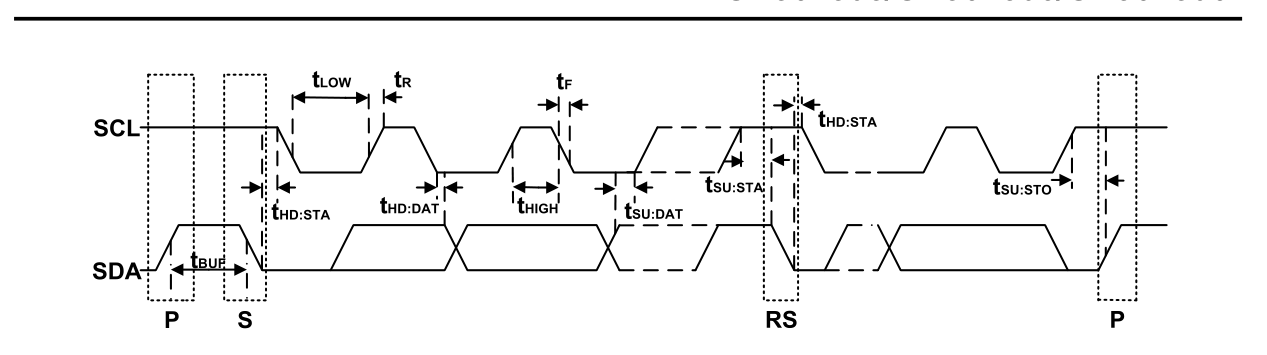
关于IIC的数据发送理解：

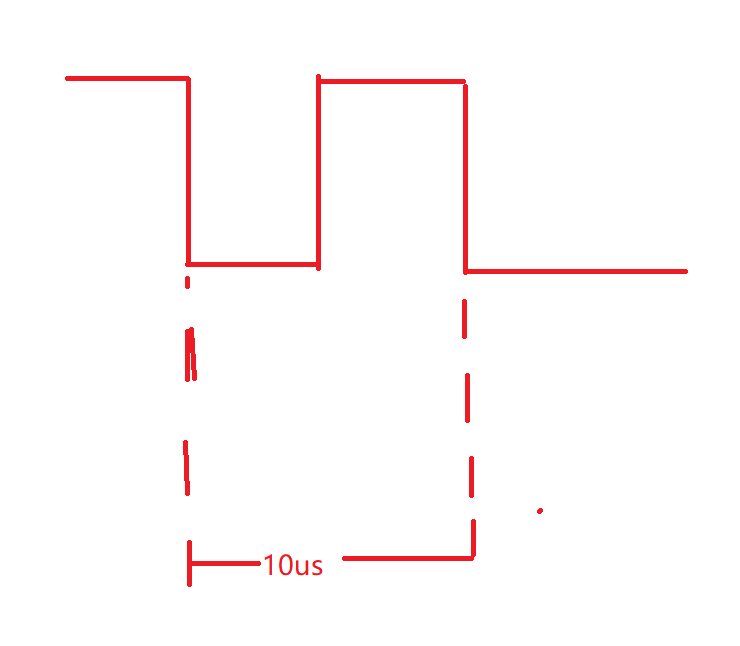
Sh267309数据发送时序图：



里面的时间长度：

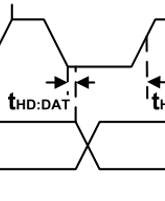


TWI频率如果我们定下来是100kHz,则每个周期是10us，这个10us是一个高电平一个低电平，那么我们需要定义出来一个delay()是5us的函数，每次需要保持数据的时候调一下这个delay().如果delay定义成10us，则我们的周期就成了20us了。



因为芯片要求scl为低sda改变电平的时候需要有个延时，有个

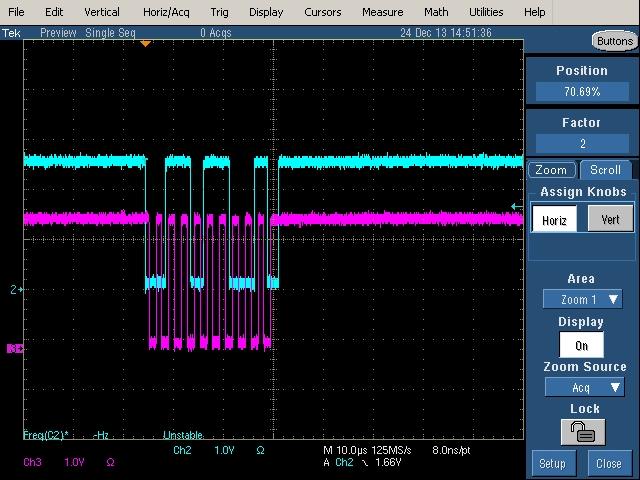


就是说scl拉低的时候sda不要直接就变化，需要稍微延时一下再去改变sda。

<https://www.pianshen.com/article/24831039784/> 链接

3、实例：

可以看到如下波形，Master发送01101100(0x6c，MSB先发)，在第9个时钟的时候SDA为高电平，表示Slave发送了NACK信号，之后整个I2C通信就结束了。这是一次失败的I2C通信，原因可能是I2C设备那边出的问题，或者访问I2C设备的地址与I2C设备实际的地址不对应，导致没接收到Master的数据从而返回NACK。



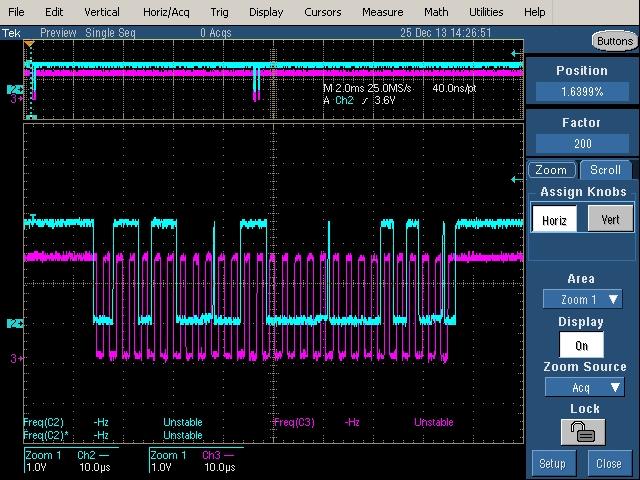
下面我拿一个OV8825 Sensor的I2C来说明，

OV8825的Slave Write Address为0x6c，OV8825的ID register Address为0x300a，0x300b，ID register里面存的Value是0x88，0x25

正常的I2C波形如下：

1）设定I2C写的地址：01101100(0x6c)  00110000(0x30) 00001010(0x0a)

Slave Write Address：0x6c，ID register address：0x300a



2）设定I2C读的地址：01101101(0x6d)  10001000(0x88)

Slave Read Address：0x6d，ID register value：0x88



20150716看到这里有点奇怪，i2c write是以ack+stop结束通信，而i2c read是以nack+stop结束通信的，原因如下：

i2c write的时候，master在写完最后一个字节之后slave会回ACK，然后master发送stop信号结束通信

i2c read的时候，master在接收完slave发送的最后一个字节之后会回NAK，因为这个时候master已经接收到足够的字节，NAK告诉slave不要在发送数据了。