

【常用传感器】DS18B20温度传感器原理详解及例程代码

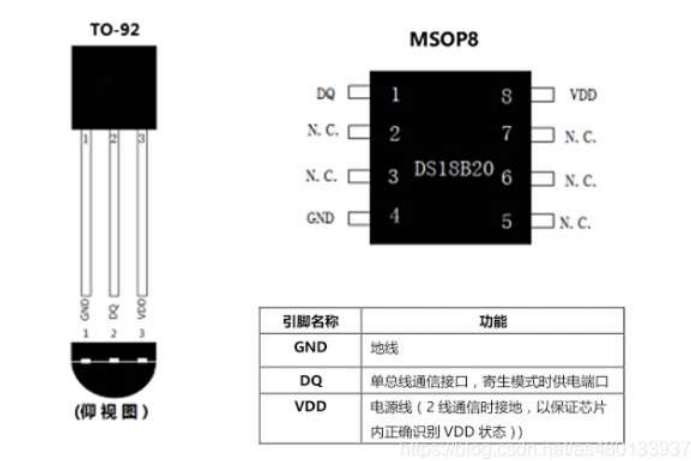
数字温度传感器（DS18B20）

传感器参数

- 测温范围为-55℃到+125℃，在-10℃到+85℃范围内误差为±0.4°。
- 返回16位二进制温度数值
- 主机和从机通信使用单总线，即使用单线进行数据的发送和接收
- 在使用中不需要任何外围元件，独立芯片即可完成工作。
- 掉电保护功能 DS18B20 内部含有 EEPROM，通过配置寄存器可以设定数字转换精度和报警温度，在系统掉电以后，它仍可保存分辨率及报警温度的设定值。
- 每个DS18B20都有**独立唯一**的64位-ID，此特性决定了它可以将任意多的DS18b20挂载到一根总线上，通过ROM搜索读取相应DS18B20的温度值
- 宽电压供电，电压2.5V~5.5V
- DS18B20返回的16位二进制数代表此刻探测的温度值，其高五位代表正负。如果高五位全部为1，则代表返回的温度值为负值。如果高五位全部为0，则代表返回的温度值为正值。后面的11位数据代表温度的绝对值，将其转换为十进制数值之后，再乘以0.0625即可获得此时的温度值。

传感器引脚及原理图

DS18B20传感器的引脚及封装图如下:

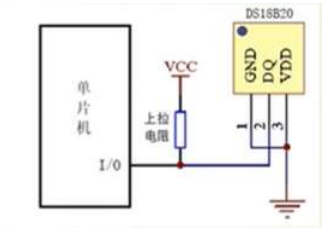


DS18B20一共有三个引脚，分别是：

- GND：电源地线
- DQ：数字信号输入 / 输出端。
- VDD：外接供电电源输入端。

DS18B20寄生电源

DS18B20的另一个特点是不需要再外部供电下即可工作。当总线高电平时能量由单线上拉电阻经过DQ引脚获得。高电平时同时充电一个内部电容，当总线低电平时由此电容供应能量。这种供电方法被称为“寄生电源”。另外一种选择是DS18B20由接在VDD的外部电源供电



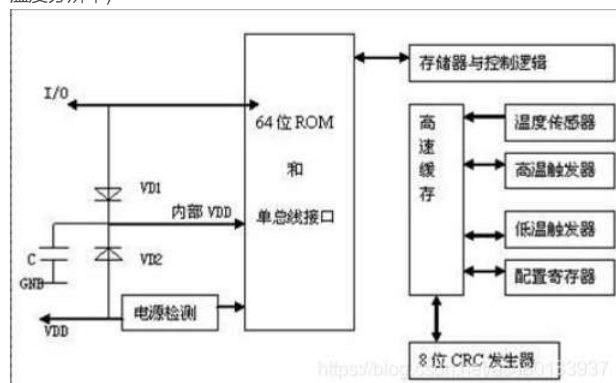
DS18B20内部构成

- 主要由以下3部分组成：**64 位ROM，高速暂存器，存储器**
- 64 位ROM存储独有的序列号

高速暂存器包含：

一个字节的温度上限和温度下限报警触发器(TH和TL)

存储器：由一个高速的RAM和一个可擦除的EEPROM组成，EEPROM存储高温和低温触发器(TH和TL)以及配置寄存器的值，(就是存储低温和高温报警值以及温度分辨率)

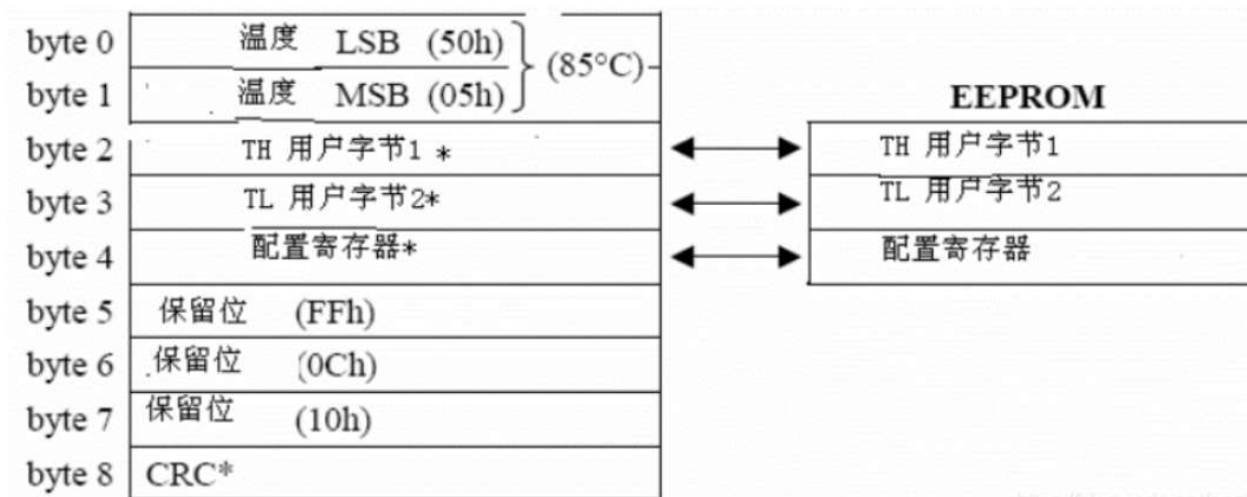


高速暂存器由9个字节组成。

字节2~3 是用户用来设置最高报警和最低报警值(TH和TL)。

字节5~7 保留位。芯片内部使用

字节8 CRC校验位。是64位ROM中的前56位编码的校验码。由CRC发生器产生。



DS18B20采用**16位补码的形式来存储温度数据**，**温度是摄氏度**。当温度转换命令发布后，经转换所得的温度值以二字节补码形式存放在高速暂存存储器的第0和第1个字节。

剩下的11位为温度数据位，对于12位分辨率，所有位全部有效，对于11位分辨率，位0(bit0)无定义，对于10位分辨率，位0和位1无定义，对于9位分辨率，位0，位1，和位2无定义

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
低字节	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
高字节	S	S	S	S	S	2^6	2^5	2^4

对应的温度计算：

当五个符号位S=0时，温度为正值，直接将后面的11位二进制转换为十进制，再乘以0.0625(12位分辨率)，就可以得到温度值；

当五个符号位S=1时，温度为负值，先将后面的11位二进制补码变为原码(符号位不变，数值位取反后加1)，再计算十进制值。再乘以0.0625(12位分辨率)，就可以得到温度值；

例如：

+125℃的数字输出07D0(00000111 11010000)

转换成10进制是2000，对应摄氏度：0.0625x2000=125℃

-55℃的数字输出为 FC90。

首先取反，然后+1，转换成原码为：11111011 01101111

数值位转换成10进制是870，对应摄氏度：-0.0625x870=-55℃

温度/℃	二进制表示	十六进制表示
+125	00000111 11010000	07D0H
+25.0625	00000001 10010001	0191H
+10.125	00000000 10100010	00A2H
+0.5	00000000 00001000	0008H
0	00000000 00000000	0000H
-0.5	11111111 11111000	FFF8H
-10.125	11111111 01011110	FF5EH
-25.0625	11111110 01101111	FE6FH
-55	11111100 10010000	FC90H

代码：

```
1 unsigned int Temp1,Temp2,Temperature; //Temp1低八位, Temp2高八位
2 unsigned char Minus Flag=0; //负温度标志位
3
4 if (Temp2&0xFC) //判断符号位是否为1
5 {
6     Minus Flag=1; //负温度标志位置1
7     Temperature=((Temp2<<8)|Temp1); //高八位第八位进行整合
8     Temperature=((Temperature)+1); //讲补码转换为原码, 求反, 补1
9     Temperature*=0.0625; //求出十进制
10 }
11 else //温度为正值
12 {
13     Minus Flag=0; //负温度标志位置0
14     Temperature =((Temp2<<8) |Temp1)*0.0625;
15 }
```

配置寄存器

在配置寄存器中，我们可以通过R0和R1设置DS18B20的转换分辨率，DS18B20在上电后默认R0=1和R1=1(12分辨率)，寄存器中的第7位和第0位到4位保留给设备内部使用。

配置寄存器

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	R1	R0	1	1	1	1	1

温度计精确度配置

R1	R0	精度	最大转换时间	
0	0	9-bit	93.75 ms	($t_{CONV}/8$)
0	1	10-bit	187.5 ms	($t_{CONV}/4$)
1	0	11-bit	375 ms	($t_{CONV}/2$)
1	1	12-bit	750 ms	(t_{CONV})

单总线系统

在每个DS18B20内部都有一个唯一的64位长的序列号，这个序列号值就存在DS18B20内部的ROM中。开始的8位是产品类型编码(DS18B20是28H)，接着的48位是每个器件唯一的序号，最后的8位是CRC校验码。

8 位 CRC 编号	48 位序列号	8 位产品系列编码
------------	---------	-----------

一线总线系统使用单总线主控来控制一个或多个从机设备。每个DS18B20都有**独立唯一**的64位-ID，此特性决定了它可以将任意多的DS18B20挂载到一根总线上，通过ROM搜索读取相应DS18B20的温度值。

DS18B20工作步骤

DS18B20的工作步骤可以分为三步：

- 1.初始化DS18B20
- 2.执行ROM指令
- 3.执行DS18B20功能指令

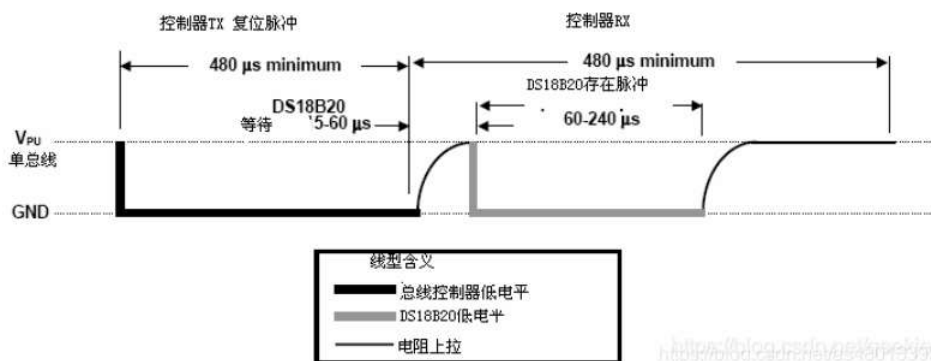
其中第二步执行ROM指令，也就是访问每个DS18B20，搜索64位序列号，读取匹配的序列号值，然后匹配对应的DS18B20，如果我们仅仅使用单个DS18B20，可以直接跳过ROM指令。而跳过ROM指令的字节是0xCC。

1.初始化DS18B20

任何器件想要使用，首先就是需要**初始化**，对于DS18B20单总线设备，首先初始化单总线为**高电平**，然后总线开始也需要检测这条总线上是否存在DS18B20这个器件。如果这条总线上存在DS18B20，总线会根据时序要求返回一个**低电平脉冲**，如果不存在的话，也就不会返回脉冲，即总线保持为高电平。

初始化具体时序步骤如下：

- 1.单片机拉低总线至少480us，产生复位脉冲，然后释放总线（拉高电平）。
- 2.这时DS18B20检测到请求之后，会拉低信号，大约60~240us表示应答。
- 3.DS18B20拉低电平的60~240us之间，单片机读取总线的电平，如果是低电平，那么表示初始化成功
- 4.DS18B20拉低电平60~240us之后，会释放总线。



代码如下：

```

1  /****初始化DS18B20****/
2  unsigned int Init_DS18B20(void)
3  {
4      unsigned int x=0;
5      DQ = 1;      //DQ复位
6      delay(4);    //稍做延时
7      DQ = 0;      //单片机将DQ拉低
8      delay(60);   //精确延时,大于480us
9      DQ = 1;      //拉高总线
10     delay(8);
11     x = DQ;      //稍做延时后,如果x=0则初始化成功,x=1则初始化失败
12     delay(4);
13     return x;
14 }

```

2.写时序

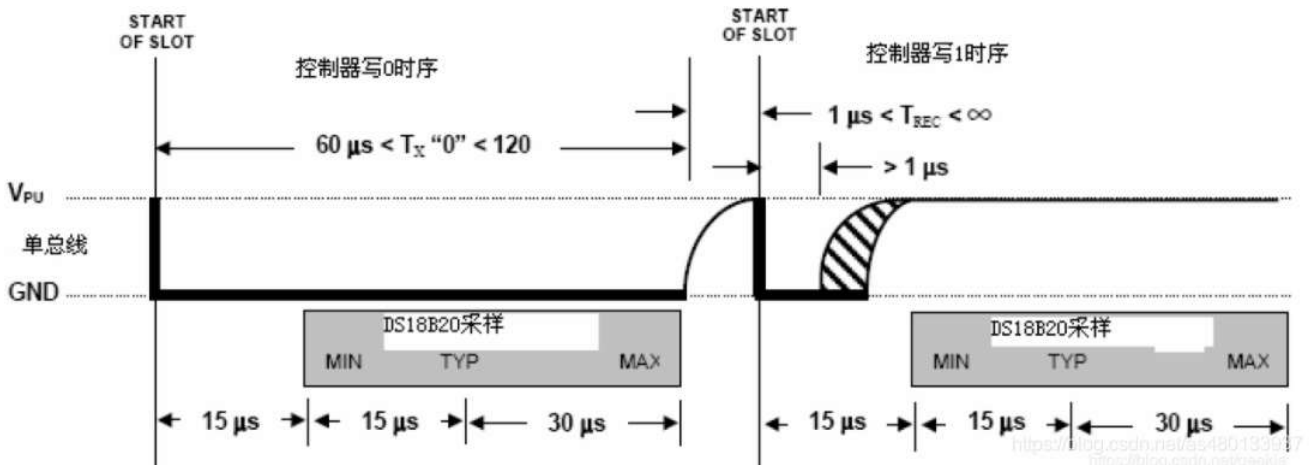
总线控制器通过控制单总线高低电平持续时间从而把逻辑1或0写入DS18B20中。每次只传输1位数据

单片机想要给DS18B20写入一个0时,需要将单片机引脚拉低,保持低电平时间要在60~120us之间,然后释放总线

单片机想要给DS18B20写入一个1时,需要将单片机引脚拉低,拉低时间需要大于1us,然后在15us内拉高总线。

在写时序起始后15μs到60μs期间,DS18B20处于采样单总线电平状态。如果在此期间总线为高电平,则向DS18B20写入1;如果总线为低电平,则向DS18B20写入0。

注意:2次写周期之间至少间隔1us



```

1  /****写一个字****/
2  void WriteOneChar(unsigned char dat)
3  {
4      unsigned char i=0;
5      for (i=8; i>0; i--)
6      {
7          DQ = 0;
8          DQ = dat&0x01; //与1按位与运算, dat最低位为1时DQ总线为1, dat最低位为0时DQ总线为0
9          delay(4);
10         DQ = 1;
11         dat>>=1;
12     }
13     delay(4);
14 }
15

```

DS18B20写入的功能命令:

ROM指令:

采用多个DS18B20时,需要写ROM指令来控制总线上的某个DS18B20

如果是单个DS18B20,直接写跳过ROM指令0xCC即可

指令名称	指令代码	指令功能
读 ROM	33H	读 DS18B20ROM 中的编码（即读 64 位地址）
ROM 匹配（符合 ROM）	55H	发出此命令之后，接着发出 64 位 ROM 编码，访问单总线上与编码相对应 DS18B20 使之作出响应，为下一步对该 DS18B20 的读写作准备
搜索 ROM	0F0H	用于确定挂接在同一总线上 DS18B20 的个数和识别 64 位 ROM 地址，为操作各器件作好准备
跳过 ROM	0CCH	忽略 64 位 ROM 地址，直接向 DS18B20 发温度变换命令，适用于单片机工作
警报搜索	0ECH	该指令执行后，只有温度超过设定值上限或下限的片子才做出响应

RAM指令，DS18B20的一些功能指令

常用的是：

温度转换 0x44

开启温度读取转换，读取好的温度会存储在高速暂存器的第0个和第一个字节中

读取温度 0xBE

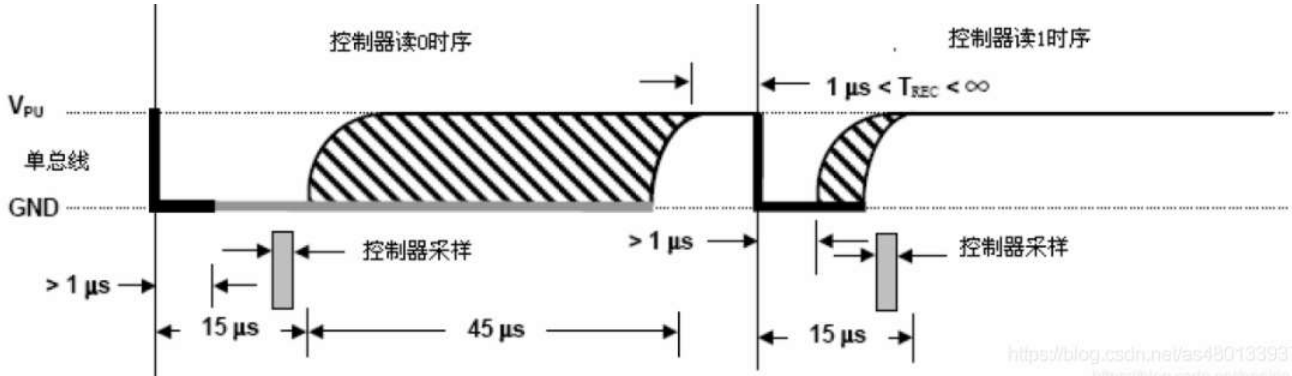
读取高速暂存器存储的数据

指令名称	指令代码	指令功能
温度变换	44H	启动 DS18B20 进行温度转换，转换时间最长为 500ms（典型为 200ms），结果存入内部 9 字节 RAM 中
读暂存器	0BEH	读内部 RAM 中 9 字节的内容
写暂存器	4EH	发出向内部 RAM 的第 3，4 字节写上，下限温度数据命令，紧跟该命令之后，是传送两字节的数据
复制暂存器	48H	将 RAM 中第 3，4 字节的内容复制到 EEPROM 中
重调 EEPROM	0B8H	EEPROM 中的内容恢复到 RAM 中的第 3，4 字节
读供电方式	0B4H	读 DS18B20 的供电模式，寄生供电时 DS18B20 发送“0”，外接电源供电 DS18B20 发送“1”

读时序

读时隙由主机拉低总线电平至少1μs然后再释放总线，读取DS18B20发送过来的1或者0

DS18B20在检测到总线被拉低1微秒后，便开始送出数据，若是要送出0就把总线拉为低电平直到读周期结束。若要送出1则释放总线为高电平。



注意：所有读时隙必须至少需要60us，且在两次独立的时隙之间至少需要1ps的恢复时间

同时注意：主机只有在发送**读暂存器**命令(0xBE)或**读电源**类型命令(0xB4)后，立即生成读时隙指令，DS18B20才能向主机传送数据。也就是先发读取指令，再发送读时隙

最后一点：写时序是先写命令的低字节，比如写入跳过ROM指令0xCC(11001100),写的顺序是“零、零、壹、壹、零、零、壹、壹”，

读时序时是先读低字节，在读高字节，**也就是先读取高速暂存器的第0个字节(温度的低8位)，在读取高速暂存器的第1个字节(温度的高8位)** 我们正常使用DS18B20读取温度读取两个温度字节即可