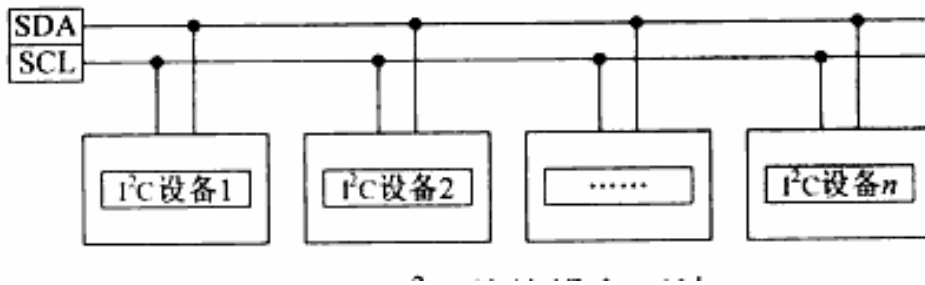


一、IIC总线介绍

IIC(Inter-Integrated Circuit,又称IIC)总线是一种由PHILIPS公司开发的串行总线，用于连接微控制器及其外围设备，它具有如下特点。

- (1)只有两条总线线路:一条串行数据线(SDA),一条串行时钟线(SCL)
- (2)每个连接到总线的器件都可以使用软件根据它的唯一的地址来识别
- (3)传输数据的设备间是简单的主/从关系
- (4)主机可以用作主机发送器或主机接收器
- (5)它是一个真正的多主机总线，两个或多个主机同时发起数据传输时，可以通过冲突检测和仲裁来防止数据被破坏
- (6)串行的8位双向数据传输，位速率在标准模式下可达100kbit/s,在快速模式下可达400kbit/s,在高速模式下可达3.4Mbit/s。



注意:当多个主机试图去控制总线时，通过仲裁可以使得只有一个主机获得总线控制权，并且它传输的信息不被破坏

二、IIC总线的信号类型

IIC总线在传送数据过程中共有3种类型信号:开始信号、结束信号和响应信号

- (1)开始信号(S):SCL 为高电平时，SDA由高电平向低电平跳变，开始传送数据
- (2)结束信号(P):SCL为高电平时，SDA由低电平向高电平跳变，结束传送数据

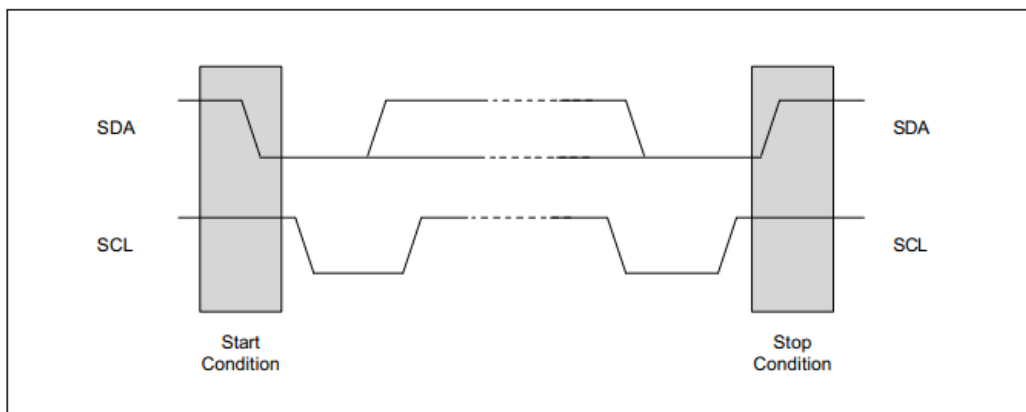


Figure 20-2. Start and Stop Condition

- (3)响应信号(ACK):接收器在接收到8位数据后，在第9个时钟周期，拉低SDA电平

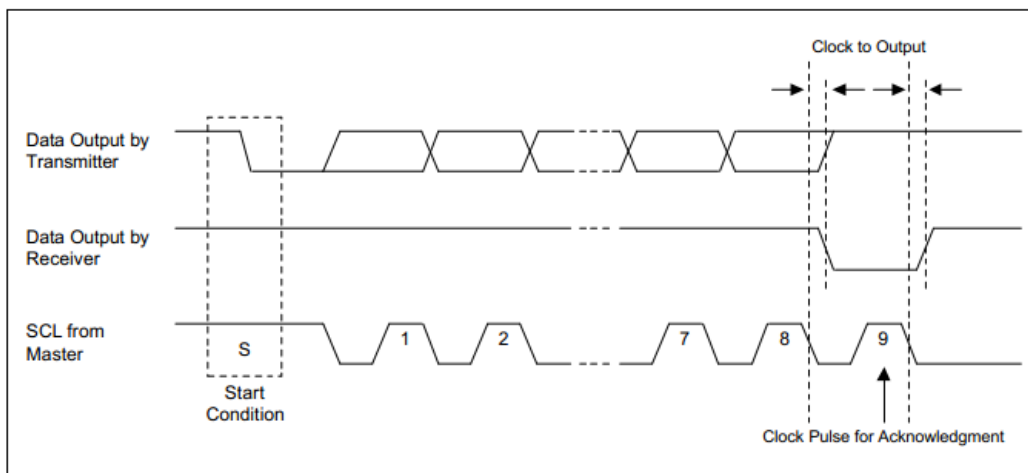


Figure 20-5. Acknowledge on the IIC-Bus

注意:SDA上传输的数据必须在SCL为高电平期间保持稳定，SDA上的数据只能在SCL为低电平期间变化

三、IIC总线的数据传输格式

发送到SDA线上的每个字节必须是8位的，每次传输可以发送的字节数量不受限制。首先传输的是数据的最高位(MSB)。

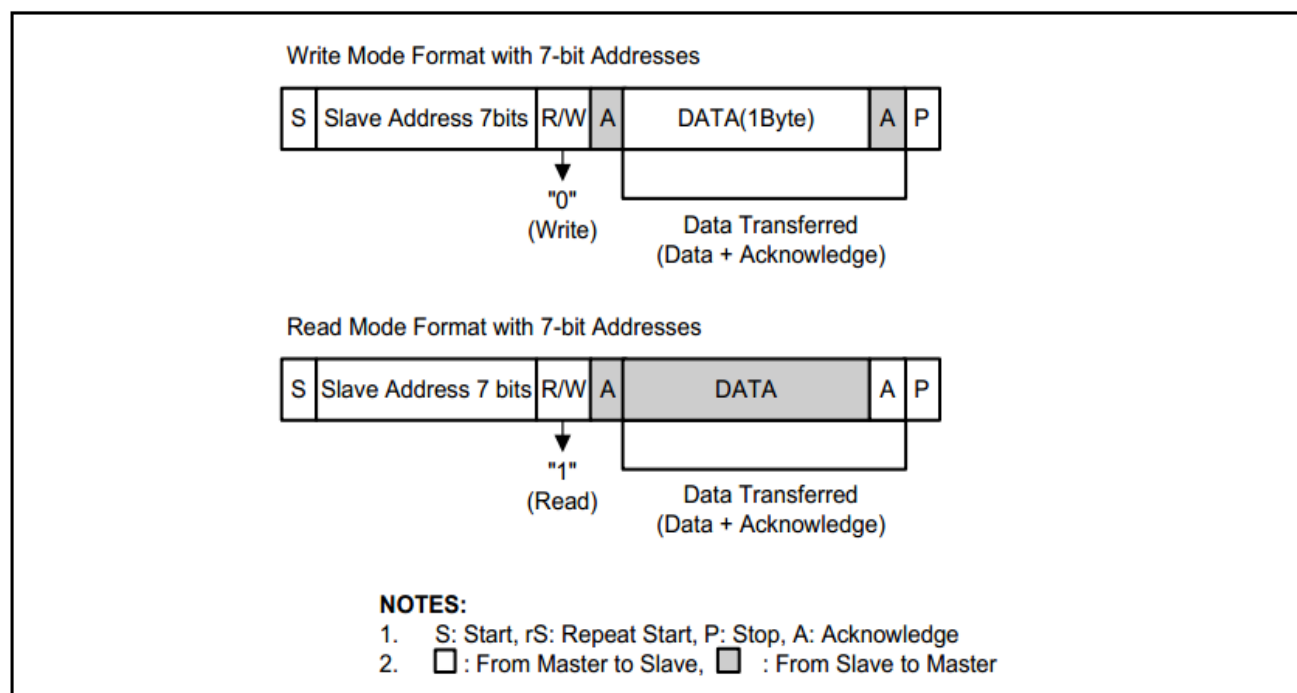


Figure 8.2-3 I²C-Bus Interface Data Format

启动一个传输时，主机先发送S信号，然后发出8位数据。这8位数据中前7位为从机的地址，第8位表示传输的方向(0表示写操作，1表示读操作)。从机收到后会发出一个ACK信号。

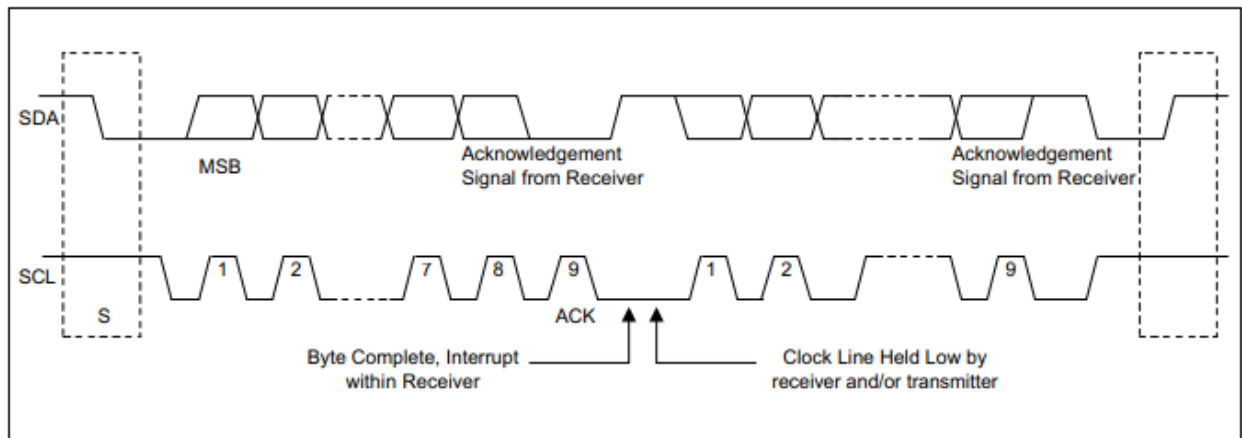


Figure 20-4. Data Transfer on the IIC-Bus

注意:主机接收器在接收到最后一个字节后,也不会发出ACK信号。于是,从机发送器释放SDA线,以允许主机发出P信号结束传输。

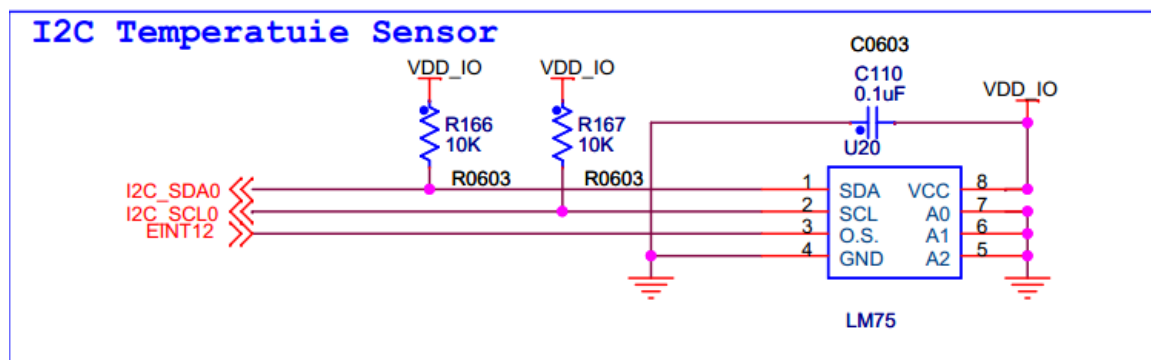
从机接收数据,主机想停止, 主机接收数据,主机想停止

主机----->从机

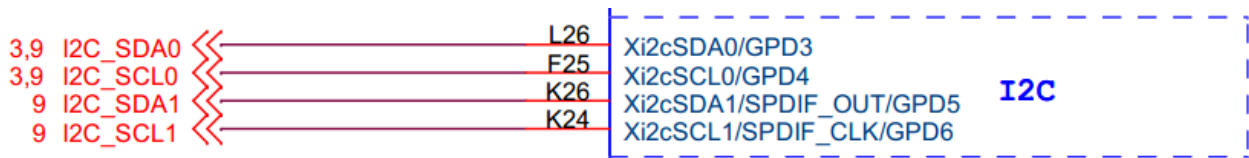
三、用GPIO模拟IIC总线时序

通过前面的介绍,我们已经了解了IIC总线。下面我们将用2个pin来模拟IIC的总线时序,来读取稳度传感器LM75的测量的温度。

先来看看LM75硬件的上的连线:



其中I2C_SDA0是数据线,I2C_SCL0是时钟线,它们分别接到s5pc100的GPD3和GPD4,如下图所示



在模拟IIC的时候,用GPD3引脚发送数据或读取数据,用GPD4引脚提供时钟信号即可。

由于s5pc100的GPD这一组的引脚很多,不好单独对其中的某一个引脚操作,为了能单独对其中的某一个管脚单独操作,这里使用了C语言的位域。

```
typedef struct
{
    uint8 GPDDAT_0:1;
    uint8 GPDDAT_1:1;
    uint8 GPDDAT_2:1;
    uint8 GPDDAT_3:1;
    uint8 GPDDAT_4:1;
}
```

```

uint8 GPDDAT_5:1;
uint8 GPDDAT_6:1;
uint8 GPDDAT_7:1;

}gpddat_t;

#define GPD_DAT (* (volatile gpddat_t *)0xE0300084)

#define SDA GPD_DAT.GPDDAT_3
#define SCL GPD_DAT.GPDDAT_4

```

(1)产生IIC起始信号

SCL 为高电平时，SDA由高电平向低电平跳变，开始传送数据

```

/*IIC START:SCL = 1,SDA = 1->0*/
void iic_start()
{
    SDA = HIGH;
    SCL = HIGH;
    delay(50);

    //高到低的跳变产生start信号
    SDA = LOW;
    delay(50);

    //在SCL高时，SDA必须保持稳定,SCL低时，SDA可以任意改变
    //此处将SCL拉低的目的是，接下来就要发送数据了
    SCL = LOW;
    delay(50);

    return;
}

```

(2)产生IIC停止信号

SCL为高电平时，SDA由低电平向高电平跳变，结束传送数据

```

/*IIC STOP:SCL = 1,SDA = 0->1*/
void iic_stop()
{
    SDA = LOW;
    SCL = LOW;
    delay(50);

    SCL = HIGH;
    delay(50);
    //SCL为高电平时，SDA从低电平跳变到高电平
    //，产生停止信号
    SDA = HIGH;
    delay(50);
}

```

```
return;  
}
```

(3)发送数据

```
/*Write 1 Byte to IIC*/1010 1001  
void iic_write_byte(uint8 data)  
{  
    uint8 loop;  
  
    data:1010 0101  
    for(loop = 8;loop > 0;loop --)  
    {  
        //先发送最高位,在SCL高电平时,SDA必须保持稳定  
        SDA = data >> 7;  
        SCL = HIGH;  
        delay(50);  
  
        //SCL为低电平时, SDA可以任意改变  
        SCL = LOW;  
        //低位向高位移动  
        data <<= 1;//data = data << 1  
        delay(50);  
    }  
  
    return;  
}
```

(4)读取数据

```
/*Read 1 byte from IIC*/  
uint8 iic_read_byte()  
{  
    uint8 loop;  
    uint8 value = 0;  
  
    for(loop = 8; loop > 0;loop --)  
    {  
        SCL = HIGH;  
  
        delay(50);  
  
        value <<= 1;//value = value << 1  
  
        //读取1位数据  
        value |= SDA;//value = value | 1  
  
        SCL = LOW;  
  
        delay(50);  
    }  
}
```

```
return value;
}
```

(5)主机向从机发送ACK信号

```
void iic_send_ack()
{
    SCL = HIGH;
    SDA = LOW;
    delay(50);

    SCL = LOW;
    delay(50);

    return;
}
```

(6)获取从机给主机的ACK信号

```
uint8 iic_get_ack()
{
    uint8 ret;

    SCL = HIGH;
    delay(50);

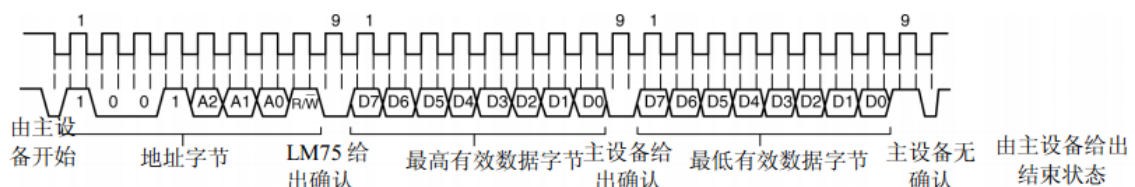
    ret = SDA;

    SCL = LOW;
    delay(50);

    return ret;
}
```

四、读取LM75测量的温度值

通过IIC读取LM75测量温度值的时序如下:



实例代码如下:

```
unsigned int __read_lm75()
{
    uint8 ack;
    uint8 high,low;

    //设置IIC连接的pin为输出模式
    SET_GPIO_MODE(GPD.GPDCON,3,1);
    SET_GPIO_MODE(GPD.GPDCON,4,1);
```

```

delay(100);

//产生起始信号
iic_start();

//发送从机地址:10010001
iic_write_byte(0x91);

//设置IIC连接的pin(SDA)为输入模式
SET_GPIO_MODE(GPD.GPDCON,3,0);

//等待从机的ACK
do{
ack = iic_get_ack();
}while(ack);

//读取从机发送过来的数据
high = iic_read_byte();

//设置IIC连接的pin(SDA)为输出模式
SET_GPIO_MODE(GPD.GPDCON,3,1);

//发送ACK信号
iic_send_ack();

//设置IIC连接的pin(SDA)为输入模式
SET_GPIO_MODE(GPD.GPDCON,3,0);

//读取从机发送过来的数据
low = iic_read_byte();

//设置IIC连接的pin(SDA)为输出模式
SET_GPIO_MODE(GPD.GPDCON,3,1);

//发送停止信号
iic_stop();

return (high << 8) | low;
}

```

实验的经验:

- 1.读不到从机发送的ACK信号，原因是没有将SDA的那根线设为输入模式
- 2.在做的过程中，一开始每次读取的时候温度的值一直没有改变，后来发现是由于每次没有发送停止信号产生的。

最后附上整个实验的代码

