

基于 USB 接口的数模转换系统设计

在控制系统中经常用到一些模拟信号,通常使用数模转换器输出所需的模拟信号。计算机控制数模转换器需要借助外部总线接口,USB 接口是常用的外部总线接口,用来控制数模转换器非常便捷。作者以典型的 USB 接口芯片和 D/A 转换器芯片为例,详细说明了基于 USB 接口的数模转换系统的电路及程序设计方法。

0 引言

在控制系统中经常需要产生和控制一些模拟信号,而常用的微处理器外部总线接口为数字信号,无法直接产生需要的模拟信号,这就用到了 D/A 转换器。可以通过 USB 接口来控制 D/A 转换器,使其输出要求的模拟量电压或模拟量电流。

USB 接口作为微处理器常用的外部总线接口,目前已经得到了广泛的应用,采用 USB 接口减少了开发的时间、风险以及费用,从而用最快捷的方法实现最经济的解决方案。

本文以 USB 接口芯片 PDIUSB12 和电压输出型 D/A 转换芯片 AD558 为例,详细阐述了基于 USB 接口的 DAC 系统电路及程序设计方法。

1 USB 接口芯片 PDIUSB12 简介

本系统选用的 USB 接口芯片是 Philips 公司生产的 USB1.1 接口芯片 PDIUSB12,它是一款性价比很高的 USB 器件,通常用作微控制器系统中实现与微控制器进行通信的高速通用并行接口。它还支持本地的 DMA 传输。PDIUSB12 芯片的主要特性如下:

- 符合 USB1.1 规范;
- 集成了 SIE、FIFO 存储器、收发器以及电压调整器;
- 完全自治的直接内存存取 (DMA) 操作;
- 集成 320 字节多结构 FIFO 存储器;
- 在批量模式和同步模式下均可实现 1MB/s 的数据传输速率;
- 具有良好 EMI 特性的总线供电能力;
- 可通过软件控制与 USB 的连接;
- 具有可编程的时钟频率输出;
- 符合 ACPI、OnNOW 和 USB 电源管理的要求;

- 高于 8kV 的在片静电防护电路;
- 双电源操作: $3.3 \pm 0.3V$ 或扩展的 5V 电源, 范围为 $3.6 \sim 5.5V$;

所有这些特性都为系统显著节约了成本, 同时使 USB 功能在外设上的应用变得容易。低功耗使它可以应用于通过总线供电的外设。

2 D/A 转换芯片 AD558 简介

本系统选用的 DAC 芯片是 ADI 公司推出的 8 位电压输出型 D/A 转换器 AD558。电压输出型 D/A 转换器主要特点是没有放大部分的延迟, 可以做到很高的速度, 但负载能力有限, 一般用于高阻抗负载的场合。

D/A 转换所采用的基本方法是将数字量转化成二进制数据。其每一位产生一个相应的电压或者电流, 而这个电压或者电流的大小正比于相应的二进制位的权。最后将这些电压或者电流相加并输出。

一个完整的 D/A 转换器主要包括 3 部分, 即加权电阻解码网、数字量控制的电子开关组合和由运算放大器构成的电流电压转换器。D/A 转换器的原理如图 1 所示。

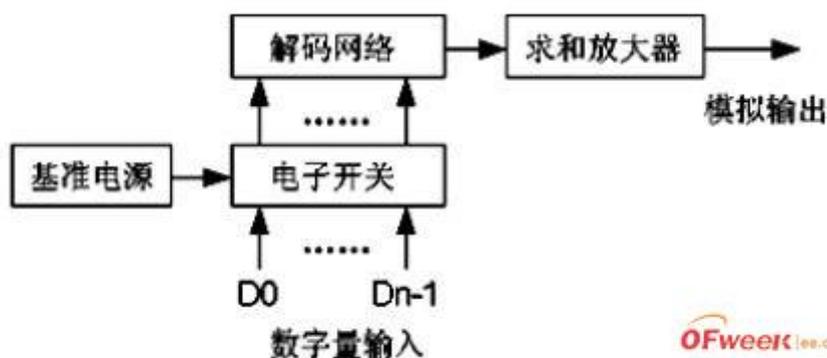


图1 D/A转换器的原理图

一般来说, 并行 D/A 转换器的数字-模拟转换速度要比串行 D/A 转换器要高。并行 D/A 转换器转换速度快的原因是在数字量的输入端, 各个数字位是同时输入到转换器相应的端口的。整个器件的转换时间取决于电源电压的建立时间和各个权位的求和时间。而这些时间一般都很短, 一般可以达到纳秒或微秒量级。因此, 对于速度要求比较高的场合, 应该优先选用并行 D/A 转换器。

在 AD558 型 D/A 转换器中, 电子开关组和输入二进制数 $D_0 \sim D_7$ 相对应。当某个二进制位为 1 时, 对应的电子开关闭合, 基准电压 V_{ref} 连接加权电阻解码网络, 使该支路电阻上有电流通过; 当某个二进制位为 0 时, 对应的电子开关断开, 该支路电阻上便无电流通过。加权电阻解码网络各个分支的电阻值与输入的二进制数据 $D_0 \sim D_7$ 的权成反比关系。权大的电阻值小, 权小的电阻值大。D/A 转换器

根据各个权位的情况以及输入的二进制数，通过求和及电流电压转换电路，得到总和的模拟电压值并输出。

在 D/A 转换器中，数字量的位数也反映了 D/A 转换器的分辨率，即输入数字量发生单位数字的变化时，对应的模拟量输出值的改变。对于 8 位电压型 D/A 转换器，输入数字量发生单位变化时，模拟量输出的改变为 $V/255$ ，其中 V 为电压输出的满量程。

AD558 的主要性能指标如下：

- 8 位并行接口；
- 电压的输出范围 $0\sim+10V$ ；
- 相对精度 $\pm (1/2)$ 最低有效位；
- 输出转换建立时间 1 s ；
- 使用单一电源供电，电源电压范围 $+4.5\sim+16.5V$ ；
- 内部具有基准电压源，无须外接基准源；
- 内部集成有数据锁存器；
- 75mW 低功耗。

3 电路设计

本系统采用 USB 控制芯片 PDIUSB12 和 D/A 转换器 AD558 来实现 DAC 控制。计算机可以通过 USB 接口控制 AD558 输出所需的各种模拟信号。由于计算机的 USB 接口具有向外供电的功能，因此这里采用 USB 接口的电源为 PDIUSB12 芯片供电。

USB 接口部分的电路原理图如图 2 所示。

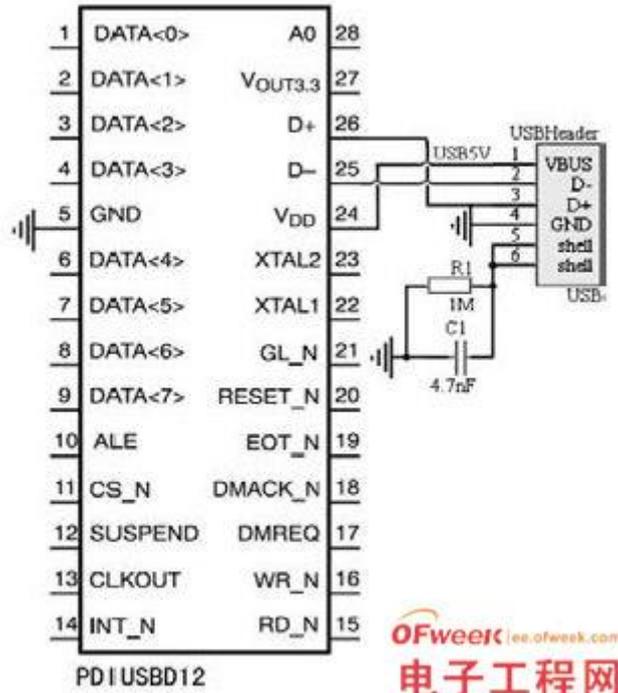


图2 USB接口部分电路原理图

AD558 的数据总线连接到 PDIUSB12. AD558 工作于 $0 \sim +10V$ 模拟电压输出模式。电容 C9 和 C10 用于改善输出波形。AD558 采用单一的 12V 供电，不需要外接基准电压源。数模转换部分的电路原理图如图 3 所示。

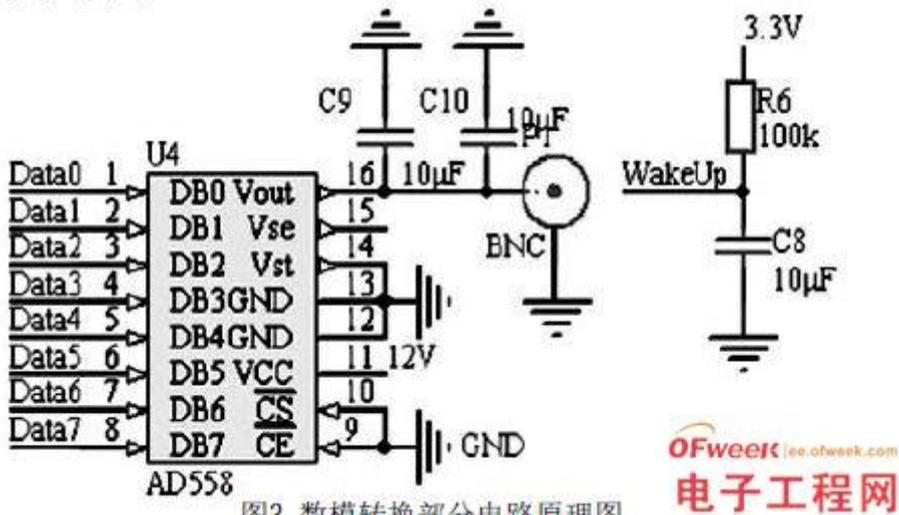


图3 数模转换部分电路原理图

在制作 PCB 的时候，应遵守高速 USB 印刷电路板的要求。为了达到较高的信号质量，需要采用 4 层板甚至 6 层板来布线。将 USB 芯片放置在信号层，并尽可能离 GND 层近些。

USB 接口中的 D+和 D-用于高速的 USB 数据传输，因此这两根信号线直接影响 USB 接口电路的稳定性，在印制电路板布线的时候需要仔细布置。D+和 D-的走线应尽可能短且相等，并且合理设置 D+和 D-的导线宽度和间距。D+和 D-之间

的差分阻抗应该为 $90\Omega \pm 10\%$. 保证在 D+ 和 D- 信号的下面是完整的 GND 层。中间断开的 GND 层将导致差分阻抗不匹配，并增加信号的干扰。

4 程序设计

上位机的程序在 VisualC++6.0 编译环境中进行设计，分为设备初始化、USB 设备检测、输出电压三个部分，程序代码分别介绍如下。

4.1 设备初始化

需要先初始化 USB 设备变量、控制端点变量和线程变量。主要是在 OnInitDialog () 函数中进行。代码示例如下：

```
BOOL CUSBADDlg::OnInitDialog()
{
    CDialog::OnInitDialog(); //初始化
    ASSERT((IDM_ABOUTBOX & 0xFFF0) ==
IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);
    CMenu* pSysMenu = GetSystemMenu(FALSE);
//菜单指针
    if (pSysMenu != NULL) //菜单
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty()) //判断是否为空
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING,
IDM_ABOUTBOX, strAboutMenu);
        }
    }
    SetIcon(m_hIcon, TRUE); //设置图标
    SetIcon(m_hIcon, FALSE);
        //添加初始化代码
    LoopThread1=NULL; //初始化线程1
    Looping1=false; //初始化循环变量1
    LoopThread2=NULL; //初始化线程2
    Looping2=false; //初始化循环变量2
    return TRUE; //返回值
}
```

4.2 USB 设备检测

在程序中检测连接的 USB 设备，代码示例如下：

```

CString str;
vID=USBDevice->VendorID; //VID
pID=USBDevice->ProductID; //PID
str.Format("VendorID=0x%x",vID);
SetDlgItemText(IDC_STATIC1,str); //显示VID
str.Format("ProductID=0x%x",pID);
SetDlgItemText(IDC_STATIC2,str); //显示PID
PCHAR dn=new CHAR[256];
dn=USBDevice->DeviceName; //设备名称
str.Format("设备名称: %s",dn);
SetDlgItemText(IDC_STATIC3,str); //显示设
备名称
}

```

4.3 输出电压

用于从文本框中读取输入的电压值，然后转换为浮点数，接着转换成最接近的并行数据，最后通过自定义请求输出给 USB 设备。代码示例如下：

```

void CUSBDAIDlg::OnButton2()
{
    CString str;
    double stdata;
    UCHAR DA;
    GetDlgItemText(IDC_EDIT1,str);
    stdata=atof(str); //字符串转换为浮点型
    DA=abs(stdata*255/10.0); //取整

    ept->Target = TGT_DEVICE; //控制请求
    ept->ReqType = REQ_VENDOR;
    ept->Direction = DIR_TO_DEVICE;
    ept->ReqCode = 0xB3; //自定义请求操作码
    ept->Value = 0;
    ept->Index = 0;
    PCHAR buf=new UCHAR[2];
    ZeroMemory(buf, 2); //清空缓冲区
    LONG buflen = 2;
    buf[0]=0xA1;
    buf[1]=DA; //DA数据
    ept->XferData(buf,buflen); //控制传输
}

```

5 小结

本文通过一个完整的实例，讲解了如何使用 USB 接口芯片实现上位机控制 D/A 转换输出模拟电压。D/A 转换器将数字信号转换为模拟信号，可以利用成熟的数字电路技术实现控制，应用非常广泛。熟练掌握这种方法以后对于相关领域的设计工作十分有用。