

基于 DSP 的 USB 接口设计

黄益彬, 金伟明

(东南大学 江苏 南京 210096)

摘要:在数据采集系统中,利用 DSP 作为微处理器对采集到的数据进行处理,再通过 USB 接口将处理过的数据传输到 PC 机上进行存储,这样可以实现数据的快速采集与存储。采用 TI 公司的 TMS320F2812 作为微处理器控制芯片,并使用 Philip 公司的 ISP1581 作为 USB 收发控制芯片,利用两者的速度优势,能够实现 DSP 与 PC 机之间的高速数据传输,从而使大量数据的快速、实时的采集、处理与存储成为可能。

关键词:ISP1581;USB;DSP;TMS320F2812

中图分类号:TP334.7

文献标识码:B

文章编号:1004-373X(2006)19-049-03

USB Interface Design Based on DSP

HUANG Yibin, JIN Weiming

(Southeast University, Nanjing, 210096, China)

Abstract: In data acquisition system, using DSP to process the collected data, and then transferring the processed data to PC by USB interface, this can archive fast data acquisition and storage. Selecting TI Corporation's TMS320F2812 for microprocessor control chip and Philip Corporation's ISP1581 for USB receiver/transfer control chip, it can achieve the high speed data transfer between DSP and PC by their speed, therefore, it can realize high speed, real-time collection, process and storage for mass data.

Keywords: ISP1581; USB; DSP; TMS320F2812

1 概述

TMS320F2812 是 TI 公司新推出的、目前国际市场上功能强大的 32 位定点 DSP 芯片。他既具有数字信号处理能力,又具有强大的事件管理能力和嵌入式控制功能,特别适用于有大批量数据处理的测控场合,如工业自动化控制、智能仪器仪表、马达伺服控制系统等。

ISP1581 是一种价格低、功能强的通用串行总线(USB)接口器件,他完全符合 USB 2.0 规范,并为基于微控制器或微处理器的系统提供了高速 USB 通信能力。ISP1581 与系统的微控制器/微处理器的通信是通过一个高速的通用并行接口来实现的。

利用 DSP 强大的数据处理功能,可以将采集到的数据经过 DSP 进行快速复杂的处理,然后通过 USB 接口传送到 PC 端作进一步处理,如图 1 所示。

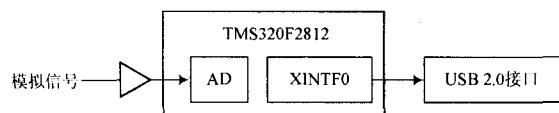


图 1 系统结构图

2 硬件设计

ISP1581 工作于通用处理器模式,8 位地址总线,16 位数据总线,片选信号连接 DSP 的外部存储器扩展的 XZCS0AND1,中断输出接 DSP 的 XINT1,ISP1581 自供电。DSP 的数据线通过 74LVC16245 双向缓冲器与 ISP1581 的数据线相连,具体电路如图 2 所示。

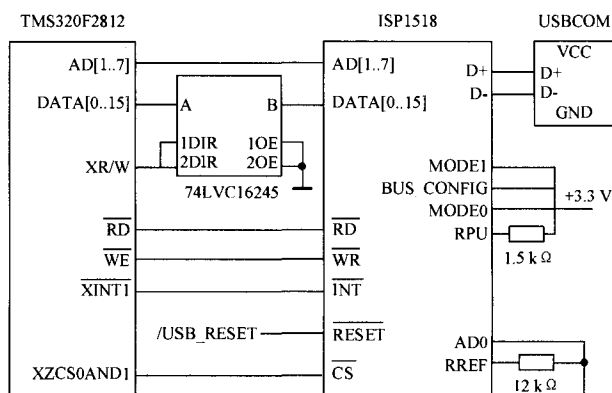


图 2 DSP 与 ISP1581 的连接

3 软件设计

软件设计主要是 DSP 端 ISP1581 的固件程序的设计,

PC 机端的驱动程序及测试程序可以参考 Philip 公司的开发包进行设计。下面介绍 ISP1581 的固件程序的设计。

3.1 初始化寄存器

固件程序必须对 ISP1581 的寄存器初始化来配置 I/O 口的信号电平以满足系统设置的要求。表 1 列出了 ISP1581 的地址寄存器、方式寄存器以及中断配置寄存器的一般初始化方法。

表 1 地址寄存器(00H),方式寄存器(0CH)
中断配置寄存器(10H)初始化方法

寄存器位 Hex	寄存器位 符号	ISP1581 信号	注 释
00.7	DEVEN	无对应的信号	1 激活设备
00.6—00.0	DEVADDR	无对应的信号	指定的 USB 设备地址
0C.7	CLKAON	无对应的信号	0 在挂起状态下关掉时钟来降低功耗
0C.3	GLINTENA	INT	1 全局中断使能
0C.2	WKUPCS	\overline{CS} 和 WAKEUP	1 \overline{CS} 有效,将 ISP1581 从挂起状态唤醒(在 WAKUP 下所有的设置不变)
0C.1	PWROFF	SUSPEND	1 挂起状态 SUSPEND 引脚为高,唤醒时为低 0 唤醒到来时 SUSPEND 引脚产生一个脉冲,其他时间仍然保持低电平
0C.0	SOFTCT	RPU	0 RPU 引脚上的 1.5 k Ω 电阻与内部 D+ 断开 1 RPU 引脚上的 1.5 k Ω 电阻与内部 D+ 相连(USB 1.1 全速功能)
10.1	INTLVL	INT	1 中断发生时在 INT 引脚上产生一个脉冲 0 只要有任何中断事件等待处理,中断处于有效状态
10.0	INTPOL	INT	1 INT 保持不变中断到来时变为高 0 INT 保持不变中断到来时变为低

函数 Init_D14_SFR()完成 USB 的初始化设置。主要是激活设备,设置中断模式以及配置和使能中断等。

```
void Init_D14_SFR(void)
{
    rega_addren = 0x80;           //激活设备
    //设置模式:时钟始终有效,全局中断使能,片选唤醒
    rega_mode |= 0x88;
    //设置全部中断为模式 1,电平触发,低电平有效
    rega_intcfg = 0x54;
    //设置中断使能寄存器
    rega_intmask_low = 0xfd71;
    rega_intmask_high = 0x0000;
}
```

3.2 软件连接

初始化完成之后,通过软件的方式将 USB 的 D+ 线拉高,即通过置为 USB 的内部寄存器的相应位,使得 D+

线通过一个 1.5 k Ω 的电阻接到 Vcc 上。

3.3 程序进入循环

当完成了 USB 的软件连接后,程序将进入一个循环,等待上位机的命令的到来。一旦上位机检测到 USB 的 D+,D- 的电压发生变化(软件连接后,D+,D- 的电压将发生变化),上位机就开始对插入的新设备进行枚举,只有正确完成枚举,上位机才能加载驱动并使用 USB 设备。

3.4 枚举过程

上位机对 USB 设备进行枚举时,需要获取设备描述符、配置描述符、接口描述符、端点描述符等,并需要为 USB 设备分配一个地址。

3.4.1 获取描述符

USB 设备通过端点 0(默认端点)来读取上位机发来的枚举数据,程序根据接收到的数据来判断上位机需要 USB 设备做什么,如果上位机需要获取 USB 设备的描述符,则程序进一步判断上位机需要什么描述符,然后将相应的描述符发送给上位机。

```
void get_descriptor(void)           //获取描述符
{
    switch(USB_Device_Request.wValue){ //判断描述符类型
        case 1:                      //获取设备描述符
            .....                    //发设备描述符给上位机
        case 2:                      //获取配置描述符
            .....                    //发配置描述符给上位机
        case 3:                      //获取字符串描述符
            .....                    //发字符串描述符给上位机
        case 4:                      //保留
            reserved();break;
        case 5:                      //获取端点描述符
            .....                    //发端点描述符给上位机
        case 6:                      //获取设备限制描述符
            .....                    //发设备限制描述符给上位机
        case 7:                      //获取速度配置描述符
            .....                    //发速度配置描述符给上位机
        case 8: break;
        default: break;
    }
}
```

枚举时,需要准确无误地完成图 3 的流程才能通过枚举,部分描述符在枚举时并不需要,但有些 PC 机端的应用程序可能需要。

3.4.2 设定地址

在设定 USB 设备的地址之前,上位机和 USB 设备都用默认地址 00H 进行通信,一旦上位机获取了 USB 设备的设备描述符,上位机将给此 USB 设备指定一个新的地址,之后的通信都将以新的地址来进行。因此,当 USB 设备接收到上位机的设定地址的请求后,就必须将新地址存储到芯片的地址寄存器中去(即更改 USB 设备的地址)。设定地址的流程如图 4 所示。

```
void set_address(void)               //设定地址
{
    .....
}
```

```
//设置地址
rega_addren |= (USB_Device_Request.wValue>>8);
ep0in_ack(); //状态响应
```

当上位机取得所需的描述符并设定了地址后,将加载设备的驱动程序,这时,上位机的应用软件就可以通过驱动程序来与USB设备进行通信了。

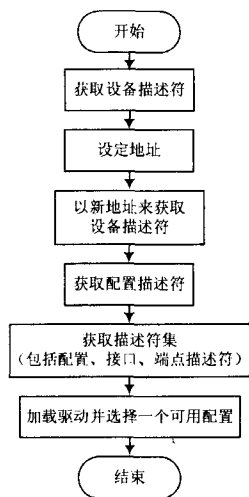


图3 枚举流程图

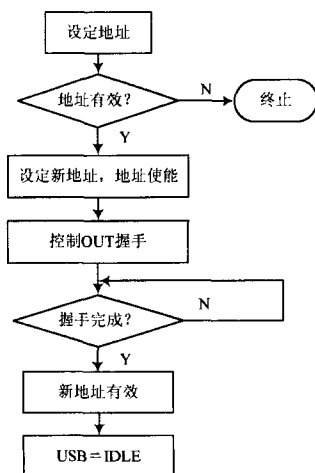


图4 设定地址流程图

4 测试

在测试USB时,需要按如下步骤进行:

- (1) 将USB电缆线的一端连接上位机(主机),另一端连接下位机(设备);
- (2) 在下位机运行USB固件程序;

作者简介 黄益彬 男,硕士研究生。研究方向为嵌入式系统设计。

金伟明 男,副教授,硕士生导师。

(3) 此时上位机将提示安装驱动,装上厂商提供的驱动程序;

(4) 打开测试程序,选择管道号,设置缓冲区大小及发送次数,然后按下开始按钮即开始测试。测试DSP向PC机发送数据的截图如图5所示,由测试结果可以看出USB工作正常。

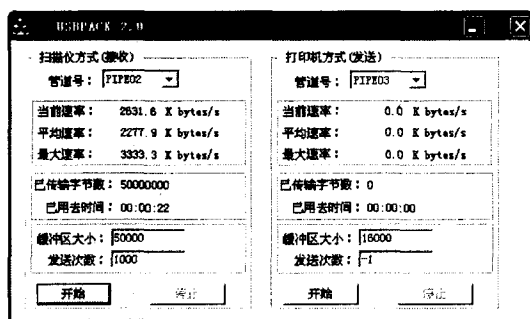


图5 测试截图

5 结语

数据采集系统往往要将数据传输到PC机上,而USB接口以其快速、灵活、方便的特性,成为数据采集系统与PC机间的数据传输的一种不错的选择。

参考文献

- [1] Universal Serial Bus Specification Revision 2.0[S]. 2000.
- [2] 萧世文. USB 2.0 硬件设计[M]. 北京:清华大学出版社,2002.
- [3] Griffiths H D. New Ideas in FM Radar Dept. of Electron. & Electr. Eng., Univ. Coll. London, UK, Electronics & Communication Engineering Journal, 1990; 185-194.
- [4] 费元春. 宽带雷达信号产生技术[M]. 北京:国防工业出版社,2002.
- [5] Venceslav F Kroupa, Vaclav Cizek, Jarmil Stursa, et al. Spurious Signals in Direct Digital Frequency Synthesizers[J]. IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control, 2000, 47(5): 1166-1172.
- [6] Abousetta M M, Cooper D C. Noise Analysis of Digitised FMCW Radar Waveforms[J]. Fac. of Electron. Eng., Univ. of Al-Tahadi, Bani-Walid, Libya, Radar, Sonar and Navigation, IEE Proceedings, 1998; 209-215.
- [7] High Stability Oscillator, Model SC10-10 MHz High Stability Ovenized Quartz Oscillator, Stanford Research Systems, Inc.
- [8] AD9854 Data Sheet, Analog Devices, Inc, 1999.
- [9] 丁鹭飞, 耿富录. 雷达原理[M]. 西安:西安电子科技大学出版社, 1995.
- [10] Merrill I, Skolnik. 雷达手册[M]. 王军, 译. 北京:电子工业出版社, 2003.

作者简介 潘胜 男,1973年出生,陕西西安人,硕士,工程师。主要研究方向为雷达系统设计。