



STM32 MCU Development

# STM32 单片机开发

-- SPI 协议介绍

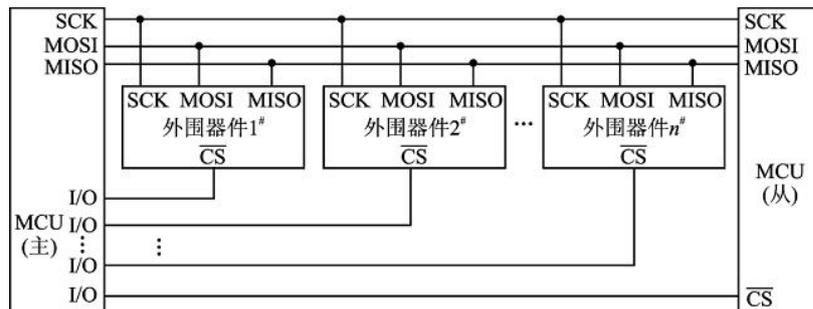
## 目录

1. SPI 协议介绍.....	3
1.1. SPI 协议简介.....	3
1.2. SPI 总线传输模式.....	3
1.3. SPI 总线数据交换.....	5

# 1. SPI 协议介绍

## 1.1. SPI 协议简介

SPI(Serial Peripheral Interface)是 Motorola 公司于 1979 年推出的一种高速全双工同步串行通信协议，它由一个主设备(Master)和一个或多个从设备(Slave)组成，其中主设备启动与从设备的同步通信，从而完成数据的交换。



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，一般需要 4 根线，事实上 3 根也可以。这四根线分别是 MISO(Master Input Slave Output)/SDI(Serial Data Input)、MOSI(Master Output Slave Input)/SDO(Serial Data Output)、SCLK(Synchronous Clock)、CS(Chip Select)。

- MISO，主设备数据输入，从设备数据输出；
- MOSI，主设备数据输出，从设备数据输入；
- SCLK，同步时钟信号，由主设备产生；
- CS，从设备使能信号，由主设备控制。当总线上有多个从设备的时候，主设备如果需要和某个从设备通信，就将该设备对应的片选引脚拉低使能；

SPI 在芯片上只占用四个管脚来实现数据传输及控制，这样为 PCB 设计在布局上节省了空间，正是出于这种简单易用的特性，现在越来越多的芯片上都集成了 SPI 接口，主要应用在 Flash、通信模块(NFC、以太网、LoRa)、数模转换器(ADC)等通信领域。

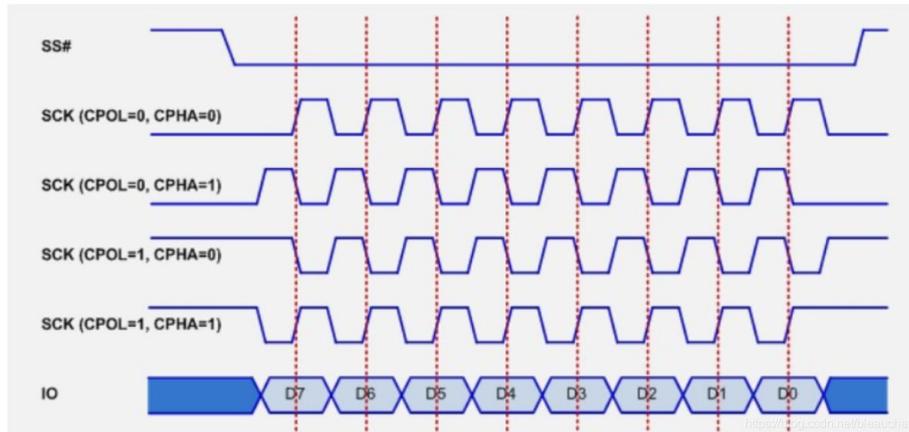
由于 SPI 采用两根数据线实现全双工的数据通信，从而能达到比 I2C 更快的通信速率。I2C 协议 v2.1 规定了 100K，400K 和 3.4M 三种速率(bps)。而 SPI 是一种事实标准，并没有一个官方标准，已知已有的器件 SPI 速率可达到 50Mbps，具体到产品中 SPI 的速率主要看主从器件 SPI 控制器的性能限制。此外，SPI 没有相应的流控和应答机制，这样跟 IIC 协议相比在数据可靠性上有一定的缺陷。

## 1.2. SPI 总线传输模式

Master 设备会根据将要交换的数据来产生相应的时钟脉冲(Clock Pulse)，时钟脉冲组成了时钟信号(Clock Signal)，时钟信号通过时钟极性 (CPOL, Clock Polarity) 和时钟相位 (CPHA, Clock Phase) 控制着两个 SPI 设备间何时数据交换以及何时对接收到的数据进行采样，来保证数据在两个设备之间是同步传输的。

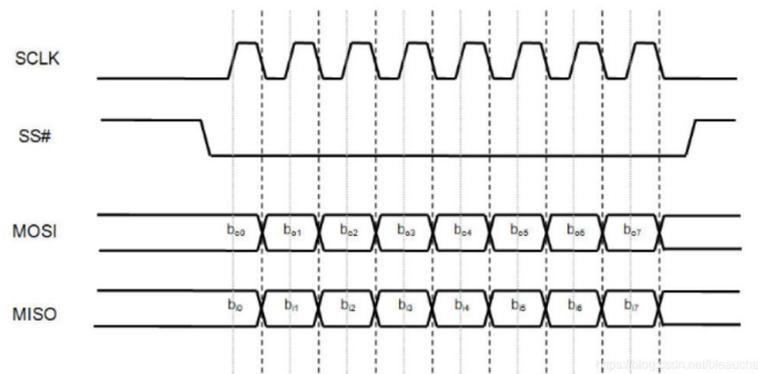
- CPOL(Clock Polarity, 时钟的极性): 规约 SPI 总线在空闲时, 时钟信号是高电平还是低电平;
- CPHA(Clock Phase, 时钟的相位): 规约 SPI 设备是在时钟信号上升沿还是下降沿触发数据采样;

根据时钟极性(CPOL)和时钟相位(CPHA)的不同配置, SPI 总线传输一共有 4 中模式。这四种模式的时序如下图所示:



- 模式 0: CPOL= 0, CPHA=0。SCK 串行时钟线空闲是为低电平, 数据在 SCK 时钟的上升沿被采样, 数据在 SCK 时钟的下降沿切换;
- 模式 1: CPOL= 0, CPHA=1。SCK 串行时钟线空闲是为低电平, 数据在 SCK 时钟的下降沿被采样, 数据在 SCK 时钟的上升沿切换;
- 模式 2: CPOL= 1, CPHA=0。SCK 串行时钟线空闲是为高电平, 数据在 SCK 时钟的下降沿被采样, 数据在 SCK 时钟的上升沿切换;
- 模式 3: CPOL= 1, CPHA=1。SCK 串行时钟线空闲是为高电平, 数据在 SCK 时钟的上升沿被采样, 数据在 SCK 时钟的下降沿切换;

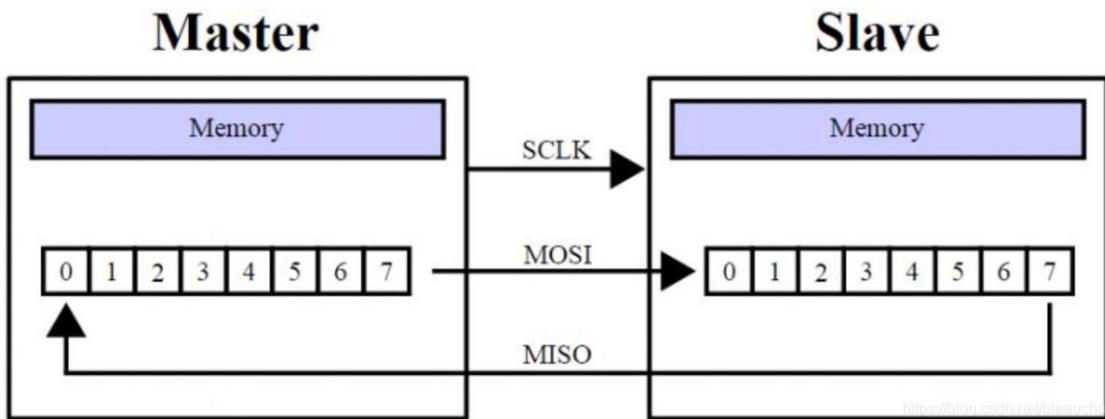
其中比较常用的模式是模式 0 和模式 3, 为了更清晰的描述 SPI 总线的时序, 下图展现了模式 0 下的 SPI 时序图。从途中我们可以清晰地看出, 在模式 0 下总线空闲时, SCK 串行时钟线为低电平, 当 SS 被主机拉低以后, 数据传输开始, 数据线 MOSI 和 MISO 的数据切换(Toggling)发生在时钟的下降沿(下图的黑色虚线), 而数据线 MOSI 和 MISO 的数据采样(Sampling)发生在数据的正中间(下图中的灰色实线)。



### 1.3. SPI 总线数据交换

一个 Slave 设备要想能够接收到 Master 发过来的控制信号，必须在此之前能够被 Master 设备进行访问 (Access)。所以，Master 设备必须首先通过拉低 SS/CS 管脚对 Slave 设备进行片选，把想要访问的 Slave 设备选上。

SPI 设备间的数据传输之所以又被称为数据交换，是因为 SPI 协议规定一个 SPI 设备不能在数据通信过程中仅仅只充当一个 "发送者(Transmitter)" 或者 "接收者(Receiver)"。事实上 SPI 设备在通信时两边各有 1 个移位寄存器，这样在发送一个 bit 的同时将会收到对方 1 个 bit 的数据，当发送完 8bit 数据后，也将收到对方的 8bit 数据，这也就意味着完成了 1 个 byte 的数据交换。



SPI 数据传输过程实际上就是主从设备两个移位寄存器的数据交换过程，所以 SPI 的读功能实际上可以由 SPI 的写功能实现，即随便写一个字节的无效数据(Dummy Data)即可。在数据传输的过程中，每次接收到的数据必须在下一次数据传输之前被采样。如果之前接收到的数据没有被读取，那么这些已经接收完成的数据将有可能被丢弃，导致 SPI 物理模块最终失效。因此，在程序中一般都会在 SPI 传输完数据后，去读取 SPI 设备里的数据，即使这些数据(Dummy Data)在我们的程序里是无用的。