

# 基于 MSP430 的模拟 SPI 串口通信的实现

张经爱<sup>1</sup>, 许凯华<sup>1</sup>, 刘玉华<sup>2</sup>

(1. 华中师范大学 物理科学与技术学院, 湖北 武汉 430079; 2. 华中师范大学 计算机科学系, 湖北 武汉 430079)

**摘 要:** SPI 接口是一种同步串行通讯接口。为了扩展系统接口, 有时必须要模拟出 SPI 的串口。提出了基于 TI 的 MSP430 系列单片机利用 I/O 软件模拟出了 SPI 串口的方法, 提高了 SPI 的接口效率, 简化了其使用方法, 并且给出连接 SPI 接口的 SST25VF020 NOR 型 FLASH 的实例。

**关键词:** 通信; SPI 接口; 模拟; 时序; I/O 函数

**中图分类号:** TP368.2 **文献标识码:** A **文章编号:** 1000-7024 (2008) 05-1169-03

## Realization of SPI interface simulated by MSP430

ZHANG Jing-ai<sup>1</sup>, XU Kai-hua<sup>1</sup>, LIU Yu-hua<sup>2</sup>

(1. College of Physical Science and Technology, Huazhong Normal University, Wuhan 430079, China;

2. Department of Computer Science, Huazhong Normal University, Wuhan 430079, China)

**Abstract:** SPI interface is a synchronously serial communication interface. It's necessary to simulate SPI interface in order to extend Interface. A realization of simulated is introduced by TI's MSP430. The method can increase the speed of the communication with SPI, and make the usage of SPI more convenient. The example of connecting SST25VF020 is provided with SPI interface based on MSP430F149 single chip microcomputer.

**Key words:** communication; SPI interface; simulation; sequence; I/O Function; MSP430

## 0 引 言

MSP430 单片机具有丰富的外围模块, 如 MSP430F149 就包含 7 组 I/O 端口, 精密模拟比较器, 硬件乘法器, 2 组频率可达 8MHz 的时钟模块, 2 个可以实现异步、同步及多址访问的 USART。在实际的应用中, USART 接口具有极大的通用性, 出于扩展接口的目的, 往往通过软件模拟 SPI, 以获得更多的 SPI 接口。串行外设接口 (serial peripheral interface, SPI) 是一种同步串行通讯接口, 用于微处理器、微控制器和外围扩展芯片之间的串行连接, 现已发展成为一种工业标准, 有着极为广泛的应用。笔者根据在实际工程中的应用, 以 MSP430F149 为例介绍模拟串口通信的方法。

## 1 MSP430 的 I/O 端口

MSP430 各个端口有大量的控制寄存器供用户使用, 其中 P1 和 P2 具有 7 个寄存器, P3、P4、P5、P6 具有 4 个寄存器。P1、P2 具有输入/输出, 中断和外部模块功能。每一个 I/O 端口有 8 位, 可以独立使用。用 Px 代表端口 P1、P2、..., P6, I/O 端口的寄存器如下所示:

(1) PxDIR 输入/输出方向寄存器

(2) PxIN 输入寄存器

(3) PxOUT 输出寄存器

(4) PxSEL 功能选择寄存器

(5) PxIES 中断触发沿选择寄存器, P1、P2 专有

(6) PxIFG 中断标志位寄存器, P1、P2 专有

(7) PxIE 中断使能寄存器, P1、P2 专有

MSP430 的管脚大多是复用的以减少引脚数量, PxSEL 选择 Px 是作为 I/O 还是作为其它的片内外设功能。I/O 口读入数据时, PxIN 就锁存数据, 访问该寄存器即可获得当前数据。I/O 写数据时, 先把数据存入 PxOUT 中, 硬件自动发送数据。由于 P1、P2 的每一位都可以产生独立的中断, 有独立的中断标志, 作为 I/O 端口功能太强, 往往用于其它用途, 同时出于通用性的考虑, 更倾向于使用普通的 I/O 口 (除 P1、P2 外的 I/O 端口)。

## 2 SPI 串口通信格式

串行外围设备 SPI 总线技术是一种同步串口接口, 具体的 3~4 线接口, 收发独立、可同步进行, 其硬件功能很强, 所以与 SPI 相关的软件相当简单, SPI 总线上可以连接多个可作为主机的 MCU (微控制器), 有 SPI 接口的输出输入设备, 但在任一瞬间只能允许一个设备作为主机, SPI 协议是以主从方式工

收稿日期: 2007-03-16 E-mail: jazzhang2@yahoo.com.cn

基金项目: 国家自然科学基金项目 (60673163)。

作者简介: 张经爱 (1981 -), 男, 湖北武汉人, 硕士研究生, 研究方向为嵌入式系统; 许凯华 (1952 -), 男, 湖北武汉人, 副教授, 研究方向为计算机网络与通信、GPS 导航; 刘玉华 (1951 -), 女, 湖北武汉人, 博士, 教授, 研究方向为计算机网络与通信、网络优化、复杂网络。

作的,这种模式通常有一个主设备和一个或多个从设备,其接口包括以下4种信号:

- (1)串行数据输入(也称为主进从出,或 MISO)
- (2)串行数据输出(也称为主出从进,或 MOSI)
- (3)串行移位时钟(也称为 SCK)
- (4)从使能信号(也称为 SS)

数据的传输由 SCK 决定,根据 SCK 时序的差别分为两种,数据在 SCK 的上升沿输出或者在 SCK 的下降沿输出。由于 SPI 接口定义的灵活性,同时也导致了各个厂商生产的具有 SPI 接口的芯片在工作时序上并不是一致的,所以使用时需要注意工作时序的差别。

### 3 基于 MSP430 模拟 SPI 串口的实现

本文采用 MSP430 的 I/O 端口来实现,选用 SCK 下降沿输出、上升沿输入的 SPI 接口类型,三线接口,定义 MSP430 的 I/O 端口 P4.4 为输出 MOSI, P5.0 是输入 MISO, P4.5 为 SCK 信号。在 P4.5SCK 制造一个下降沿,输出数据 data 的高位 BIT7,然后把 P4.5SCK 的电平拉高, data 中的数据依次右移一位,再将 P4.5 的电平拉低 SCK 输出低电平,又一次输出 data 的高位,这时,输出的就是原来 data 中的次高位 BIT6 位,如此重复, data 右移八次就完成了一个字符的输出。实现该功能的代码如下:

```
for(i=0;i<8;i++){ //BITx 是指 8 位字节的第
x 位为 1,其它位全为 0
    P4OUT &=~BIT5; //p4.5 输出 0, SCK 下降沿
    if(data & BIT7) //高位是 1 把 data 中的高位发出去
        P4OUT |= BIT4; //P4.4OUT 1;
    else
        P4OUT &=~BIT4; //P4.4 OUT 0
    P4OUT |=BIT5; //SCK 上升沿
    data<<=1; //data 中的数据右移
}
```

同样的原理,把 P4.5SCK 的输出电平拉低,制造一个下降沿,检测输入脚 P5.0 的电平,记入 data 中,把 P4.5SCK 的输出电平拉高,就完成了一个字节的接受,然后将 P4.5 的输出电平拉低,制造一个下降沿,再次检测输入脚 P5.0 的电平,记入 data 中,如此 8 次后就收到了一个字符,代码如下:

```
for(i=0;i<8;i++){
    P4OUT &=~BIT5; //sck 下降沿
    if(P5IN & BIT0) { //判断收到的电平信号
        data<<=1;
        data =data+BIT0;
    } else
        data<<=1;
    P4OUT |=BIT5; //sck 上升沿
}
```

如果是选用 SCK 下降沿输入、上升沿输出的 SPI 接口类型,只需要把上面循环中的 SCK 上升沿和 SCK 下降沿交换一下,就能满足新时序的要求。上述代码段代码精简,灵活性高。可以在具体的 SPI 读写程序中实现硬件接口的功能。其

串口速率主要由系统时钟来决定,1 个 SPI 的时钟周期就是执行上述模拟串口的 C 指令所需要的时间,提高系统时钟频率能够进一步的提高 SPI 串口的传输速率。

### 4 构造函数验证模拟 SPI 功能

SST25VF020 是一款 2Mbit SPI Serial NOR 型 Flash 芯片,读写操作时电流只有 7 mA,功耗低,读写速度快,擦除内部内容最大需要 70 ms,最大支持 20 MHz 的时钟频率,有专门的硬件写保护来保护数据,它通过 SPI 的串口接受指令和数据,支持 3/4 线的 SPI 协议,其工作时序如图 1 所示。

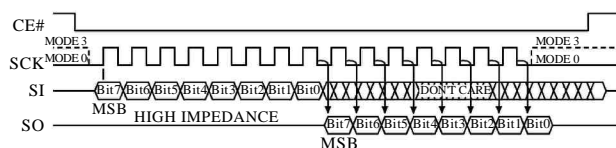


图 1 SST25VF020 的工作时序图

从图 1 可以看出,CE# 低电平使能芯片正常工作,该芯片在 SCK 的上升沿 SI 是主机读入信号,在 SCK 下降沿的时候 SO 主机输出信号。SST25VF020 芯片的内部有一个出厂 ID,位于 flash 的内部地址 0001H 处,读取这个 ID 号可以直接验证模拟程序的有效性。该芯片的其它功能请查阅相关的技术文档。现在选用 MSP430F149 来连接 flash 芯片,具体芯片引脚的连接如图 2 所示。

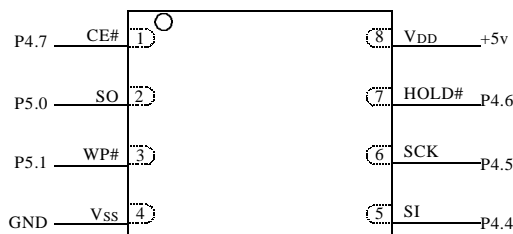


图 2 引脚连接

CE# 为 Chip Enable,低电平有效, SCK 为 Serial Clock, SI 为 Serial Data Input, SO 为 Serial Data Output, WP# 为 Write Protect, HOLD# Hold, VDD 为电源正(2.7v~3.6v), VSS 为地。首先需要定义 MSP430 的 I/O 端口, P4.4 为 MOSI, P4.5 为 SCK, P4.7 为 CE, P5.0 为 MISO。

由于是使用 MSP430 的 I/O 模拟,并未使用到中断函数,只需要一个子函数就能实现其功能。相比硬件实现而言,在操作上更加简单。定义读取 ID 的子函数如下:

```
void ReadID() { //read device's ID;
    int i;
    char command[5]={0x90,0x00,0x00,0x01};
    //0x90 是 sst25vf020 的读 ID 指令,后面 3 个为 000001H
    //是 ID 的地址
    P4OUT&=~BIT7; //CE 设为 0,
    SpiWrite(command,4); //给 flash 芯片发指令
    SpiRead(5); //读取 5 个字节
    P4OUT |=BIT7; //CE 拉高
}
```

在以上的函数中需要调用如下的两个子函数,分别实现 SPI 串口的读写功能,SpiWrite()函数的主要功能是向 flash 芯片发送指令,把需要发送的指令通过 SPI 模拟功传给 flash 芯片。其函数代码如下:

```
void SpiWrite(char *Buffer, int cnt){ //Buffer 是写入的字符串, cnt 是字符串的长度
    char data;
    int i,j;
    for(j=0;j<cnt;j++){
        data = Buffer[j];
        for(i=0;i<8;i++){
            P4OUT &=~BIT5; //SCK 下降沿
            if(data & BIT7) //高位是 1;
                P4OUT |= BIT4; //P4.2OUT 1;
            else
                P4OUT &=~BIT4; //P4.2 OUT 0
            P4OUT |=BIT5; //SCK 上升沿
            data<<=1;
        }
    }
}
```

在 SpiRead()函数中,通过 SPI 串口接收 flash 芯片发出的 ID 信息,该函数的主要功能就是通过软件的方式把 SPI 串口上的信息读取出来,放到 RXBuffer[]中。其函数代码如下:

```
void SpiRead(int cnt) { //cnt 是读入数据的个数,读入数据在 RXBuffer[]中
    int i,j;
    char data;
    for(j=0;j<cnt;j++){
        for(i=0;i<8;i++){
            P4OUT &=~BIT5; //SCK 的下降沿
            if(P5IN & BIT0) {
                data<<=1;
                data =data+BIT0;
            } else
                data<<=1;
        }
    }
}
```

```
P4OUT |=BIT5; // SCK 的上升沿
}
RXBuffer[j]=data;
}
P4OUT &=~BIT5; //down 0
}
```

在 IAR 开发环境下通过了测试,能够准确地读出 flash 芯片的内部 ID43H,也证实了该方法的可行性,开发者可以根据自己的实际需要,参考该芯片的手册,利用上述程序完成 flash 的擦除、写入和读取功能。

## 5 结束语

本文介绍了通过软件模拟 SPI 总线接口,实现与 SPI 接口的存储器件之间数据传输,给出了在 IAR 编译环境下编写的模拟 SPI 连接 SST25VF020 的子程序。经实测,能够很好的实现 SPI 接口的通信功能,不需要额外的中断函数来配合。功能函数的结构简单。可靠性高,该方法同样适用于 51 内核的单片机、ARM、微型计算机等。

## 参考文献:

- [1] 沈建华. MSP430 系列 FLASH 型超低功耗 16 位单片机原理与应用[M]. 北京:清华大学出版社,2004.
- [2] MSP430x1xx Family User's Guide[Z]. Texas Instruments Incorporated, 2002.
- [3] MSP430x14x Datasheets[DB/OL]. Texas Instruments Incorporated,2002.
- [4] MSP430 系列软件用户指南[Z]. 利尔达(中国)电子有限公司, 2001.
- [5] SST25VF020/SST25VF040 data sheet[S]. Silicon Storage Technology Inc,2006.
- [6] 胡大可. MSP430 系列单片机 C 语言程序设计与开发 [M]. 北京:北京航空航天大学出版社,2003.
- [7] 吴振纲,陈虎. PLC 的人机接口与编程[J].微计算机信息,2005,8 (1):21-23.
- [8] 郭玉辉,王彦瑜,乔卫民,等.基于 MSP430 单片机的远程束流诊断系统编程[J].计算机工程与设计,2005,26(1):94-96.

(上接第 1168 页)

## 参考文献:

- [1] 李庆诚. 翰林系列电子书的版权控制与支撑体系 [EB/OL]. <http://www.jinke.com.cn/>,2000.
- [2] 张珂.基于 UML 描述的本质安全 Linux 电子阅读器体系与实现研究[D].天津:南开大学信息技术科学学院,2005.
- [3] 陈章龙,唐志强,涂时亮.嵌入式技术与系统: Intel XScale.结构与开发[M].北京:北京航空航天大学出版社,2004.
- [4] 石文昌,孙玉芳.安全操作系统研究的发展(上)[J].计算机科学, 2002,29(6):5-12.
- [5] 石文昌,孙玉芳.安全操作系统研究的发展(下)[J].计算机科学, 2002,29(7):9-12.
- [6] 卿斯汉,刘文清,刘海峰.操作系统安全导论[M].北京:科学出版社,2003.
- [7] Bishop Brock, Karthick Rajamani. Dynamic power management for embedded systems[DB/OL]. Proceedings of the IEEE International SOC Conference,2003:17-20.
- [8] Karim Yaghmour. Building embedded Linux systems[M].北京: O'Reilly,2003.
- [9] 魏永明,耿岳,钟书毅. Linux 设备驱动程序[M].北京:中国电力出版社,2006.