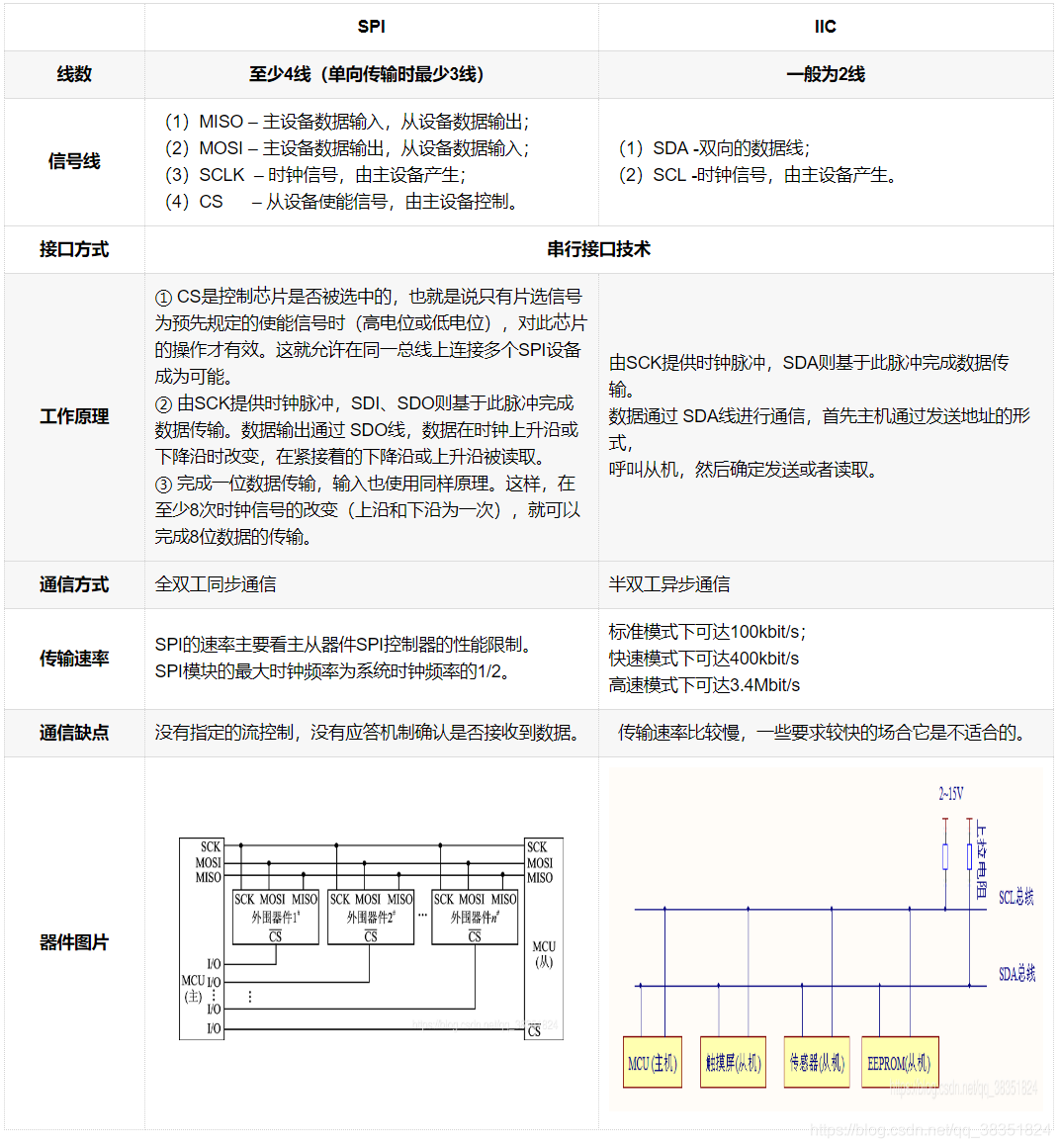
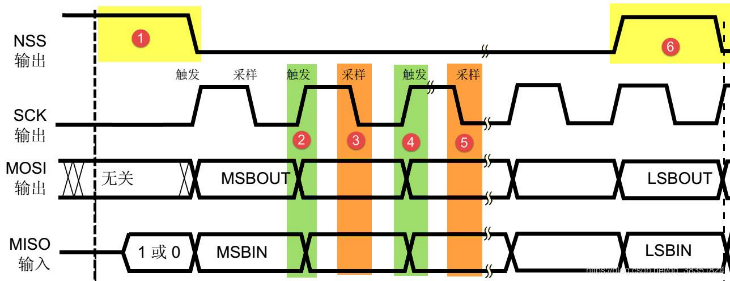
# SPI与IIC对比

1. SPI 与 IIC 通信协议比对：



1. SPI 通信协议：



（1）引脚简介

 NSS、 SCK、 MOSI 信号都由主机控制产生， MISO 的信号由从机产生，主机通过该信号线读取从机的数据。

 MOSI 与 MISO 的信号只在 NSS 为低电平的时候才有效，在 SCK 的每个时钟周期 MOSI 和 MISO 传输一位数据。

（2）起始信号：标号 ①；停止信号：标号 ⑥

标号①处， NSS 信号线由高变低，是 SPI 通讯的起始信号。

标号⑥处， NSS 信号线由低变高，是 SPI 通讯的停止信号，表示本次通讯结束，从机的选中状态被取消。

（3）数据的有效性：② ③ ④ ⑤

SPI 使用 MOSI 及 MISO 信号线来传输数据，使用 SCK 信号线进行数据同步。 MOSI及 MISO 数据线在 SCK 的每个时钟周期传输一位数据，且数据输入输出是同时进行的。数据传输时， MSB 先行或 LSB 先行并没有作硬性规定，但要保证两个 SPI 通讯设备之间使用同样的协定，一般都会采用图中的 MSB 先行模式。

观察图中的② ③ ④ ⑤标号处， MOSI 及 MISO 的数据在 SCK 的上升沿期间变化输出，在 SCK 的下降沿时被采样。

SPI 每次数据传输可以 8 位或 16 位为单位，每次传输的单位数不受限制。

（4）CPOL/CPHA 及通讯模式

上图中的时序只是 SPI 中的其中一种通讯模式，SPI 一共有四种通讯模式，它们的主要区别是总线空闲时 SCK 的时钟状态以及数据采样时刻。主机与从机需要工作在相同的模式下才可以正常通讯，实际中采用较多的是“模式 0”与“模式 3”



CPOL=0，表示当SCLK=0时处于空闲态，所以有效状态就是SCLK处于高电平时

CPOL=1，表示当SCLK=1时处于空闲态，所以有效状态就是SCLK处于低电平时

CPHA=0，表示数据采样是在第1个边沿，数据发送在第2个边沿

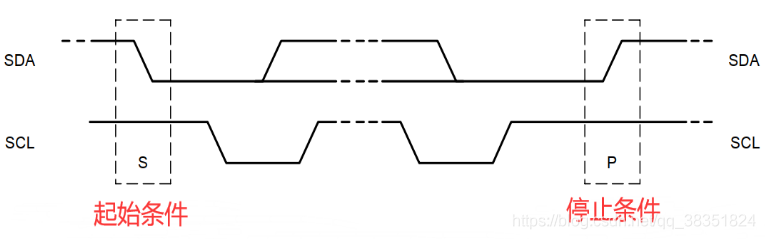
CPHA=1，表示数据采样是在第2个边沿，数据发送在第1个边沿

三、IIC 通信协议：

（1）起始条件、停止条件

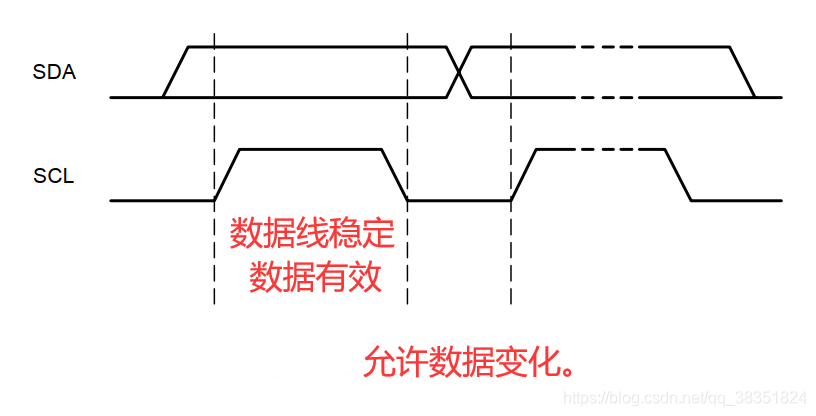
SCL线是高电平时 SDA 线从高电平向低电平切换表示起始条件。

SCL线是高电平时 SDA 线从低电平向高电平切换表示停止条件。

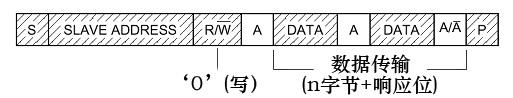


（2）数据的有效性

SDA 线上的数据必须在时钟SCL线的高电平周期保持稳定 ；数据线的高或低电平状态只有在 SCL 线的时钟信号是低电平时才能改变。



（3）主机写数据到从机



S：起始条件

SLAVE ADDRESS：从机地址（从机地址可以是 7 位或 10 位）。

R/￣W：W代表写，R代表读，因为W上面有上划线，所以该位位低电平（即0）时，为写操作。

A：从机发来应答(ACK)信号（有了应答信号才能接着进行下一步）。

DATA：要发送的数据（数据包的大小为 8 位）。

A/ ￣A：从机发来应答(ACK)或非应答(NACK)信号（有了应答信号才能接着进行下一步）。

P：停止条件

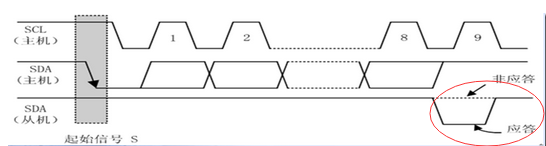
（4）应答信号(ACK)与非应答信号(NACK)

当设备(无论主从机)接收到 I2C 传输的一个字节数据或地址后，若希望对方继续发送数据，则需要向对方发送“应答(ACK)”信号，发送方会继续发送下一个数据；

若接收端希望结束数据传输，则向对方发送“非应答(NACK)”信号，发送方接收到该信号后会产生一个停止信号，结束信号传输。

应答信号：主机发送完一个8位数据后，会等待从机的回答一个ACK信号就是SDA将会拉低。

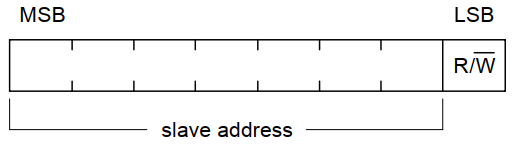
非应答信号：主机发送完一个8位数据后，会等待从机的回答一个NACK信号就是SDA将会拉高。



（5）字节格式

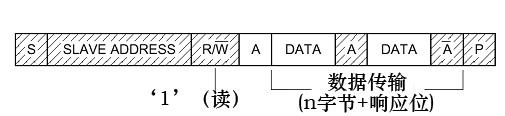
发送到 SDA 线上的每个字节必须为 8 位，每次传输可以发送的字节数量不受限制，每个字节后必须跟一个响应位。

（6）发送从机地址（SLAVE ADDRESS）的格式



从机地址（从机地址可以是 7 位或 10 位）一般是7位，再加上一位读/写位。具体见下面。

（7）主机由从机中读数据



S：起始条件

SLAVE ADDRESS：从机地址（从机地址可以是 7 位或 10 位）。

R/￣W：W代表写，R代表读，因为W上面有上划线，所以该位位低电平（即0）时，为写操作。

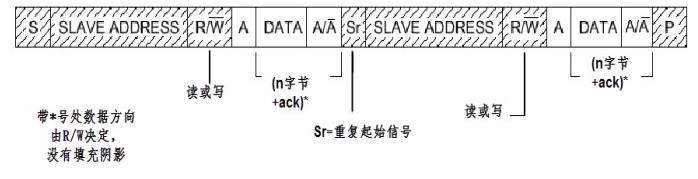
A：主机发送应答(ACK)信号（有了应答信号才能接着进行下一步）。

DATA：要接收的数据（数据包的大小为 8 位）。

A/ ￣A：主机发送应答(ACK)或非应答(NACK)信号（有了应答信号才能接着进行下一步）。

P：停止条件

（8）通讯复合格式



S：起始条件

SR：重复起始信号

SLAVE ADDRESS：从机地址（从机地址可以是 7 位或 10 位）。

R/￣W：W代表写，R代表读，因为W上面有上划线，所以该位位低电平（即0）时，为写操作。

A：应答(ACK)信号（有了应答信号才能接着进行下一步）。

DATA：要传输的数据（数据包的大小为 8 位）。

A/ ￣A：应答(ACK)或非应答(NACK)信号（有了应答信号才能接着进行下一步）。

P：停止条件

# 几种常用通信协议：IIC协议、SPI协议、UART协议

通信可以形象的比喻成两个人讲话：1、你说的别人得能听懂：双方约定信号的协议。2、你的语速别人得能接受：双方满足时序要求。

一、IIC协议：

2C串行总线一般有两根信号线，一根是双向的数据线SDA，另一根是时钟线SCL。所有接到I2C总线设备上的串行数据SDA都接到总线的SDA上，各设备的时钟线SCL接到总线的SCL上。两条线可以挂多个设备。IIC设备（稍微有点智能的）里有个固化的地址。只有在两条线上传输的值等于我（IIC设备）的地址时，我才作出响应。

开始信号：处理器让SCL时钟保持高电平，然后让SDA数据信号由高变低就表示一个开始信号。同时IIC总线上的设备检测到这个开始信号它就知道处理器要发送数据了。

停止信号：处理器让SCL时钟保持高电平，然后让SDA数据信号由低变高就表示一个停止信号。同时IIC总线上的设备检测到这个停止信号它就知道处理器已经结束了数据传输，我们就可以各忙各个的了，如休眠等。

再看数据怎么传：SDA上传输的数据必须在SCL为高电平期间保持稳定：因为外接IIC设备在SCL为高电平的期间采集数据方知SDA是高或低电平。SDA上的数据只能在SCL为低电平期间翻转变化。

响应信号（ACK）：处理器把数据发给外接IIC设备，如何知道IIC设备数据已经收到呢？就需要外接IIC设备回应一个信号给处理器。处理器发完8bit数据后就不再驱动总线了（SDA引脚变输入），而SDA和SDL硬件设计时都有上拉电阻，所以这时候SDA变成高电平。那么在第8个数据位，如果外接IIC设备能收到信号的话接着在第9个周期把SDA拉低，那么处理器检测到SDA拉低就能知道外接IIC设备数据已经收到。

IIC数据从最高位开始传输。IIC总线是允许挂载多个设备的，如何访问其中一个设备而不影响其他设备呢？

主机向从机发送7位地址寻址从IIC设备，用7bit表示从地址，那么可以挂载的从设备数是2的7次方128个。处理器想写的话：先发送起始位，再发一个8bit数据：前7bit表示从地址，第8bit表示读或者写。0write是处理器往IIC从设备发，1read是IIC从设备往处理器发。第9个时钟周期回复响应信号。例如写过程：

首先发出一个start信号，从设备地址，R/W（0，写），回应ACK表示有这个从设备存在。这时候是处理器从指定的从设备读数据的从设备里8bit存储地址的指定。所以这里R/W是0为写。ACK回应有这个设备的话，处理器把要访问的从设备里的8bit存储地址写好。ACK对方回应。继续一个start信号+从设备地址，最低位是高电平表示读数据（等待ACK回应），回应ACK表示有这个从设备存在。在读数据的时候，每发出一个时钟，处理器会SDA上的数据存起来。那么发出8个时钟后处理器就能得到8位的数据。这时候若想连续读就不断回应ACK信号否则就发出停止信号。

读的过程：start信号，从设备地址，写，待读取存储地址，再一个start信号，从设备地址，读，8个时钟，从设备就把对应的数据反馈给处理器。

二、SPI协议：

SPI：高速同步串行口，是一种标准的四线同步全双工串行总线。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（用于单向传输时，也就是半双工方式）。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），SCLK（时钟），CS（片选）。 SPI 主要应用在 EEPROM, Flash, 实时时钟(RTC), 数模转换器(ADC), 数字信号处理器(DSP) 以及数字信号解码器之间. 它在芯片中只占用四根管脚 (Pin) 用来控制以及数据传输, 节约了芯片的 pin 数目, 同时为 PCB 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技术.SPI只有主模式和从模式之分，没有读和写的说法，因为实质上每次SPI是主从设备在交换数据。也就是说，你发一个数据必然会收到一个数据；你要收一个数据必须也要先发一个数据。

工作过程：

上升沿发送、下降沿接收、高位先发送；

上升沿到来的时候，SDO上的电平将被发送到从设备的寄存器中；

下降沿到来的时候，SDI上的电平将被接收到主设备的寄存器中。

一个完整的传送周期是16位，即两个字节。因为主机首先要发送命令给从机，然后从机根据主机的指令准备数据，所以主机在下一个8位时钟周期才把数据读回来。SPI总线有四种工作方式(SP0, SP1, SP2, SP3)，其中使用的最为广泛的是SPI0和SPI3方式。SPI接口有四种不同的数据传输时序，取决于CPOL和CPHL这两位的组合，CPOL是用来决定SCK时钟信号空闲时的电平，CPOL＝0，空闲电平为低电平，CPOL＝1时，空闲电平为高电平。CPHA是用来决定采样时刻的，CPHA=0，在每个周期的第一个时钟沿采样，CPHA＝1，在每个周期的第二个时钟沿采样。

三、UART协议：

UART:Universal Asynchronous Receiver/Transmitter，通用异步接收/发送装置，UART是一个并行输入成为串行输出的芯片，通常集成在主板上，多数是16550AFN芯片。因为计算机内部采用并行数据，不能直接把数据发到Modem，必须经过UART整理才能进行异步传输，其过程为：CPU先把准备写入串行设备的数据放到UART的寄存器（临时内存块）中，再通过FIFO（First Input First Output，先入先出队列）传送到串行设备，若是没有FIFO，信息将变得杂乱无章，不可能传送到Modem中。它是用于控制计算机与串行设备的芯片。有一点要注意的是，它提供了RS-232C数据终端设备接口，这样计算机就可以和调制解调器或其它使用RS-232C接口的串行设备通信了。作为接口的一部分，UART还提供以下功能：将由计算机内部传送过来的并行数据转换为输出的串行数据流。将计算机外部来的串行数据转换为字节，供计算机内部使用并行数据的器件使用。它包括了RS232、RS499、RS423、RS422和RS485等接口标准规范和总线标准规范，即UART是异步串行通信口的总称。

通信过程：

UART首先将接收到的并行数据转换成串行数据来传输。消息帧从一个低位起始位开始，后面是5~8个数据位，一个可用的奇偶位和一个或几个高位停止位。接收器发现开始位时它就知道数据准备发送，并尝试与发送器时钟频率同步。如果选择了奇偶，UART就在数据位后面加上奇偶位。奇偶位可用来帮助错误校验。在接收过程中，UART从消息帧中去掉起始位和结束位，对进来的字节进行奇偶校验，并将数据字节从串行转换成并行。UART也产生额外的信号来指示发送和接收的状态。例如，如果产生一个奇偶错误，UART就置位奇偶标志。

数据发送的思想是，当启动字节发送时，通过TxD先发起始位，然后发数据位和奇偶数效验位，最后再发停止位，发送过程由发送状态机控制，每次中断只发送1个位，经过若干个定时中断完成1个字节帧的发送。

数据接收的思想是，当不在字节帧接收过程时，每次定时中断以3倍的波特率监视RxD的状态，当其连续3次采样电平依次为1、0、0时，就认为检测到了起始位，则开始启动一次字节帧接收，字节帧接收过程由接收状态机控制，每次中断只接收1个位，经过若干个定时中断完成1个字节帧的接收。。UART典型为38400，9600波特 。如下图1，UART字符格式为1个起始位，5~8个数据位，1个地址位或奇偶位(可选)，1个停止位。由于接收器、发送器异步工作，无需联接接收和发送时钟。接收器采取对输入数据流高度采样方式，通常采样为16，并根据采样值确定位值。按惯例，使用16个采样值的中间三个值。

四、三种协议的区别

第一个区别当然是名字：

SPI(Serial Peripheral Interface：串行外设接口);

I2C(INTER IC BUS：意为IC之间总线)

UART(Universal Asynchronous Receiver Transmitter：通用异步收发器)

第二，区别在电气信号线上：

SPI 总线由三条信号线组成：串行时钟(SCLK)、串行数据输出(SDO)、串行数据输入(SDI)。SPI总线可以实现 多个SPI设备互相连接。提供 SPI串行时钟的SPI设备为SPI主机或主设备(Master)，其他设备为SPI从机或从设备(Slave)。主从设备间可以实现全双工通信，当有多个从设备时，还可以增加一条从设备选择线。

如果用通用IO口模拟SPI总线，必须要有一个输出口(SDO)，一个输入口(SDI)，另一个口则视实现的设备类型而定，如果要实现主从设备，则需输入输出口，若只实现主设备，则需输出口即可，若只实现从设备，则只需输入口即可。

I2C总线是双向、两线(SCL、SDA)、串行、多主控（multi-master）接口标准，具有总线仲裁机制，非常适合在器件之间进行近距离、非经常性的数据通信。在它的协议体系中，传输数据时都会带上目的设备的设备地址，因此可以实现设备组网。

如果用通用IO口模拟I2C总线，并实现双向传输，则需一个输入输出口(SDA)，另外还需一个输出口(SCL)。（注：I2C资料了解得比较少，这里的描述可能很不完备）

UART总线是异步串口，因此一般比前两种同步串口的结构要复杂很多，一般由波特率产生器(产生的波特率等于传输波特率的16倍)、UART接收器、UART发送器组成，硬件上由两根线，一根用于发送，一根用于接收。

显然，如果用通用IO口模拟UART总线，则需一个输入口，一个输出口。

第三，从第二点明显可以看出，SPI和UART可以实现全双工，但I2C不行；

第四，看看牛人们的意见吧！

wudanyu：I2C 线更少，我觉得比UART、SPI更为强大，但是技术上也更加麻烦些，因为I2C需要有双向IO的支持，而且使用上拉电阻，我觉得抗干扰能力较弱，一般用于同一板卡上芯片之间的通信，较少用于远距离通信。SPI实现要简单一些，UART需要固定的波特率，就是说两位数据的间隔要相等，而SPI则无所谓，因为它是有时钟的协议。

quickmouse：I2C的速度比SPI慢一点，协议比SPI复杂一点，但是连线也比标准的SPI要少。