

基于 USB 通信的多功能智能插座

汤荣秀 陈三宝 杜占磊 武汉理工大学自动化学院(430063)

Abstract

This article brings an imagine about a kind of new socket.The socket is a device used in collection of information in buildings.A design of the circuit and software about the USB communication is introduced in this paper.In order to use this socket in industry job-site,this paper preliminarily approaches a transfer card about the USB to RS-485.

Keywords:USB,firmware,drive-program of device,transfer card

摘要

本文提出了一种关于新型智能插座的设想。该插座是一种家庭信息采集设备,具有普通插座的功能能够监测家用电器的工作状态,并能实现网络监测功能,通过其内部通信接口与上位机通信,进一步可以连到互联网上。文章着重介绍了该插座与上位机通信的 USB 实现。文章对于通信的硬件电路和软件编程进行了初步设计。为了将该插座的用途拓展到工业现场,文章对使用一种 RS485 到 USB 接口转换卡进行了初步探讨。

关键词:USB,固件,设备驱动程序,转换卡

为了实现插座的信息化和智能化,我们将研究设计的智能信息系统与插座结合,将该智能设备与电源插座合为一体,设计成多功能智能监测插座。该插座由于在建筑电气安装时一次性固定完成,采用照明电源供电,人们使用时就好像使用普通电源插座一样,免除了过多的连接线。

1 系统功能与组成

1.1 功能

该多功能智能监测插座是在普通插座的基础上,结合信息的需要,增加了如下功能:

- 1)对接入插座的电器的电压、电流、频率进行监测。
- 2)对插座工作环境温度进行监测。

3)将监测到的电器的工作电压或电流与其额定值进行比较,如果超过额定值,主机就发送命令给执行单元,自动切断该电器的电源,实现了保护功能。

4)通信功能:将监测到的信息经过通信接口传给上位机,在家庭内部或工业现场可以连成一个局域网。在主机端人们可以通过一个人机界面定期或不定期查询接入插座的诸电器的工作状况。如果想实现进一步的方便,可以将该局域网通过公用电话网连到 Internet 上,这样就可以在家庭以外的地方用手机或 PC 机随时方便地查询家用电器的工作情况了。

1.2 组成

该智能监测插座由电源系统、监测系统、控制系统、通信系统四部分组成。

1)电源系统:完成照明电源到直流稳压电源的变换,提供给该智能系统工作电源。由隔离降压、整流、滤波、稳压等电路组成。

2)监测系统:由电压传感器、电流传感器和温度传感器以及模/数变换电路组成,负责信息的采集处理。

3)控制系统:由微处理机(单片机)和执行器件组成。微控制器一方面控制采集部分的模拟量到数字量的转换;另一方面接受上位机的命令,然后将这个命令交给执行单元去完成,比如切断用电器的电源。

4)通信系统:该插座的通信系统主要是在其内部增加一个通信接口,以便联网和服务器进行通信。由于用该设备构成局域网,设计的时候要采用分层次的设计思想,在这篇文章里,只考

虑设计最低的一层:物理层的通信实现。

2 物理层通信的设计

在串行通信口的缺点中有三项最为突出:低速、复杂的电缆和端口数的限制。通用串行总线(USB)不仅克服了所有这三项缺点,而且具有支持热插拔、容易扩展、低成本、低干扰等优点。USB 标准已经成为业界普遍认同的一种事实上的接口标准,支持这一标准各种新产品正在大量涌现,而且现在的微型计算机上都有 USB 接口,所以我们就选择 USB 作为该多功能插座和主机串行通信的接口。

2.1 USB 接口芯片 PDIUSB D12

Philips 公司推出的 PDIUSB D12 是一种 USB 专用的接口芯片(以下简称 D12),它完全遵从 USB1.1 协议。它的 8 位数据并行接口可以方便地与外部 CPU 进行连接,就好比 51 系列微处理器的一个带有 8 位数据线和一位地址线的存储器设备,支持多路地址/数据总线复用方式和非复用方式以及直接存储器(DMA)方式。D12 的串行接口引擎实现全部 USB 协议层,完全由硬件实现而不需要固件的参与。该模块的功能包括同步模式的识别、并行/串行的转换、位填充/解除填充、CRC 校验/产生、PID 校验/产生、地址识别和握手评估等。在高速态时,D12 用内存管理单元来缓冲 USB 和微处理器并口之间的速度差异。

2.2 通信硬件电路设计

本文我们选用 89C52 单片机作为该系统的微控制器,其内部有 8K 字节的程序存储器可以存放 USB 固件程序。PDIUSB-D12 和 89C52 的电路连接如图 1 所示。该接口电路工作在非 DMA 传输方式,PDIUSB D12 与 89C52 的数据交换采用中断方式,其中断通过图中的外部中断 0(INT0)来完成。在这一方式下,89C52 通过控制 D12 来使集线器与主机通信并完成 USB 协议的处理包括描述符请求、地址设置、端点的配置等。图中,地址线 P2.7 作为 D12 的片选,P2.6 作为 D12 的命令/数据选择线,地址 0X7FF 为写命令,0X3FF 为读写数据。D12 输出状态可以通过 LED 进行监控,当该 USB 设备连接建立以后,LED 就会发光,数据传输时 LED 会闪烁,常亮或一直不亮都说明 USB 接口有问题。

2.3 通信系统软件设计

USB 设备的软件设计主要包括两部分:一、USB 设备端的

单片机软件,被固化到微处理器的程序存储器里,称为固件(Firmware),主要完成USB协议处理与数据交换。二、PC端的程序,由USB驱动程序和用户服务程序两部分组成,用户服务程序通过USB驱动程序通信,由系统完成USB协议的处理与数据传输。

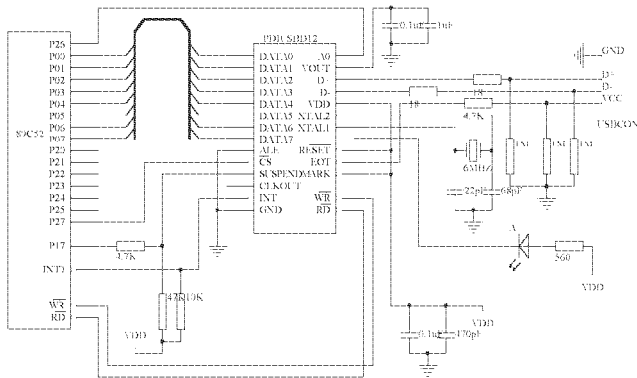


图1 USB接口通信电路

2.3.1 USB设备固件(firmware)设计

设备固件是设备运行的核心,采用汇编语言设计。其主要功能是控制芯片PDIUSBD12接受并处理USB驱动程序请求(如请求设备描述符、请求或设置设备状态、请求设备设置、请求或设置设备接口等共10种USB1.1标准请求);控制芯片PDIUSBD12接受应用程序的控制指令;控制A/D模块的数据采集;通过PDIUSBD12存储数据并实时上传PC。

下位机程序主要的工作是初始化D12,通过D12进行数据传输以及按协议对数据进行格式转换。USB单片机控制程序通常由三部分组成:第一,初始化部分,完成单片机和所有的外围电路(包括D12)的初始化;第二,主循环部分,等待来自数据采集设备或上位机的数据,并启动数据格式转换程序,进行数据通信,是固件的主体部分;第三,中断服务程序,由上位机和数据采集设备触发,进行一些低工作量的实时处理(如置相应标志位),然后在主循环部分对数据作进一步的处理。总之,主程序主要是处理数据,中断服务程序主要是以最快的速度进行数据传输。

2.3.2 USB设备驱动程序

对PC而言,无线USB传输模块是它的一个USB外设,因此必须提供USB设备的驱动程序。驱动程序的设计采用WDM模型,使用DriverStudio进行开发。DriverStudio是一款优秀的驱动程序开发软件。启动DriverStudio中DriverWizard,填写USB设备的端点配置,便会自动生成驱动程序源代码和相应的测试程序源代码,只需稍做改动便可用VC++6.0编译运行。驱动程序向用户程序提供标准的IO接口:WriteFile()、ReadFile()、DeviceIoControl()。用户程序调用相应的函数,即可实现对USB设备的访问。下面的例子是实现向USB总线写数据的功能。

```
hFile = open_file(threadParam->pipe_name); //打开需要使用的端点
.....bResult = WriteFile(hFile,
    threadParam->pclouBuffer, //等待传送的数据缓冲区地址
    threadParam->uiLength, //数据长度
    &nBytes,
    NULL); //向该端点写入数据
if (bResult != TRUE) { //出错处理
    DeviceIoControl(hFile, //复位该端点
    IOCTL_D12_RESET_PIPE,
    0, 0, 0, 0,
```

```
&nBytes,
NULL);}
```

2.3.3 系统应用程序设计

PC机应用程序是该插座数据采集系统的中心,在Visual C++6.0开发环境下进行。根据实际要求,本设计需要在软件中对采集的数据进行整理、分析并显示。限于篇幅,这里只介绍应用程序与USB设备通信的程序设计部分。在应用层,用Visual C++6.0编制的程序通过WDM^[1]设备驱动程序与USB设备进行数据交换。应用程序向USB设备传递数据的部分关键代码如下:

```
Void CmcausbView::senddata(char comusb,int time0,int data)
{
...
//打开USB设备
wsprintf(FileName,SCANUSB,i);
m_Chand=CreateFile(FileName,
    GENERIC_WRITE|GENERIC_READ,
    0,
    NULL,
    OPEN_EXISTING,
    FILE_FLAG_DELETE_ON_CLOSE,
    NULL);
if(m-Chand==INVALID_HANDLE_VALID)
    MessageBox("open usb device fail!"); //发出写请求给USB设备文件
m-whand=WriteFile(m-chand,buff,16,&rwet,null);
if(m-whand==0)
    MessageBox("write usb device fail");
free(buff); //完成USB设备的使用,关闭文件句柄
close Handle(m-chand);
}
```

3 工业用的智能监测插座

3.1 USB通信在工业现场的局限

USB电缆的最大传输距离是5米,因此上述的USB通信方案仅适用于家庭用的监测插座。即使增加了中继或HUB,USB传输距离通常也不超过几十米,这对工业现场而言显然是太小了。目前,工业现场有大量采用RS-485传输数据的采集设备。RS-485总线传输距离可以达到1200m以上,并且可以挂接多个设备。不足之处是传输速度慢、可靠性差、需要板卡的支持、成本高、安装麻烦等。RS-485的这些缺点恰好能被USB所弥补,而USB传输距离的限制又是RS-485的优势所在。为此,我们采用一种RS-485和USB的接口转换卡来解决上述问题。

3.2 RS-485和USB转换卡的原理与硬件电路图

3.2.1 转换卡的工作原理

在采集现场,数据采集设备将采集的数据利用485总线进行传输。在主机端,利用转换卡接收来自485总线的数据并通过USB接口传送至PC进行分析处理。而主机向设备发送数据的过程正好相反,主机向USB口发送数据,通过485/USB转换卡转换为485协议向远端输送。转换卡的主要功能是完成信号电平转换和数据格式转换。RS-485和USB电平的转换主要依靠硬件,通过各自的接口芯片将信号转换为TTL电平。数据格式的转换则主要是软件方面的工作,MCU读到某一个接口的数据,然后按照另一个接口协议中对数据帧的要求进行打包。设计的核心是通过MCU协调两个接口的数据收发,由于相当部分的工

(上接第 17 页)

作已经由各个接口电路完成,所以设计工作量为减少。

3.2.2 转换卡硬件电路

硬件部分主要是 MCU 与 485 和 USB 的接口电路设计,如图 2 所示。微控制器采用 89C2051,USB 接口芯片为 PDIUSBD12,485 接口芯片为 MAX485。

3.3 转换卡软件设计

软件设计包括单片机软件(固件)设计和主机部分软件设计。单片机软件使用伟福公司提供的 WAVE 仿真软件开发,并通过其仿真器进行在线调试。主机部分软件又包括驱动程序和应用程序两部分,分别使用 DDK 和 VC6.0 生成。

4 结束语

对于该智能监测插座的研究目前仍然处于理论的设想与设计过程中,要做出产品来,仍然有很长的路要走,需要更多的人投入更多的精力。尤其是利用 USB 通信这一块,USB 驱动程序的编写是个难点。随着技术的不断更新、完善和发展,相信该多功能插座的实现会越来越容易,在未来的信息家电市场上,必将占有一席之地。

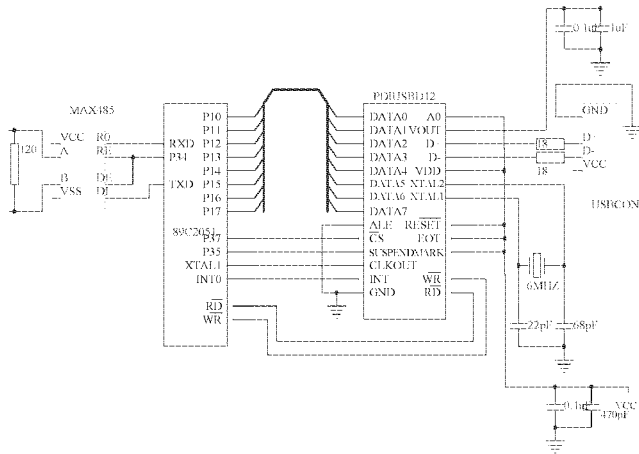


图 2 USB 接口转换卡电路图

参考文献

- 1 徐爱钧. 单片机高级语言 C51 Windows 环境编程与应用[M]. 北京: 电子工业出版社, 2001 [收稿日期: 2003.9.16]