理外设的状态变化。总线识别程序将重启动外设并赋给外设一个单独的地址;宿主从各个外设取得配置信息,并根据它来配置每个外设,此时外设处于可用状态,再依据外设生产商的特殊要求对功能块加以配置,为应用端点建立管道,并把若干端点组合成应用界面。

1.2.2 宿主的工作流程

位于宿主最上层的客户应用程序既是数据的生产者也是消费者,当它要求进行数据传输时,将发送输入/输出请求包(IRP)给串行总线驱动程序,后者将调用其下面各层来完成数据的传输,USB系统的工作流程(参看图1)如下。

WDM 类型。

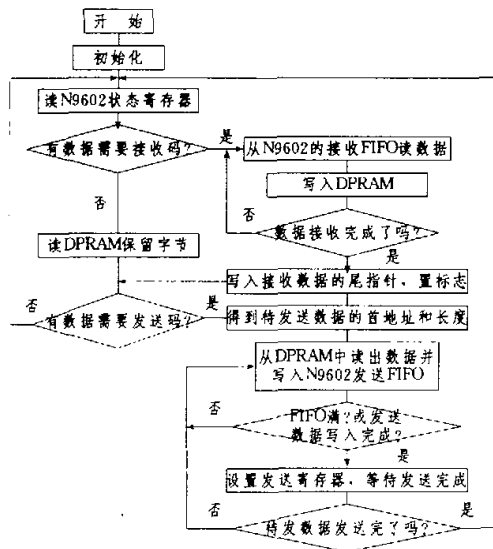


图3 USB接口电路的固件设计流程图

Fig. 3 Flowchart of firmware design of USB interface circuit

NT 驱动程序和 DLL 类似,它向操作系统显露一个名称为 DriverEntry() 的函数,在启动驱动程序的时候,操作系统将调用这个入口。DriverEntry 除了做一些必要的设备初始化工作外,还初始化一些 Dispatch 例程入口。Device 记录着设备的特征和状态信息,对系统上的每个虚拟的、逻辑的和物理的设备都有一个 Device 对象。Device Extension 是连接到 Device 对象的一个很重要的数据结构,它的数据结构是由驱动程序设计者自己来确定的,在调用 IoCreateDevice 的时候应该指定它的大小。Device Extension 其实是由操作系统为每个 Device 对象分配的一块内存。一般来说和设备有关的任何需要保持的信息都应该放到 Device Extension 里去。

在 NT 中,几乎所有的 I/O 都是包驱动的,可以说驱动程序、操作系统和其他部分都是通过 I/O 请求包来进行交互的。通过设置 Device 对象的 Flag,可以选择控制处理读写请求的 I/O 缓冲策略。在执行缓冲 I/O 时,I/O 管理器将在非分页池中分配内存,如果调用者的缓冲区比较大时,分配的非分页池也将比较大。非分页池是系统的宝贵资源,因此,如果调用者的缓冲区比较大时,一般采用直接 I/O 的方式(例如磁盘读写请求等),这样不仅节省系统资

源,另一方面由于省去了 I/O 管理器在系统缓冲区和调用者缓冲区之间的数据拷贝,也提高了效率,这对存在大量数据传送的驱动程序尤其明显。

实际软件编制过程中应注意的事项如下。

(1) USBN9602 芯片所组成接口电路的软件主要由配套驱动程序、用户程序和设备端的固件组成。用户端程序通过驱动程序与系统 USBDI(USB Device Interface)进行通信,由系统产生 USB 数据的传送动作。

(2) 用户程序需要 VC6.0 编程环境。如果想自己编制 Driver,还需要 DDK 工具。编制高质量的 Driver 是开发成功的关键,直接影响到系统的性能和可靠性。

(3) 设备端的单片机程序,除需要响应各种来自系统的 USB 的标准请求外,还需要完成各种数据的交换工作和事件处理。USB 协议中的很多 Descriptor 请求,必须按照一定的顺序准确无误地传输到主机,只要有一个数据包传输错误就会导致设备配制的失败。对设备正常工作时出现的各种事件,如收到数据、或主机请求数据等,也必须准确无误地进行处理,否则会导致设备无法正常工作甚至整个系统崩溃。

3 结束语

USB 接口的开发是一个比较复杂的工程,难点在驱动程序的开发和固件的设计。本文简介了 USB 的基本原理,并结合实际给出了一种实际的 USB 接口方案。通过实际的硬件软件设计,表明本方案是正确的、可行的。

参 考 文 献

- [1] 孙涵芳,徐爱卿.单片机原理及应用[M].北京:北京航空航天大学出版社,1997.
- [2] 原嵩,林洪,刘伟.通用串行总线的原理及实现[J].微型计算机,1999,(5):10-21.

(编辑:龙能芬)