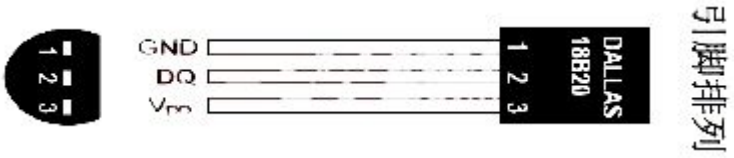
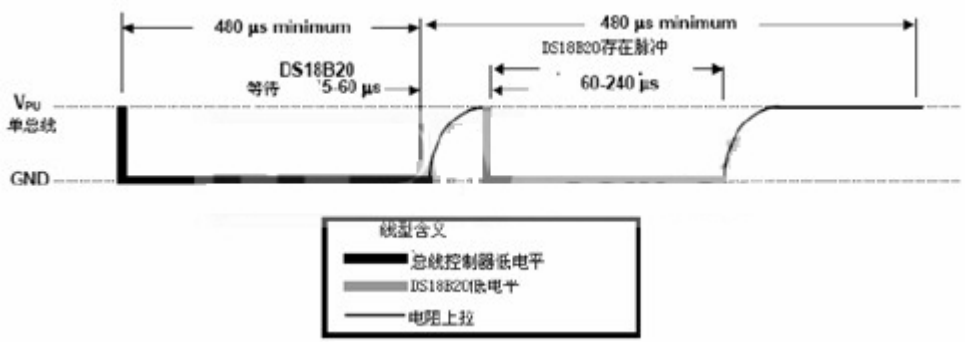
单片机数据通信之单总线数据传输分析

纯单片机干不了大事，必须得配上各种外设，那么了解单片机与传感器之间的数据通信就显得必不可少了。常见的单片机数据通信方式有 SPI,IIC,RS232，单总线等等。每种通信方式都有相应的时序图，分析时序图并完成代码的编写是单片机学习者的必修课。本文以 DS18B20 为例分析一下单总线数据传输。

DS18B20 是单总线数据传输，因此对于时序的要求就非常的高，学会分析其时序图是非常有必要的。



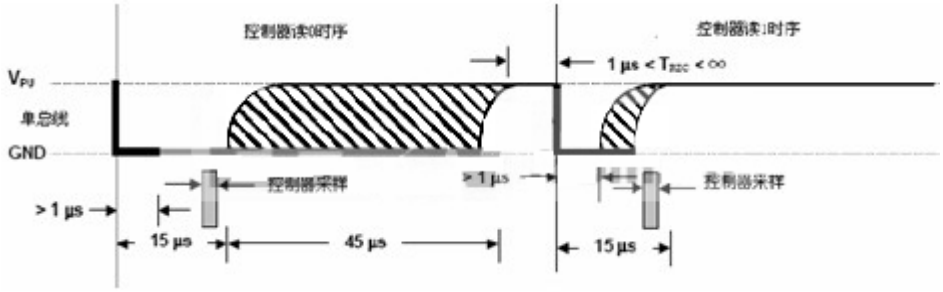
1.初始化时序图分析：



首先是由总线控制器拉低总线，维持 480us。在 480us 后释放总线，由上拉电阻讲总线拉高。等待 5-60us 后，DS18B20 开始响应，会将数据总线拉低 60-240us.之后便释放总线，由上拉电阻拉高总线。转换为代码如下：

```
u8dsbInit()//初始化，返回 0 表示 DS18B20 无反应，反之有响应
{
dsbDQStat(0);//控制器拉低总线
delay500us();//拉低总线一段时间
dsbDQStat(1);//释放总线
delay60us();//等待 DS18B20 响应
if(dsb_DQ)//如果没有相应直接返回 0
{
return0;
}
delay240us();//有响应则等待响应结束
return1;//返回初始化状态
}
```

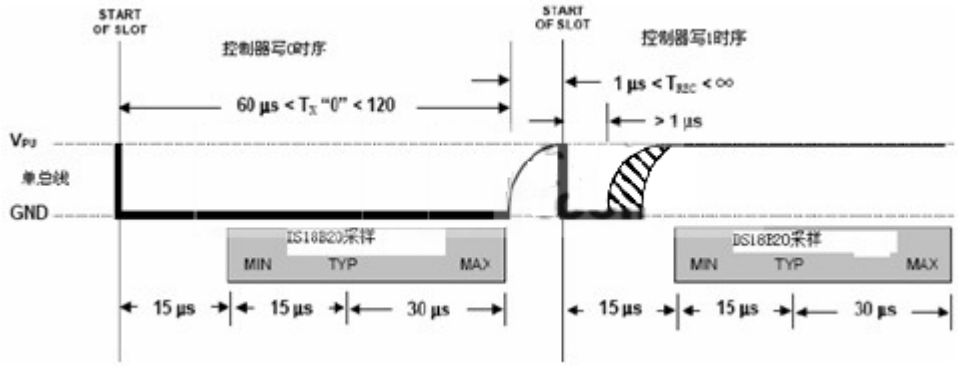
2.读时序图分析：



首先由控制器将总线拉低>1us 的时间，此时控制器释放总线，如果此时控制器采样为低电平，那么读到的值便是 0，如果为高电平，则读到的值为 1。注意图中标有一个 15us，其意思便是控制器采样在 15us 内完成。15us 后是由上拉电阻将总线拉高维持 45us。整个读周期为 15+45=60us。这个周期的时间也是得控制的。转换为代码如下：

```
u8dsbReadByte()//读出一个字节的数据，从低位开始读取
{
u8i,tmp=0;
for(i=0;i<8;i++)
{
dsbDQStat(0);//控制器拉低总线
tmp>>=1;//低位开始读
dsbDQStat(1);//释放总线
if(dsb_DQ)tmp|=0x80;
delay15us();
delay45us();//控制周期时间
}
returntmp;
}
```

3.写时序图分析：



首先由控制器拉低总线 15us，之后，如果要写入 0，则继续拉低总线并为此 45us.如果要写入 1 则释放总线由上拉电阻拉高总线，也为此 45us。写时序相对比较简单，转换为代码如下：

```
voiddsbWriteByte(u8dat)//写一个字节的数据，从低位开始
{
u8i;
for(i=0;i<8;i++)
{
dsbDQStat(0);//控制器拉低总线
delay15us();//维持 15us
if(dat&0x01)dsbDQStat(1);
elsedsbDQStat(0);
dat>>=1;
delay45us();
dsbDQStat(1);//45us 后释放总线
}
}
```

DS18B20 的三个时序图就分析完了，DS18B20 只是单总线数据通信中的一个例子，大家了解了 DS18B20 时序图的分析，那么就可以试试分析 DHT11 的时序图完成其初始化函数，以及读数据函数。