**1.SPI协议简介**

**1.1.SPI协议概括**

　　SPI，是英语Serial Peripheral interface的缩写，顾名思义就是串行外围设备接口。是Motorola首先在其MC68HCXX系列处理器上定义的。SPI接口主要应用在 EEPROM，FLASH，实时时钟，AD转换器，还有数字信号处理器和数字信号解码器之间。SPI，是一种高速的，全双工，同步的通信总线，并且在芯片的管脚上只占用四根线，节约了芯片的管脚，同时为PCB的布局上节省空间，提供方便，正是出于这种简单易用的特性，现在越来越多的芯片集成了这种通信协议，比如AT91RM9200.

　　SPI的通信原理很简单，它以主从方式工作，这种模式通常有一个主设备和一个或多个从设备，需要至少4根线，事实上3根也可以（单向传输时）。也是所有基于SPI的设备共有的，它们是SDI（数据输入），SDO（数据输出），SCK（时钟），CS（片选）。

* SDO     – 主设备数据输出，从设备数据输入
* SDI      – 主设备数据输入，从设备数据输出
* SCLK   – 时钟信号，由主设备产生
* CS        – 从设备使能信号，由主设备控制

**CS**: 其中CS是控制芯片是否被选中的，也就是说只有片选信号为预先规定的使能信号时（高电位或低电位），对此芯片的操作才有效，这就允许在同一总线上连接多个SPI设备成为可能。

**SDI/SDO/SCLK**: 通讯是通过数据交换完成的，这里先要知道SPI是串行通讯协议，也就是说数据是一位一位的传输的。这就是SCK时钟线存在的原因，由SCK提供时钟脉冲，SDI，SDO则基于此脉冲完成数据传输。数据输出通过 SDO线，数据在时钟上升沿或下降沿时改变，在紧接着的下降沿或上升沿被读取。完成一位数据传输，输入也使用同样原理。这样，在至少8次时钟信号的改变（上沿和下沿为一次），就可以完成8位数据的传输。

　　要注意的是，SCK信号线只由主设备控制，从设备不能控制信号线。同样，在一个基于SPI的设备中，至少有一个主控设备。这样传输的特点：这样的传输方式有一个优点，与普通的串行通讯不同，普通的串行通讯一次连续传送至少8位数据，而SPI允许数据一位一位的传送，甚至允许暂停，因为SCK时钟线由主控设备控制，当没有时钟跳变时，从设备不采集或传送数据，也就是说，主设备通过对SCK时钟线的控制可以完成对通讯的控制。SPI还是一个数据交换协议：因为SPI的数据输入和输出线独立，所以允许同时完成数据的输入和输出。不同的SPI设备的实现方式不尽相同，主要是数据改变和采集的时间不同，在时钟信号上沿或下沿采集有不同定义，具体请参考相关器件的文档。

在点对点的通信中，SPI接口不需要进行寻址操作，且为全双工通信，显得简单高效。在多个从设备的系统中，每个从设备需要独立的使能信号，硬件上比I2C系统要稍微复杂一些。

最后，SPI接口的一个缺点：**没有指定的流控制，没有应答机制确认是否接收到数据**。

AT91RM9200的SPI接口主要由4个引脚构成：SPICLK、MOSI、MISO及 /SS，其中SPICLK是整个SPI总线的公用时钟，MOSI、MISO作为主机，从机的输入输出的标志，MOSI是主机的输出，从机的输入，MISO 是主机的输入，从机的输出。/SS是从机的标志管脚，在互相通信的两个SPI总线的器件，/SS管脚的电平低的是从机，相反/SS管脚的电平高的是主机。在一个SPI通信系统中，必须有主机。SPI总线可以配置成单主单从，单主多从，互为主从。

SPI的片选可以扩充选择16个外设,这时PCS输出=NPCS,说NPCS0~3接4-16译码器,这个译码器是需要外接4-16译码器，译码器的输入为NPCS0~3，输出用于16个外设的选择。

**1.2.协议举例**

SPI是一个环形总线结构，由ss（cs）、sck、sdi、sdo构成，其时序其实很简单，主要是在sck的控制下，两个双向移位寄存器进行数据交换。  
       假设下面的8位寄存器装的是待发送的数据10101010，上升沿发送、下降沿接收、高位先发送。  
       那么第一个上升沿来的时候 数据将会是sdo=1；寄存器=0101010x。下降沿到来的时候，sdi上的电平将所存到寄存器中去，那么这时寄存器=0101010sdi，这样在 8个时钟脉冲以后，两个寄存器的内容互相交换一次。这样就完成里一个spi时序。

举例：   
       假设主机和从机初始化就绪：并且主机的sbuff=0xaa，从机的sbuff=0x55，下面将分步对spi的8个时钟周期的数据情况演示一遍:假设上升沿发送数据

这样就完成了两个寄存器8位的交换，上面的上表示上升沿、下表示下降沿，sdi、sdo相对于主机而言的。其中ss引脚作为主机的时候，从机可以把它拉底被动选为从机，作为从机的是时候，可以作为片选脚用。根据以上分析，一个完整的传送周期是16位，即两个字节，因为，首先主机要发送命令过去，然后从机根据主机的命令准备数据，主机在下一个8位时钟周期才把数据读回来。

　　SPI 总线是Motorola公司推出的三线同步接口，同步串行3线方式进行通信:一条时钟线SCK，一条数据输入线MOSI，一条数据输出线MISO;用于CPU与各种外围器件进行全双工、同步串行通讯。SPI主要特点有:可以同时发出和接收串行数据;可以当作主机或从机工作;提供频率可编程时钟;发送结束 中断标志;写冲突保护;总线竞争保护等。下图示出SPI总线工作的四种方式，其中使用的最为广泛的是SPI0和SPI3方式 (实线表示):

                                                                                               SPI总线四种工作方式

SPI 模块为了和外设进行数据交换，根据外设工作要求，其输出串行同步时钟极性和相位可以进行配置，时钟极性（CPOL）对传输协议没有重大的影响。

* 如果 CPOL=0，串行同步时钟的空闲状态为低电平；
* 如果CPOL=1，串行同步时钟的空闲状态为高电平。时钟相位（CPHA）能够配置用于选择两种不同的传输协议之一进行数据传输。
* 如果CPHA=0，在串行同步时钟的第一个跳变沿（上升或下降）数据被采样；
* 如果CPHA=1，在串行同步时钟的第二个跳变沿（上升或下降）数据被采样。SPI主模块和与之通信的外设备时钟相位和极性应该一致。

**补充**：

上文中最后一句话：SPI主模块和与之通信的外设备时钟相位和极性应该一致。个人理解这句话有2层意思：其一，主设备SPI时钟和极性的配置应该由外设来决定；其二，二者的配置应该保持一致，即主设备的SDO同从设备的SDO配置一致，主设备的SDI同从设备的SDI配置一致。因为主从设备是在SCLK的控制下，同时发送和接收数据，并通过2个双向移位寄存器来交换数据。工作原理演示如下图：

上升沿主机SDO发送数据1，同时从设备SDO发送数据0；紧接着在SCLK的下降沿的时候从设备的SDI接收到了主机发送过来的数据1，同时主机也接收到了从设备发送过来的数据0.

**1.3.协议心得**

SPI接口时钟配置心得：

　　在主设备这边配置SPI接口时钟的时候一定要弄清楚从设备的时钟要求，因为主设备这边的时钟极性和相位都是以从设备为基准的。因此在时钟极性的配置上一定要搞清楚从设备是在时钟的上升沿还是下降沿接收数据，是在时钟的下降沿还是上升沿输出数据。但要注意的是，由于主设备的SDO连接从设备的SDI，从设备的SDO连接主设备的SDI，从设备SDI接收的数据是主设备的SDO发送过来的，主设备SDI接收的数据是从设备SDO发送过来的，所以主设备这边SPI时钟极性的配置（即SDO的配置）跟从设备的SDI接收数据的极性是相反的，跟从设备SDO发送数据的极性是相同的。下面这段话是Sychip Wlan8100 Module Spec上说的，充分说明了时钟极性是如何配置的：

The 81xx module will always input data bits at the rising edge of the clock, and the host will always output data bits on the falling edge of the clock.

意思是：主设备在时钟的下降沿发送数据，从设备在时钟的上升沿接收数据。因此主设备这边SPI时钟极性应该配置为下降沿有效。

又如，下面这段话是摘自LCD Driver IC SSD1289：

SDI is shifted into 8-bit shift register on every rising edge of SCK in the order of data bit 7, data bit 6 …… data bit 0.

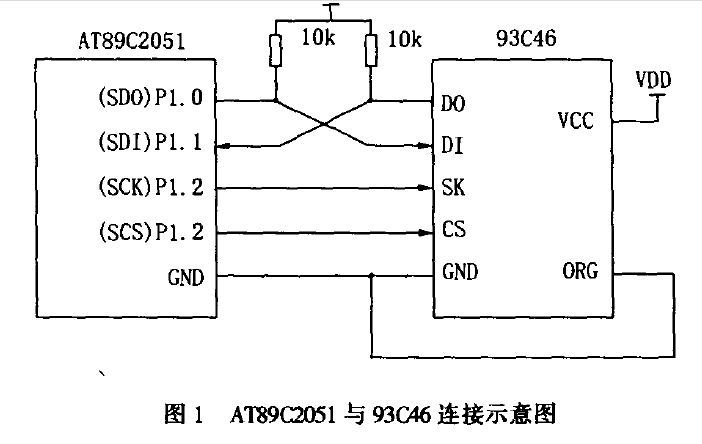
意思是：从设备SSD1289在时钟的上升沿接收数据，而且是按照从高位到地位的顺序接收数据的。因此主设备的SPI时钟极性同样应该配置为下降沿有效。

时钟极性和相位配置正确后，数据才能够被准确的发送和接收, 因此应该对照从设备的SPI接口时序或者Spec文档说明来正确配置主设备的时钟。

**2.SPI协议软件模拟**

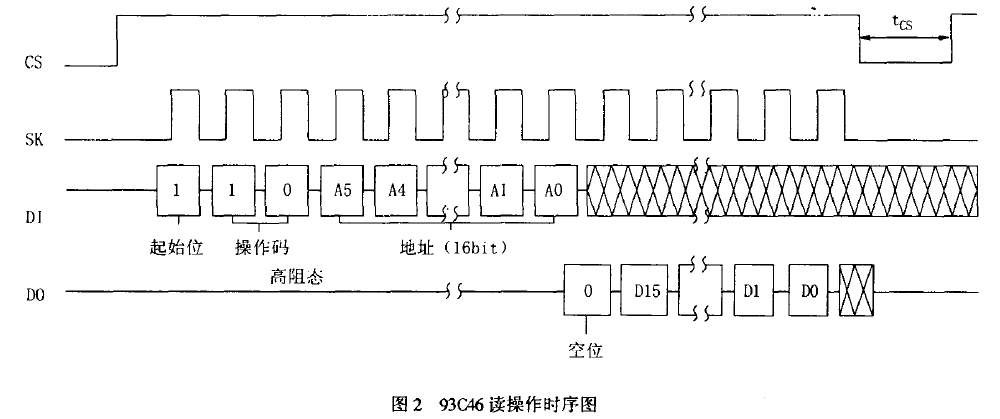
**2.1.单片机软件模拟SPI接口—加深理解SPI总线协议**

          SPI(Serial Peripheral Interfacer 串行外设接口)是摩托罗拉公司推出的一种同步串行通讯接口，用于微处理器臌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，目前，各半导体公司推出了大量的带有SPI接口的具有各种各样功能的芯片，如RAM，EEPROM，FlashROM，A／D转换器、D／A转换器、LED／LED显示驱动器、I／O接口芯片、实时时钟、UART收发器等等，为用户的外围扩展提供了极其灵活而价廉的选择。由于SPI总线接口只占用微处理器四个I／O口线，采用SPI总线接口可以简化电路没计，节省很多常规电路中的接口器件和I／O口线，提高设计的可靠性。  
  
        现以 AT89C205l单片机模拟SPI总线操作串行EEPROM 93CA6为例，如图1所示，介绍利用单片机的I／O口通过软件模拟SPI总线的实现方法。在这里，仅介绍读命令的时序和应用子程序。



**2.2.93C46存储器SPI总线的工作原理**

          93CA6作为从设备，其SPI接口使用4条I／O口线：串行时钟线(SK)、输出数据线DO、输入数据线DI和高电平有效的从机选择线CS。其数据的传输格式是高位(MSB)在前，低位(LsB)在后。93C46的SPI总线接口读命令时序如图2所示。



**2.3.软件模拟SPI接口的实现方法**

          对于不带SPI串行总线接口的AT89C2051单片 机来说，可以使用软件来模拟SPI的操作，图1所示 为AT89C2051单片机与串行EEPROM 93C46的硬件 连接图，其中，P1．0模拟SPI主设备的数据输出端 SDO，P1．2模拟SPI的时钟输出端SCK，P1．3模拟 SPI的从机选择端SCS，P1．1模拟SPI的数据输入 SDI。  
         上电复位后首先先将P1．2(SCK)的初始状态设置为0(空闲状态)。  
         读操作：AT89C2051首先通过P1．0口发送1位起始位(1)，2位操作码(10)，6位被读的数据地址(A5A4A3A2A1A0)，然后通过P1．1口读1位空位(0)，之后再读l6位数据(高位在前)。  
         写操作：AT89C2051首先通过P1．0口发送1位起始位(1)，2位操作码(01)，6位被写的数据地址(A5A4A3A2A1A0)，之后通过P1．0口发送被写的l6位数据(高位在前)，写操作之前要发送写允许命令，写之后要发送写禁止命令。  
         写允许操作(WEN))：写操作首先发送1位起始位(1)，2位操作码(00)，6位数据(11XXXX)。

         写禁止操作(WDS))：写操作首先发送1位起始位(1)，2位操作码(00)，6位数据(00XXXX)。

        下面介绍用C51模拟SPI的子程序。

[复制代码](javascript:void(0);)

1 //首先定义好I/O口

2 sbit SDO=P1^0；

3 sbit SDI=P1^1；

4 sbit SCK=P1^ 2；

5 sbit SCS=P1^3；

6 sbit ACC\_7= ACC^7；

7 unsigned int SpiRead(unsigned char add)

8 {

9 unsigned char i；

10 unsigned int datal6；

11 add&=0x3f；/\*6位地址\*/

12 add |=0x80；/\*读操作码l0\*/

13 SDO=1；/\*发送1为起始位\*/

14 SCK=0；

15 SCK=1；

16 for(i=0；<8；i++)/\*发送操作码和地址\*/

17 {

18 if(add&0x80==1)

19 SDO=1；

20 else

21 SDO=0；

22 SCK=0；/\*从设备上升沿接收数据\*/

23 SCK=1；

24 add<<= 1；

25 }

26 SCK=1；/\*从设备时钟线下降沿后发送数据，空读1位数据\*/

27 SCK=0；

28 datal6<<= 1;/\*读16位数据\*/

29 for(i=0；<16；i++)

30 {

31 SCK= 1；

32 \_nop\_()；

33 if(SDI==1)

34 datal6|=0x01；

35 SCK =0；

36 datal6< < =1；

37 }

38 return datal6；

39 }

[复制代码](javascript:void(0);)

　　对于不同的串行接口外围芯片，它们的时钟时序是不同的。上述子程序是针对在SCK的上升沿输入(接收)数据和在下降沿输出(发送)数据的器件。这些子程序也适用于在串行时钟)的上升沿输入和下降沿输出的其它各种串行外围接口芯片，只要在程序中改变P1．2(SCK)的输出电平顺序进行相应调整即可。