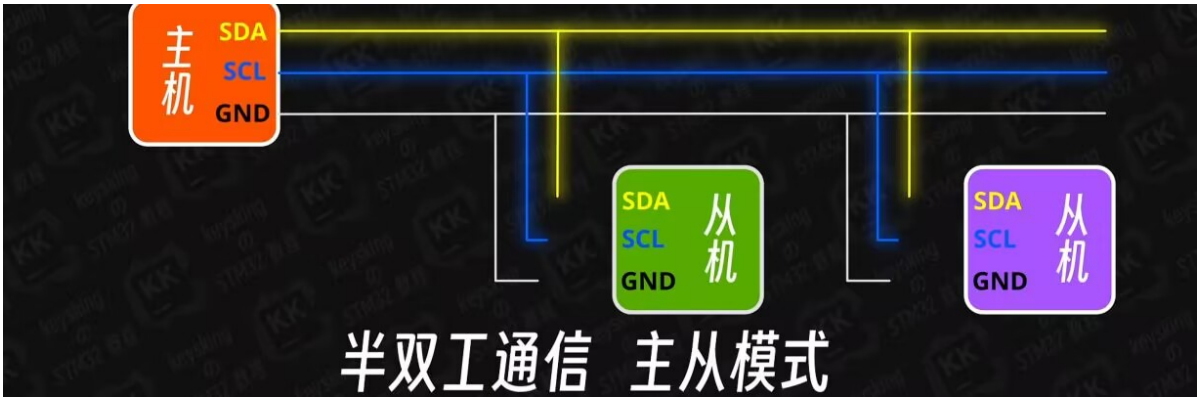
I2C的说明

I2C为半双工通信：允许双向通信，但是同一时刻只能一端发送无法两端同时进行

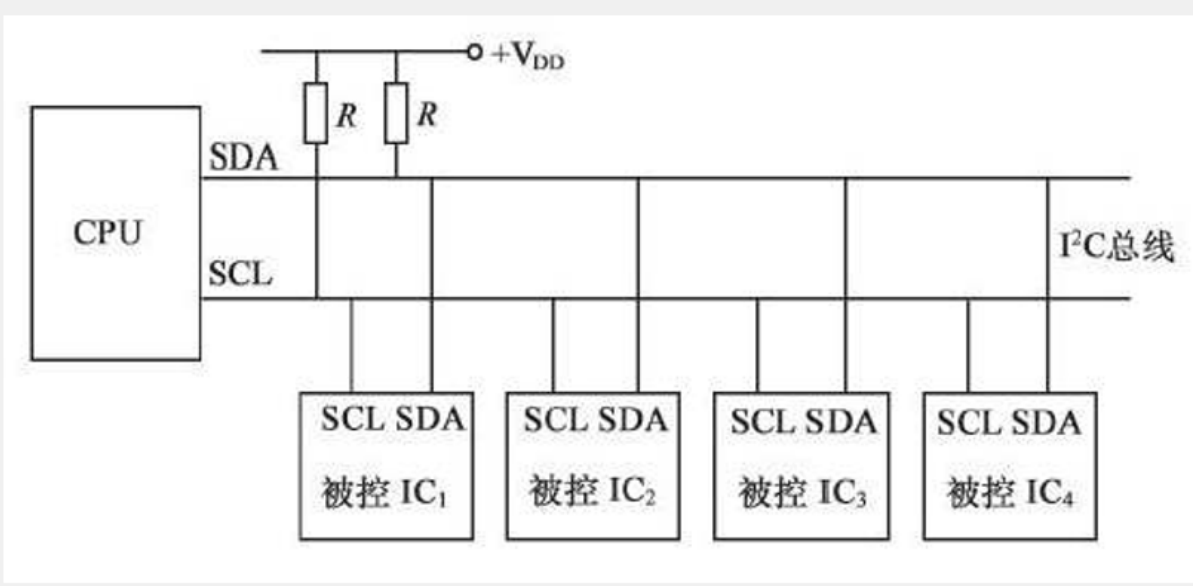


采用一主多从的形式，（也有多主多从情况，总线上任何一个模块都可以主动跳出来当老大）

其中SCL提供时钟信号，用于同步数据的发送和接收。**时钟信号仅由主设备**生成。因此，I2C与串口不同，为同步通信（即由主机通过SCL发送固定频率的脉冲信号）
（顺带一提，串口的异步通信即为通信双方首先约好通信速度，即比特率，再进行读写）

其中SDA负责传输数据。数据以串行方式在主设备和从设备之间传输，每次传输一个比特。并且基本上配置为**开漏输出。**

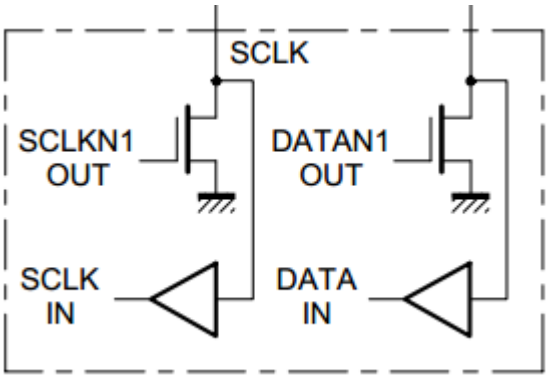
GND：电压其实是描述电势的变化量的物理量，如果不连接GND会导致各个设备有不同的基准，导致数据传输错误



同时，SCL和SDA各添加一个上拉电阻，阻值一般为4.7，1.7,2.2KΩ左右

为什么是开漏输出?

虽然主机在SCL线拥有绝对控制权可以使用推挽输出，但是SDA在输入输出中反复切换，如果因为差错导致了主机从机同时输出，且一个高电平一个低电平，那么就会导致电源短路，故禁止所有设备输出强上拉高电平，采用外置弱上拉电阻加开漏输出的电路结构，引脚内部结构如下：

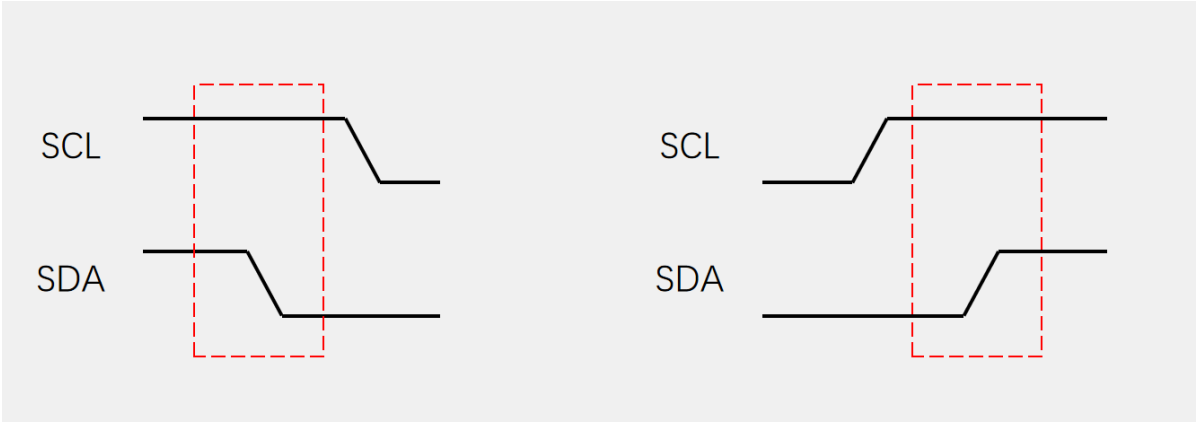


为了避免高电平造成的引脚浮空，故在总线外面SCL,SDA每条线各添加一个上拉电阻，通过一个电阻拉到高电平，故为弱上拉。完全杜绝电源短路现象（只要有任意一个设备输出低电平，总线为低电平，只有所有设备设备都输出高电平，总线才属于高电平，同时它避免了引脚模式频繁切换，开漏模式下输出高电平就相当于断开，故无需切换为输入模式，直接输出高电平即可）

I2C的时序

起始与终止条件：

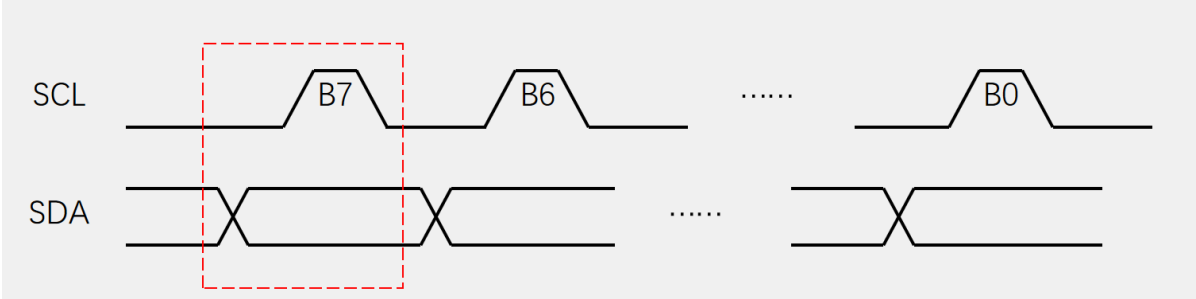
起始条件：SCL高电平期间，SDA从高到低；
终止条件：SCL高电平期间，SDA从低到高
（SCL 高电平期间是数据采样的时间点）



发送与接收一个字节：

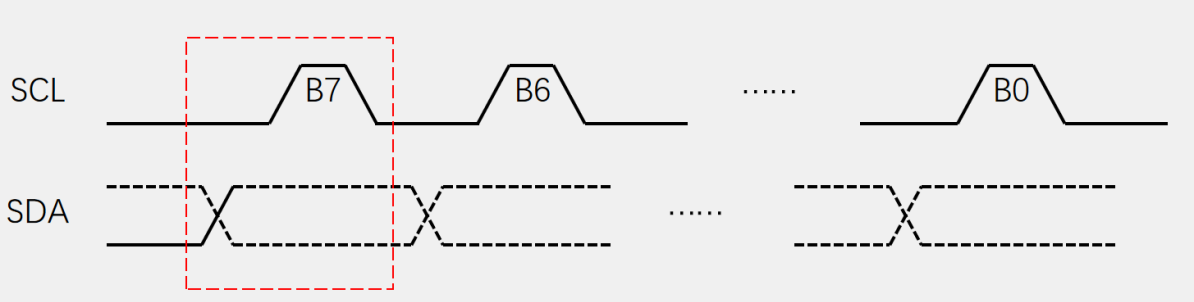
发送：

SCL低电平期间，主机将数据位依次放到SDA线上（高位先行），然后释放SCL，从机将在SCL高电平期间读取数据位，所以SCL高电平期间SDA不允许有数据变化，依次循环上述过程8次，即可发送一个字节



接收：

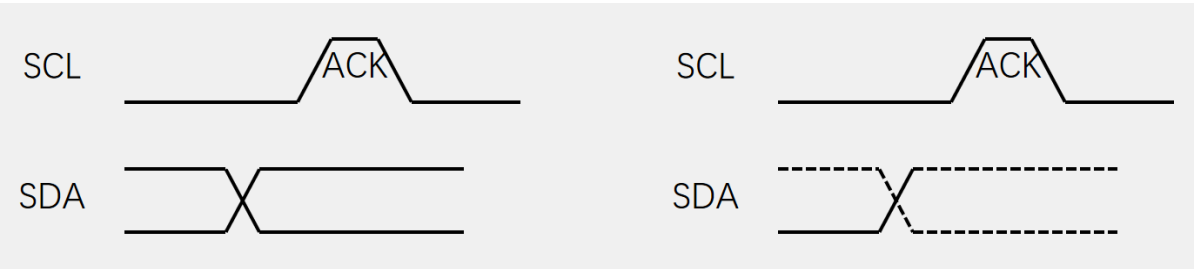
SCL低电平期间，从机将数据位依次放到SDA线上（高位先行），然后释放SCL，主机将在SCL高电平期间读取数据位，所以SCL高电平期间SDA不允许有数据变化，依次循环上述过程8次，即可接收一个字节（主机在接收之前，需要释放SDA（其实就相当于切换为输入模式），因为总线是线与特征，任何一个设备拉低了，总线就是低电平，如若主机接收之前不释放SDA，则总线始终是低电平）



发送与接收应答

发送：主机在接收完一个字节之后，在下一个时钟发送一位数据，数据0表示应答，数据1表示非应答

接收：主机在发送完一个字节之后，在下一个时钟接收一位数据，判断从机是否应答，数据0表示应答，数据1表示非应答（主机在接收之前，需要释放SDA）



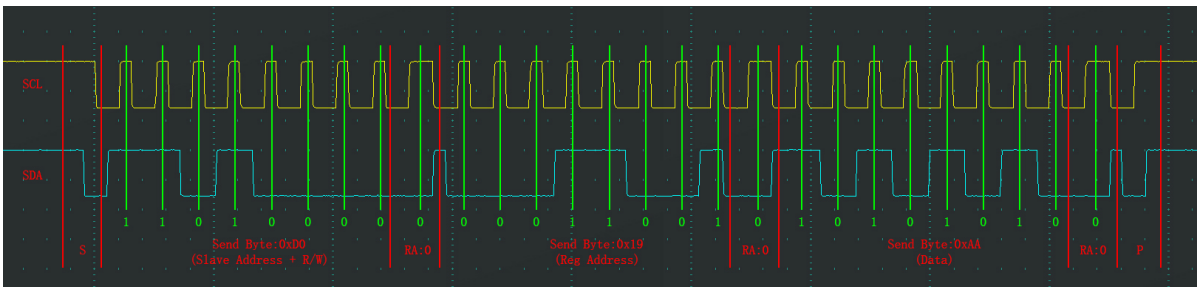
如何确定主机要访问的是哪个设备？

主机在起始条件之后，会发送一个字节，此字节就是想要联系的设备的地址（所有从机都会收到第一个字节，与自己的地址进行比较，故I2C总线里面，挂载的每个设备地址必须不一样），每个I2C设备出场时厂商一般都会为他分配一个七位地址（如MPU6050的地址为1101000），**如果有相同型号芯片挂载在同一条总线怎么办？**

运用地址中的可变部分，一般情况下，器件地址的最后几位是可以在电路中改变的（比如MPU6050地址最后一位可由板子上引脚AD0决定，接低电平为1101000，接高电平为1101001），关于地址，前几个位由厂商决定，后几位一般由引脚决定。

指定地址写

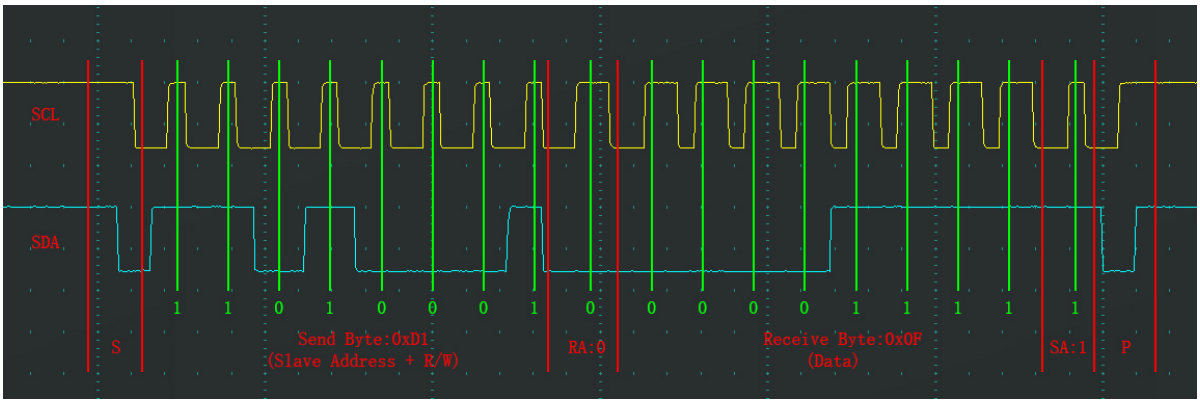
对于指定设备（Slave Address），在指定地址（Reg Address）下，写入指定数据（Data）



首先产生起始条件，之后紧跟着的时序必须是发送一个字节的时序（找到从机），字节内容必须是从机地址加读写位，读写位即确定是要发送还是要读出，0为写入，1为读出；之后从机返回应答；此时其实主机释放了SDA，但被从机控制保持了低电位。后面的上升沿便是从机释放SDA产生的，交出了SDA控制权。之后发送了寄存器地址（从机设备是可以自己定义第二个字节之后的用途的，第二个字节可以是寄存器地址或者存储器地址等），再然后就发送了我们想要发送的内容了，不想继续输入，则产生停止信号即可

当前地址读

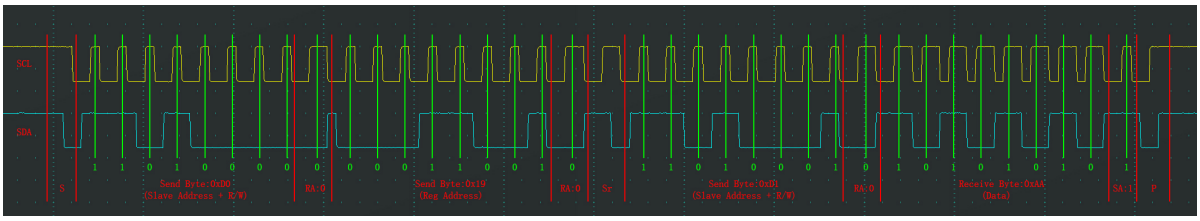
对于指定设备（Slave Address），在当前地址指针指示的地址下，读取从机数据（Data）



第一个字节读写位为1，表示读取，之后由从机进行SDA的控制。但读取的是哪个寄存器的数据应该如何判断？在**当前地址读**中主机无法决定自己读取的寄存器，在从机中所有的寄存器被分配到一个线性区域中（连续），并且会有一个单独的指针变量指示着其中一个寄存器（该指针上电默认一般指向0地址，即它的固定起始状态），并且每写入/读出一个字节后，该指针自动自增一次指向下一个字节。故未指定地址时，从机返回当前指针指向的寄存器的值

指定地址读

对于指定设备（Slave Address），在指定地址（Reg Address）下，读取从机数据（Data）



前半部分为指定地址写，后半部分为当前地址读（此类为复合格式），Sr意为重复起始条件，相当于另起一个时序，因为**指定读写标志位只能是跟着起始条件后的第一个字节**，之后如同当前地址读一样操作即可（前半部分没有进行写入/读出一个字节，故指针没有变化）

或者，你也可以在Sr之前加一个停止条件，这样就是完整的两个时序

写，读多个字节

指定地址写：最后一部分多重复几次即可，此时存储的数据地址是连续的

当前地址读/指定地址读：最后一部分多重复几次即可，连续读出一片区域的寄存器

注意：仅读一个字节就停止的话，一定要让主机发送非应答，让从机释放SDA，同理，多个字节读的话在最后一个字节发送非应答即可