

文章编号: 1005-8451 (2009) 05-0053-04

基于 AVR 的 USB 通讯模块设计

王 晶, 邱建东, 甘 艳

(兰州交通大学 机电技术研究所, 兰州 730070)

摘 要: 以 AVR 单片机 ATmega128 为微控制器和 USB 接口器件 PDIUSB12 芯片为核心, 设计了一种 USB 接口模块。介绍了设备的硬件实现及系统软件的体系结构, 包括固件程序设计及驱动程序设计。实现了 USB 设备与主机之间的高速数据传输。

关键词: AVR; USB; 固件; 通讯模块

中图分类号: TP391

文献标识码: A

Design of USB communication module based on AVR

WANG Jing, QIU Jian-dong, GAN Yan

(Mechanical and Electrical Technology Institute, Lanzhou Jiaotong University, Lanzhou 730070, China.)

Abstract: Designed a USB interface module based on AVR microcontroller ATmega128 and USB interface chip PDIUSB12 as the core. Introduced its hardware realization and its system software structure, including the design of firmware program and the design of driver program. Achieved the high transmission between the USB devices and the host.

Key words: AVR; Universal Serial Bus; firmware; communication module

USB (Universal Serial Bus) 就是通用串行总线, 具有高速传输、热拔插、即插即用、配置方便、

易于扩展及产品成本低等特点。是一种适应低、中、高速的 PC 机外挂总线, 而且它还支持最多 127 个外设的同时串联, 目前成为 PC 机与外部通信的主流接口。本文采用 USB 接口器件 PDIUSB12 和 AVR 系列单片机 ATmega128 设计了一种 USB

收稿日期: 2008-11-20

作者简介: 王 晶, 在读硕士研究生; 邱建东, 工程师。

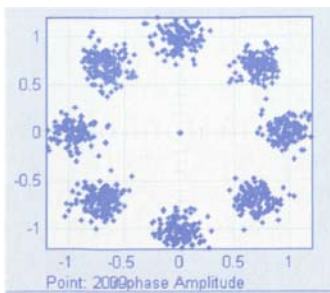


图 6 8PSK 信号的星座图

3 结束语

采用 Simulink 仿真模块建立 MPSK 系统仿真模型, 简单易行, 避免了采用 M 文件编程仿真带来的很多不便。仿真结果与理论分析高度一致。采用 Simulink 直接进行仿真, 可以根据实际需要反

复修改电路模型的参数, 在频谱仪模块和离散时间星座图仪模块的窗口上实时观察仿真结果, 达到最佳设计要求, 极大地简化了设计流程, 减轻了设计负担, 体现了其优越的性能, 具有一定的实用价值, 将会得到广泛应用。

参考文献:

- [1] 张重阳. 数字移动通信技术[M]. 西安: 西安电子科技大学出版社, 2006.
- [2] 徐明远, 邵玉斌. MATLAB 仿真在通信与电子工程中的应用[M]. 西安: 西安电子科技大学出版社, 2005.
- [3] 邓 华. MATLAB 通信仿真及应用实例详解[M]. 北京: 北京邮电大学出版社, 2003.
- [4] 薛定宇, 陈阳泉. 基于 MATLAB/ Simulink 的系统仿真技术与应用[M]. 北京: 清华大学出版社, 2002.
- [5] 刘 敏, 魏 玲. MATLAB 通信仿真与应用[M]. 北京: 国防工业出版社, 2002.

接口模块。该模块可方便地应用在其他产品中,有利于产品的整合。

1 硬件系统的实现

1.1 USB 接口芯片的选择

PDIUSB11/12是一个性能优化的USB器件,通常用于微控制器和系统,且与微控制器通过高速通用并行接口进行通信,也支持本地DMA传输。该器件采用模块化的方法实现一个USB接口,允许在众多可用的微控制器中选择最合适的作为系统微控制器,也允许使用现存的体系结构并使固件投资减到最小。它集成了SIE、FIFO存储器、收发器以及电压调整器;可与任何外部微处理器实现高速并行接口,速度可达2 MB/s;在批量模式和同步模式下均可实现1 MB/s的数据传输速度;可通过软件控制与USB的连接。

PDIUSB12完全符合USB 1.1规范,也能适应大多数设备类规范的设计,如成像类、大容量存储类、通信类、打印类和人工输入设备等,

PDIUSB12是纯粹的USB接口芯片,主要特点是价格便宜,接口方便,可靠性强,尤其适用于产品的改型设计。因此非常适用于外围设备,如打印机、扫描仪、外部大容量存储器(Zip驱动器)和数码相机等。现在用SCSI实现的很多设备如用USB来实现,可直接降低成本。PDIUSB12挂起时的低功耗及LazyClock输出符合ACPI、OnNOWT和USB电源管理设备要求。低功耗工作允许实现总线供电的外围设备。PDIUSB12还集成了像SoftConnect, GoodLink, 以及可编程时钟输出、低频晶振和终端电阻等特性。所有这些特性都能在系统实现时节省成本,同时在外围设备上很容易实现更高级的USB功能。

1.2 微控制器的选择

利用PDIUSB12芯片来实现USB接口,设计者可以根据需要和实际情况合理选择微控制器进行开发。这样,不仅减少了开发成本,而且大大缩短了开发周期。本设计微控制器MCU采用ATmega128单片机,它为基于AVR RISC结构的8 bit低功耗CMOS微处理器。由于其先进的指令集及单周期指令执行时间,ATmega128单片机的吞吐速率高达1 MIPS/MHz,故可以缓减系统的功

耗和处理速度之间的矛盾。且其内部带有128 kbit的系统内可编程FLASH程序存储器,具有写的过程中还可以读的能力,即同时读写。适合于做纯粹的USB接口芯片的MCU。

1.3 硬件电路的设计

在选定USB接口芯片PDIUSB12之后,就要选择一个微控制器芯片进行协议处理和数据交换。USB设备接口的硬件设计主要以PDIUSB12为中心,设计它与AVR系列单片机ATmega128之间的连接。这里采用总线供电方式设计全速USB设备接口,其接口电路如图1。

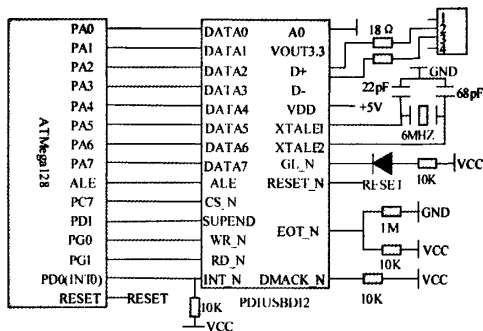


图1 ATmega128与PDIUSB12的电路连接

硬件电路说明:

(1) 由于PC机接供给USB的是5V电压,因此GND接USB接口中的GND, VDD接USB接口中的VDD, VOUT3.3悬空, D+和D-分别接USB接口中的对应位;

(2) 由GND接一个1 MΩ电阻,再从USB电源VCC接一个10 kΩ的电阻,然后将1 MΩ和10 kΩ接在一起,引至EOT_N引脚,借此检测USB设备是否已经连接到USB口,SoftConnect功能只有在检测到此信号时才会进行连接,如果不接此信号,则D12总认为还没有插入到USB接口中,SoftConnect永远不会连接;

(3) CS_N和DMACK_N信号任一为低时,均可选中此芯片。如果不使用DMA方式,则应将DMACK_N接高电平,使用CS_N作为片选;

(4) ALE和A0的接法必须组合在一起,根据USB芯片与MCU之间数据地址总线情况的不同,有两种接法:

a.如果总线和地址复用,则可以将ALE接至MCU的ALE, A0接高电平,这种情况下D12会

在ALE的下降沿锁存地址信号,直接将数据写入对应的USB地址码中。比如D0是D12的地址使能命令字,则直接将要使能的USB地址写入D0中,在MCU的ALE下降沿,D12先将D0保存下来,然后再将端口地址作为数据送至D12;

b.如果MCU总线和地址是分开的,则ALE总接低,A0为高时表示DATA0~DATA7上收到的是命令字,A0为低时表示收到的是数据,通过将地址线的高位接至A0,D12就可以有独立的命令和数据端口,同样,端口使用时,先将D0写入命令端口,再将端口号写入数据端口,不同的端口其实只有A0的变化,从而告之D12当前发送的是什么内容。

本设计采用总线和地址复用接法。

(5) GL_N接发光二极管后加电阻接高电平,它的亮暗将反映USB芯片的工作状态;

(6) PDIUSBD12片内集成6 MHz—8 MHz时钟乘法PLL,可以使用低成本的6 MHz晶振,晶振电路所用的电容分别为22 pF和68 pF,可提高电路的稳定性。

2 软件系统的实现

2.1 固件程序设计

完成上述硬件电路的连接后,即可开始准备固件程序,主要包括3部分:(1)初始化单片机和所有的外围电路(包括PDIUSBD12);(2)主循环部分,主要完成Setup包的建立,其任务是可以中断的;(3)中断服务程序,其任务是对时间敏感的,必须马上执行。固件结构如图2。

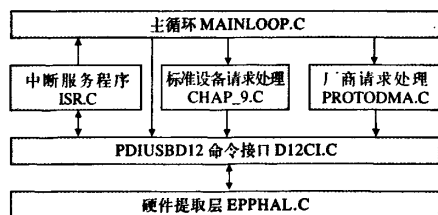


图2 固件结构

图2中各模块的功能如下:

(1) 硬件提取层:这是固件中的最底层代码,它完成微处理器与USB器件底层电路间的数据通

信工作,对单片机的I/O口和数据总线等硬件接口进行操作;

(2) PDIUSBD12命令接口:对PDIUSBD12器件进行操作的模块子程序集。是一套压缩了所有访问PDIUSBD12功能的命令接口;

(3) 中断服务程序:这部分代码处理由PDIUSBD12产生的中断,当PDIUSBD12向单片机发出中断请求时,读取PDIUSBD12的中断传输来的数据,并设定事件标志“EPPFLAGS”和Setup包数据缓冲区“CONROL_XFER”传输给主循环程序;

(4) 标准请求处理程序:对USB的标准设备请求进行处理;

(5) 厂商请求处理程序:对用户添加的厂商请求进行处理;

(6) 主循环程序:主循环检查事件标志和SETUP包数据缓冲区,并发送USB请求、处理USB总线事件和用户功能处理等。它还包含人机接口的代码。

固件程序可实现USB的枚举过程。在主机与USB设备通信前,主机必须为该设备分配一个地址,主机需要知道设备支持哪些传输类型,有几个端点,电源要求,及安装何种驱动程序,主机通过一个被称为“枚举”的信息交换过程来完成这些工作,只有枚举成功后,D12与主机之间才能进行数据传输。

USB标准设备请求是用来完成USB设备枚举的命令,USB设备必须对标准设备请求做出响应。所有的标准设备请求都是使用默认管道(控制端点0)来传输的。

枚举实际上就是从设备读取一些信息,知道设备是什么样的设备,如何进行通信,这样主机就可以根据这些信息来加载合适的驱动程序。

2.2 驱动程序设计

Windows下的USB驱动程序遵循WDM标准,有用户模式驱动程序和内核模式驱动程序两种,而内核模式驱动程序同时也包含类驱动程序(Class Driver)、客户驱动程序(Client Driver)和微驱动程序(Mini-driver)等类型。

USB驱动程序是WDM类型,该类型的驱动程序采用分层驱动的方法,在应用程序和硬件之间存在几个驱动程序层次。USB驱动程序分为3层,

描述如下:

(1) 物理设备驱动程序: 管理实际数据传输和控制特定类型物理设备操作的驱动程序, 包括开始和完成 I/O 操作, 处理中断和执行特定设备要求的任何差错处理;

(2) 中间驱动程序: 该层次驱动程序拦截和修改高层发送给系统类驱动程序的请求。Windows NT 允许在物理设备驱动程序上层存在任意数目的中间驱动程序;

(3) 文件系统驱动程序: 该层驱动程序通常负责维护各种文件系统所需要的文件磁盘结构。

本驱动程序使用的软件有: Microsoft VC++ 6.0、Microsoft WindowsXP DDK 及 Driver Studio。并按这个顺序安装这三中软件。其中 Driver Studio 是一套用来简化 Windows 平台下设备驱动程序开发、调试和测试的工具包。用它不仅写代码的时候思路清晰, 而且与 DDK 的结合很好, 可以方便地建立驱动程序的框架。

安装软件后运行 Driver Studio 中的 Driver Works 和 Driver wizard 驱动程序向导, 按照每一步要求填入相应的数据, 就可以生成一个框架性的驱动程序代码以及一个安装信息配置文件 (.inf 文件)。修改编译一下 DriverStudio 产生的测试程序 Test_TEST 程序, 就可以通过命令行来测试驱动程序了。

根据要求装入驱动程序后, 可以通过 USBDE-BUG 调试工具软件来测试 USB 设备, 对它进行数据的读写操作, 实时监测从 USB 设备发来的数据。

3 USB 设备与 PC 机间的数据通信

驱动程序安装成功后, 可通过相应的应用程序实现数据的采集及存储等功能。

下面是简单读写的应用程序的部分代码:

```
while(1)
{ if (bEPPflags.bits.bus_reset)
    { CLI();
      bEPPflags.bits.bus_reset = 0;
      SEI();
    }
  if (bEPPflags.bits.setup_packet)
    { CLI();
```

```
bEPPflags.bits.setup_packet = 0;
    SEI();
    control_handler();
  } // if setup_packet
  if (bEPPflags.bits.configuration)
  {
    if (bEPPflags.bits.ep2_rxdone )
      { // 端点 2 收到数据
        CLI();
        bEPPflags.bits.ep2_rxdone = 0;
        // 清空端点 2 收到数据标志
        SEI();
        D12_WriteEndpoint(5, 10, EpBuf); ,
        // 发送 10 个字节到 PC 机
      }
    }
  }
}
```

这段代码通过对端点 2 的控制来实现 USB 设备的接收收数据, 再用 VC++6.0 编写相应的 PC 端应用程序便可实现 USB 设备与 PC 机间的数据通信。采用 PDIUSBD12 和 AVR 高速单片机开发 USB 接口, 可满足在工业生产和科学技术研究中, 对各种数据进行采集并对受控设备发送控制信息日益发展的需求。

4 结束语

USB 的高速率、易扩展性、热插拔等特性适合于现场采样和科研活动, 并可方便灵活地控制采集过程。在不久的将来, 外设将尽可能多地通过 USB 设备与主机相连。USB 接口在嵌入式系统中也将成为数据存储、交换及 PC 机交换数据方式的首选之一。

参考文献:

- [1] 包海波, 宋文涛, 张灵艳. 基于 Windows NT 平台的 USB 设备驱动程序开发[J]. 微计算机信息, 2004, 20 (12): 132-133.
- [2] 周立功. PDIUSBD12 USB 固件编程与驱动开发[M]. 北京: 北京航空航天大学出版社, 2003.
- [3] 陈冬云, 杜敬仓, 何柯燕. ATmega128 单片机原理与开发[M]. 北京: 机械工业出版社, 2005.