**[i2c协议简要分析（转载）](http://www.cnblogs.com/jontian/p/5717324.html)**

**I2C协议**

I2C是由Philips公司发明的一种串行数据通信协议，仅使用两根信号线：SerialClock（简称SCL）和SerialData（简称SDA）。I2C是总线结构，1个Master，1个或多个Slave，各Slave设备以7位地址区分，地址后面再跟1位读写位，表示读（=1）或者写（=0），所以我们有时也可看到8位形式的设备地址，此时每个设备有读、写两个地址，高7位地址其实是相同的。

数据格式

I2C数据格式如下：  
无数据：SCL=1，SDA=1；  
开始位（Start）：当SCL=1时，SDA由1向0跳变；  
停止位（Stop）：当SCL=1时，SDA由0向1跳变；  
数据位：当SCL由0向1跳变时，由发送方控制SDA，此时SDA为有效数据，不可随意改变SDA；  
当SCL保持为0时，SDA上的数据可随意改变；  
地址位：定义同数据位，但只由Master发给Slave；  
应答位（ACK）：当发送方传送完8位时，发送方释放SDA，由接收方控制SDA，且SDA=0；  
否应答位（NACK）：当发送方传送完8位时，发送方释放SDA，由接收方控制SDA，且SDA=1。

传输过程

当数据为单字节传送时，格式为：  
开始位，8位地址位（含1位读写位），应答，8位数据，应答，停止位。  
当数据为一串字节传送时，格式为：  
开始位，8位地址位（含1位读写位），应答，8位数据，应答，8位数据，应答，……，8位数据，应答，停止位。

注意事项

SCL一直由Master控制，SDA依照数据传送的方向，读数据时由Slave控制SDA，写数据时由Master控制SDA。当8位数据传送完毕之后，应答位或者否应答位的SDA控制权与数据位传送时相反。

开始位“Start”和停止位“Stop”，只能由Master来发出。

地址的8位传送完毕后，成功配置地址的Slave设备必须发送“ACK”。否则否则一定时间之后Master视为超时，将放弃数据传送，发送“Stop”。

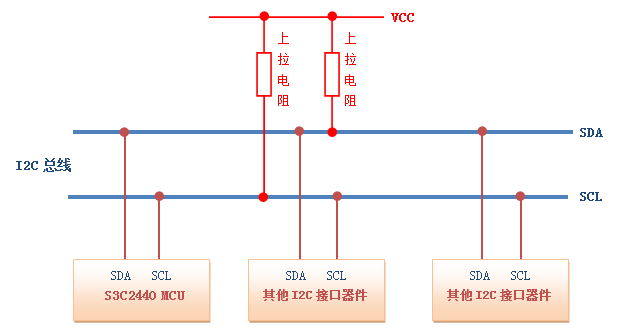
当写数据的时候，Master每发送完8个数据位，Slave设备如果还有空间接受下一个字节应该回答“ACK”，Slave设备如果没有空间接受更多的字节应该回答“NACK”，Master当收到“NACK”或者一定时间之后没收到任何数据将视为超时，此时Master放弃数据传送，发送“Stop”。

当读数据的时候，Slave设备每发送完8个数据位，如果Master希望继续读下一个字节，Master应该回答“ACK”以提示Slave准备下一个数据，如果Master不希望读取更多字节，Master应该回答“NACK”以提示Slave设备准备接收Stop信号。

当Master速度过快Slave端来不及处理时，Slave设备可以拉低SCL不放（SCL=0将发生“线与”）以阻止Master发送更多的数据。此时Master将视情况减慢或结束数据传送。

**1.1 I2C总线知识**

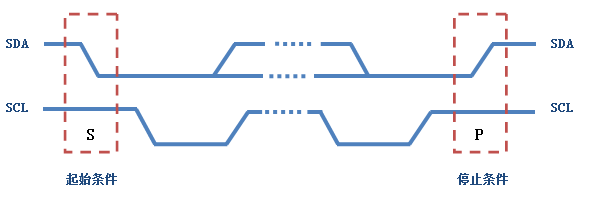
**1.1.1  I2C总线物理拓扑结构**



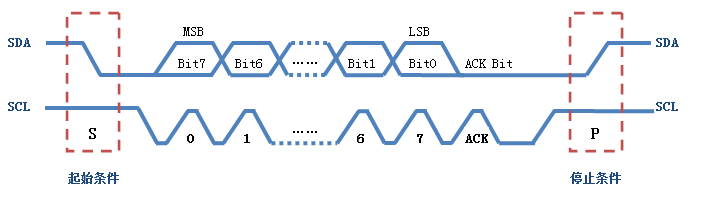
 I2C 总线在物理连接上非常简单，分别由SDA(串行数据线)和SCL(串行时钟线)及上拉电阻组成。通信原理是通过对SCL和SDA线高低电平时序的控制，来 产生I2C总线协议所需要的信号进行数据的传递。在总线空闲状态时，这两根线一般被上面所接的上拉电阻拉高，保持着高电平。

**1.1.2  I2C总线特征**  
    I2C总线上的每一个设备都可以作为主设备或者从设备，而且每一个设备都会对应一个唯一的地址(可以从I2C器件的数据手册得知)，主从设备之间就通过这 个地址来确定与哪个器件进行通信，在通常的应用中，我们把CPU带I2C总线接口的模块作为主设备，把挂接在总线上的其他设备都作为从设备。  
    I2C总线上可挂接的设备数量受总线的最大电容400pF 限制，如果所挂接的是相同型号的器件，则还受器件地址位的限制。  
    I2C总线数据传输速率在标准模式下可达100kbit/s，快速模式下可达400kbit/s，高速模式下可达3.4Mbit/s。一般通过I2C总线接口可编程时钟来实现传输速率的调整，同时也跟所接的上拉电阻的阻值有关。  
    I2C总线上的主设备与从设备之间以字节(8位)为单位进行双向的数据传输。

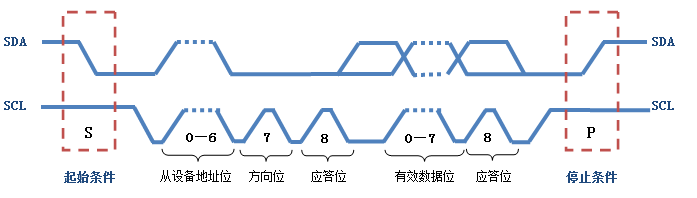
**1.1.3  I2C总线协议**  
    I2C协议规定，总线上数据的传输必须以一个起始信号作为开始条件，以一个结束信号作为传输的停止条件。起始和结束信号总是由主设备产生。总线在空闲状态 时，SCL和SDA都保持着高电平，当SCL为高电平而SDA由高到低的跳变，表示产生一个起始条件；当SCL为高而SDA由低到高的跳变，表示产生一个 停止条件。在起始条件产生后，总线处于忙状态，由本次数据传输的主从设备独占，其他I2C器件无法访问总线；而在停止条件产生后，本次数据传输的主从设备 将释放总线，总线再次处于空闲状态。如图所示：



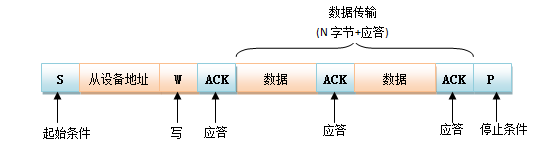
 在了解起始条件和停止条件后，我们再来看看在这个过程中数据的传输是如何进行的。前面我们已经提到过，数据传输以字节为单位。主设备在SCL线上产生每个 时钟脉冲的过程中将在SDA线上传输一个数据位，当一个字节按数据位从高位到低位的顺序传输完后，紧接着从设备将拉低SDA线，回传给主设备一个应答位， 此时才认为一个字节真正的被传输完成。当然，并不是所有的字节传输都必须有一个应答位，比如：当从设备不能再接收主设备发送的数据时，从设备将回传一个否 定应答位。数据传输的过程如图所示：



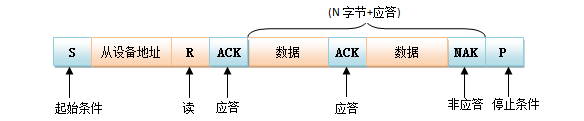
在前面我们还提到过，I2C总线上的每一个设备都对应一个唯一的地址，主从设备之间的数据传输是建立在地址的基础上，也就是说，主设备在传输有效数据之前 要先指定从设备的地址，地址指定的过程和上面数据传输的过程一样，只不过大多数从设备的地址是7位的，然后协议规定再给地址添加一个最低位用来表示接下来 数据传输的方向，0表示主设备向从设备写数据，1表示主设备向从设备读数据。如图所示：



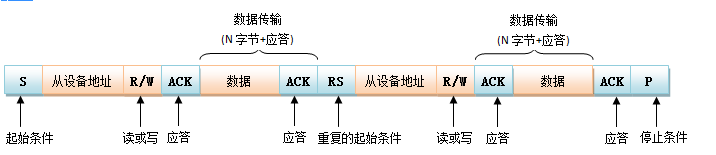
**1.1.4  I2C总线操作**  
    对I2C总线的操作实际就是主从设备之间的读写操作。大致可分为以下三种操作情况：  
    第一，主设备往从设备中写数据。数据传输格式如下：



 第二，主设备从从设备中读数据。数据传输格式如下：



 第三，主设备往从设备中写数据，然后重启起始条件，紧接着从从设备中读取数据；或者是主设备从从设备中读数据，然后重启起始条件，紧接着主设备往从设备中写数据。数据传输格式如下：



第三种操作在单个主设备系统中，重复的开启起始条件机制要比用STOP终止传输后又再次开启总线更有效率。