

DS18B20

正确接线

如图。左负右正，一旦接反就会立刻发热，有可能烧毁！接反是导致该传感器总是显示 85°C 的原因。面对着扁平的那一面，左负右正。

概述

DS18B20 数字温度传感器接线方便，封装后可应用于多种场合，如管道式，螺纹式，磁铁吸附式，不锈钢



封装式，型号多种多样，有 LTM8877，LTM8874 等等。主要根据应用场合的不同而改变其外观。封装后的 DS18B20 可用于电缆沟测温，高炉水循环测温，锅炉测温，机房测温，农业大棚测温，洁净室测温，弹药库测温等各种非极限温度场合。耐磨耐碰，体积小，使用方便，封装形式多样，适用于各种狭小空间设备数字测温和控制领域。

1: 技术性能描述

- ①、独特的单线接口方式，DS18B20 在与微处理器连接时仅需要一条口线即可实现微处理器与 DS18B20 的双向通讯。
- ②、测温范围 $-55^{\circ}\text{C}\sim+125^{\circ}\text{C}$ ，固有测温误差（注意，不是分辨率，这里之前是错误的） 0.5°C 。
- ③、支持多点组网功能，多个 DS18B20 可以并联在唯一的三线上，最多只能并联 8 个，实现多点测温，如果数量过多，会使供电电源电压过低，从而造成信号传输的不稳定。
- ④、工作电源: $3\sim 5\text{V}/\text{DC}$ （可以数据线寄生电源）
- ⑤、在使用中不需要任何外围元件
- ⑥、测量结果以 9~12 位数字量方式串行传送
- ⑦、不锈钢保护管直径 $\Phi 6$
- ⑧、适用于 DN15~25, DN40~DN250 各种介质工业管道和狭小空间设备测温
- ⑨、标准安装螺纹 M10X1, M12X1.5, G1/2" 任选
- ⑩、PVC 电缆直接出线或德式球型接线盒出线,便于与其它电器设备连接。

DS18B20+ 和 Maxim Integrated 信息^[1]

Manufactured by Maxim Integrated, DS18B20+ is a 温度传感器.

应用范围

- 2.1 该产品适用于冷冻库，粮仓，储罐，电讯机房，电力机房，电缆线槽等测温和控制领域
- 2.2 轴瓦，缸体，纺机，空调，等狭小空间工业设备测温和控制。
- 2.3 汽车空调、冰箱、冷柜、以及中低温干燥箱等。
- 2.4 供热/制冷管道热量计量，中央空调分户热能计量和工业领域测温和控制

型号规格

型 号	测温范围	安装螺纹	电缆长度	适用管道
TS-18B20	-55~125	无	1.5 m	

TS-18B20A -55~125 M10X1 1.5m DN15~25
TS-18B20B -55~125 1/2"G 接线盒 DN40~ 60

编辑本段 接线说明

特点 独特的一线接口，只需要一条口线通信 多点能力，简化了分布式温度传感应用 无需外部元件 可用数据总线供电，电压范围为 3.0 V 至 5.5 V 无需备用电源 测量温度范围为-55 ° C 至+125 ° C 。华氏相当于是 67 ° F 到 257 华氏度 -10 ° C 至+85 ° C 范围内精度为±0.5 ° C

温度传感器可编程的分辨率为 9~12 位 温度转换为 12 位数字格式最大值为 750 毫秒 用户可定义的非易失性温度报警设置 应用范围包括恒温控制，工业系统，[消费电子产品](#)温度计，或任何热敏感系统

描述该DS18B20 的数字温度计提供 9 至 12 位（可编程设备温度读数）。由于DS18B20 是一条口线通信，所以[中央微处理器](#)与DS18B20 只有一个一条口线连接。为读写以及温度转换可以从数据线本身获得能量，不需要外接电源。因为每一个DS18B20 的包含一个独特的序号，多个ds18b20s可以同时存在于一条总线。这使得温度传感器放置在许多不同的地方。它的用途很多，包括空调环境控制，感测建筑物内温设备或机器，并进行过程监测和控制。

8 引脚封装 TO-92 封装 用途 描述

5 1 接地 接地

4 2 数字 信号输入输出，一线输出：源极开路

3 3 电源 可选电源管脚。见"寄生功率"一节细节方面。电源必须接地，为行动中，寄生虫功率模式。

不在本表中所有管脚不须接线。

概况框图图 1 显示的主要组成部分DS18B20 的。DS18B20 内部结构主要由四部分组成：64 位光刻ROM、温度传感器、非挥发的温度报警触发器TH和TL、配置寄存器。该装置信号线高的时候，内部电容器 储存能量通由 1 线通信线路给片子供电，而且在低电平期间为片子供电直至下一个高电平的到来重新充电。DS18B20 的电源也可以从外部 3V-5 .5V的电压得到。

DS18B20 采用一线通信接口。因为一线通信接口，必须在先完成 ROM 设定，否则记忆和控制功能将无法使用。主要首先提供以下功能命令之一： 1) 读 ROM， 2) ROM 匹配， 3) 搜索 ROM， 4) 跳过 ROM， 5) 报警检查。这些指令操作作用在没有一个器件的 64 位光刻 ROM 序列号，可以在挂在一线上多个器件选定某一个器件，同时，总线也可以知道总线上挂有多少，什么样的设备。

若指令成功地使 DS18B20 完成温度测量，数据存储在 DS18B20 的存储器。一个控制功能指挥指示 DS18B20 的演出测温。测量结果将被放置在 DS18B20 内存中，并可以让阅读发出记忆功能的指挥，阅读内容的片上存储器。温度报警触发器 TH 和 TL 都有一字节 EEPROM 的数据。如果 DS18B20 不使用报警检查指令，这些寄存器可作为一般的用户记忆用途。在片上还载有配置字节以理想的解决温度数字转换。写 TH,TL 指令以及配置字节利用一个记忆功能的指令完成。通过缓存器读寄存器。所有数据的读，写都是从最低位开始。

DS18B20 有 4 个主要的数据部件：

(1) 光刻ROM中的 64 位序列号是出厂前被光刻好的，它可以看作是该DS18B20 的地址序列码。64 位光刻ROM的排列是：开始 8 位（28H）是产品类型标号，接着的 48 位是该DS18B20 自身的序列号，最后 8 位是前面 56 位的[循环冗余校验码](#)（CRC=X8+X5+X4+1）。光刻ROM的作用是使每一个DS18B20 都各不相同，这样就可以实现一根总线上挂接多个DS18B20 的目的。

(2) DS18B20 中的温度传感器可完成对温度的测量，以 12 位转化为例：用 16 位符号扩展的二进制补码读数形式提供，以 0.0625°C/LSB 形式表达，其中 S 为符号位。

表 1 DS18B20 温度值格式表

4.3.1

DS18B20 的管脚排列如图 4.4 所示。

图 4.4DS18B20 的管脚排列如图

DS18B20 内部结构主要由四部分组成：64 位光刻 ROM，温度传感器，温度报警触发器 TH 和 TL,配置寄存器。DS18B20 内部结构图如图 4.5 所示。

图 4.5 DS18B20 内部结构图

4.3.2 存储器

DS18B20 的存储器包括高速暂存器 RAM 和可电擦除 RAM，可电擦除 RAM 又包括温度触发器 TH 和 TL，以及一个配置寄存器。存储器能完整的确定一线端口的通讯，数字开始用写寄存器的命令写进寄存器，接着也可以用读寄存器的命令来确认这些数字。当确认以后就可以用复制寄存器的命令来将这些数字转移到可电擦除 RAM 中。当修改过寄存器中的数时，这个过程能确保数字的完整性。

高速暂存器 RAM 是由 8 个字节的存储器组成；第一和第二个字节是温度的显示位。第三和第四个字节是复制 TH 和 TL，同时第三和第四个字节的数字可以更新；第五个字节是复制配置寄存器，同时第五个字节的数字可以更新；六、七、八三个字节是计算机自身使用。用读寄存器的命令能读出第九个字节，这个字节是对前面的八个字节进行校验。存储器的结构图如图 4.6 所示。

图 4.6 存储器的结构图

4.3.3 64-位光刻 ROM

64 位光刻 ROM 的前 8 位是 DS18B20 的自身代码，接下来的 48 位为连续的数字代码，最后的 8 位是对前 56 位的 CRC 校验。64-位的光刻 ROM 又包括 5 个 ROM 的功能命令：读 ROM，匹配 ROM，跳跃 ROM，查找 ROM 和报警查找。64-位光刻 ROM 的结构图如图 4.7 所示。

图 4.7 位 64-位光刻 ROM 的结构图

4.3.4 DS18B20 外部电源的连接方式

DS18B20 可以使用外部电源 VDD，也可以使用内部的寄生电源。当 VDD 端口接 3.0V—5.5V 的电压时是使用外部电源；当 VDD 端口接地时使用了内部的寄生电源。无论是内部寄生电源还是外部供电，I/O 口线要接 5KΩ 左右的上拉电阻。连接图如图 4.8、图 4.9 所示。

图 4.8 使用寄生电源的连接图

图 4.9 外接电源的连接图

4.3.4 DS18B20 温度处理过程

4.3.4.1 配置寄存器

配置寄存器是配置不同的位数来确定温度和数字的转化。配置寄存器的结构图如图 4.10 所示。

图 4.10 配置寄存器的结构图

由图 4.9 可以知道 R1, R0 是温度的决定位，由 R1, R0 的不同组合可以配置为 9 位，10 位，11 位，12 位的温度显示。这样就可以知道不同的温度转化位所对应的转化时间，四种配置的分辨率分别为 0.5℃，0.25℃，0.125℃ 和 0.0625℃，出厂时以配置为 12 位。温度的决定配置图如图 8 所示。

图 4.11 温度的决定配置图

4.3.4.2 温度的读取

DS18B20 在出厂时以配置为 12 位，读取温度时共读取 16 位，所以把后 11 位的 2 进制转化为 10 进制后在乘以 0.0625 便为所测的温度，还需要判断正负。前 5 个数字为符号位，当前 5 位为 1 时，读取的温度为负数；当前 5 位为 0 时，读取的温度为正数。16 位数字摆放是从低位到高位，温度的关系图如图 4.12 所示。

图 4.12 为温度的关系图

4.3.4.3. DS18B20 控制方法

DS18B20 有六条控制命令，如表 4.1 所示：

表 4.1 为 DS18B20 有六条控制命令

指令 约定代码 操作说明

温度转换 44H 启动 DS18B20 进行温度转换

读暂存器 BEH 读暂存器 9 位二进制数字

写暂存器 4EH 将数据写入暂存器的 TH、TL 字节

复制暂存器 48H 把暂存器的 TH、TL 字节写到 E2RAM 中

重新调 E2RAM B8H 把 E2RAM 中的 TH、TL 字节写到暂存器 TH、TL 字节

读电源供电方式 B4H 启动 DS18B20 发送电源供电方式的信号给主 CPU

4.3.4.4 DS18B20 的初始化

(1) 先将数据线置高电平“1”。

- (2) 延时（该时间要求的不是很严格，但是尽可能的短一点）
- (3) 数据线拉到低电平“0”。
- (4) 延时 750 微秒（该时间的时间范围可以从 480 到 960 微秒）。
- (5) 数据线拉到高电平“1”。

(6) 延时等待（如果初始化成功则在 15 到 60 微妙时间之内产生一个由 DS18B20 所返回的低电平“0”。据该状态可以来确定它的存在，但是应注意不能无限的进行等待，不然会使程序进入死循环，所以要进行超时控制）。

(7) 若 CPU 读到了数据线上的低电平“0”后，还要做延时，其延时的时间从发出的高电平算起（第（5）步的时间算起）最少要 480 微秒。

- (8) 将数据线再次拉高到高电平“1”后结束。

其时序如图 4.13 所示：

图 4.13 初始化时序图

4.3.4.5 DS18B20 的写操作

- (1) 数据线先置低电平“0”。
- (2) 延时确定的时间为 15 微秒。
- (3) 按从低位到高位顺序发送字节（一次只发送一位）。
- (4) 延时时间为 45 微秒。
- (5) 将数据线拉到高电平。
- (6) 重复上（1）到（6）的操作直到所有的字节全部发送完为止。
- (7) 最后将数据线拉高。

DS18B20 的写操作时序图如图 4.14 所示。

图 4.14 DS18B20 的写操作时序图

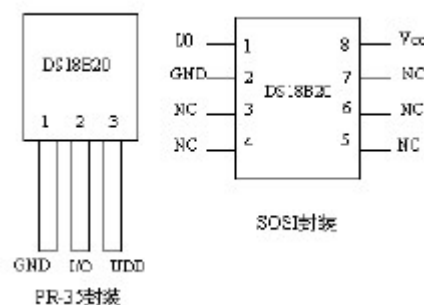
4.3.4.6 DS18B20 的读操作

- (1) 将数据线拉高“1”。
- (2) 延时 2 微秒。
- (3) 将数据线拉低“0”。
- (4) 延时 3 微秒。
- (5) 将数据线拉高“1”。
- (6) 延时 5 微秒。
- (7) 读数据线的状态得到 1 个状态位，并进行数据处理。
- (8) 延时 60 微秒。

DS18B20 的读操作时序图如图 4.15 所示。

图 1.15 DS18B20 的读操作图

数字温度传感器 DS18B20 介绍



1、DS18B20 的主要特性

1.1、适应电压范围更宽，电压范围：3.0~5.5V，在寄生电源方式下可由数据线供电

1.2、独特的单线接口方式，DS18B20 在与微处理器连接时仅需要一条口线即可实现微处理器与 DS18B20 的双向通讯

- 1.3、DS18B20 支持多点组网功能，多个 DS18B20 可以并联在唯一的三线上，实现组网多点测温
- 1.4、DS18B20 在使用中不需要任何外围元件，全部 传感元件及转换电路集成在形如一只三极管的集成电路内
- 1.5、温范围 $-55^{\circ}\text{C}\sim+125^{\circ}\text{C}$ ，在 $-10\sim+85^{\circ}\text{C}$ 时精度为 $\pm 0.5^{\circ}\text{C}$
- 1.6、可编程 的分辨率为 9~12 位，对应的可分辨温度分别为 0.5°C 、 0.25°C 、 0.125°C 和 0.0625°C ，可实现高精度测温
- 1.7、在 9 位分辨率时最多在 93.75ms 内把温度转换为数字，12 位分辨率时最多在 750ms 内把温度值转换为数字，速度更快
- 1.8、测量结果直接输出数字温度信号，以"一线总线"串行传送给 CPU，同时可传送 CRC 校验码，具有极强的抗干扰纠错能力
- 1.9、负压特性：电源极性接反时，芯片不会因发热而烧毁，但不能正常工作。

2、DS18B20 的外形和内部结构

DS18B20 内部结构主要由四部分组成：**64 位光刻 ROM**、温度传感器、非挥发的温度报警触发器 TH 和 TL、配置寄存器。

DS18B20 的外形及管脚排列如下图 1:

DS18B20 引脚定义:

- (1)DQ 为数字信号输入/输出端;
- (2)GND 为电源地;
- (3)VDD 为外接供电电源输入端（在寄生电源接线方式时接地）。

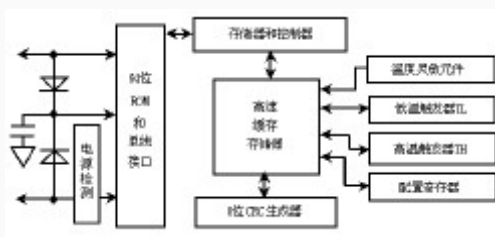
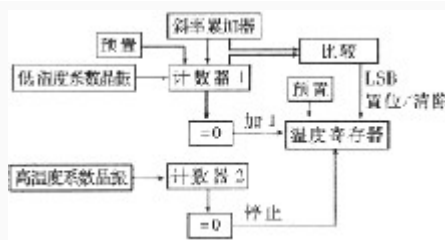


图 2：DS18B20 内部结构图

3、DS18B20 工作原理

DS18B20 的读写时序和测温原理与 DS1820 相同，只是得到的温度值的位数因分辨率不同而不同，且温度转换时的延长时间由 2s 减为 750ms。DS18B20 测温原理如图 3 所示。图中低温度系数晶振的振荡频率受温度影响很小，用于产生固定频率的脉冲信号送给计数器 1。高温温度系数晶振 随温度变化其振荡率明显改变，所产生的信号作为计数器 2 的脉冲输入。计数器 1 和温度寄存器被预置在 -55°C 所对应的一个基数值。计数器 1 对 低温度系数晶振产生的脉冲信号进行减法计数，当计数器 1 的预置值减到 0 时，温度寄存器的值将加 1，计数器 1 的预置将重新被装入，计数器 1 重 新开始对低温度系数晶振产生的脉冲信号进行计数，如此循环直到计数器 2 计数到 0 时，停止温度寄存器值的累加，此时温度寄存器中的数值即为所测温度。图 3 中的斜率累加器用于补偿和修正测温过程中的非线性，其输出用于修正计数器 1 的预置值。

图 3：DS18B20 测温原理框图



DS18B20 有 4 个主要的数据部件:

(1) 光刻 ROM 中的 64 位序列号是出厂前被光刻好的，它可以看作是 该 DS18B20 的地址序列码。64 位光刻 ROM 的排列是：开始 8 位（28H）是产品类型标号，接着的 48 位是该 DS18B20 自身的序列号，最后 8 位是前面 56 位的循环冗余校验码（ $\text{CRC}=\text{X8}+\text{X5}+\text{X4}+1$ ）。光刻 ROM 的作用 是使每一个 DS18B20 都各不相同，这样就可以实现一根总线上挂接多个 DS18B20 的目的。

(2) DS18B20 中的温度传感器可完成对温度的测量，以 12 位转化为例：用 16 位符号扩展的二进制补码读数形式提供，以 0.0625°C/LSB 形式表达，其中 S 为符号位。

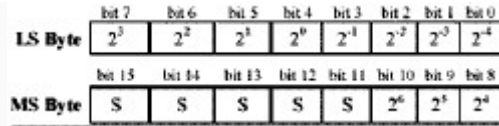


表 1: DS18B20 温度值格式表

这是 12 位转化后得到的 12 位数据，存储在 18B20 的两个 8 比特的 RAM 中，二进制中的前面 5 位是符号位，如果测得的温度大于 0，这 5 位为 0，只要将测到的数值乘于 0.0625 即可得到实际温度；如果温度小于 0，这 5 位为 1，测到的数值需要取反加 1 再乘于 0.0625 即可得到实际温度。例如 +125°C 的数字输出为 07D0H，+25.0625°C 的数字输出为 0191H，-25.0625°C 的数字输出为 FE6FH，-55°C 的数字输出为 FC90H。

表 2: DS18B20 温度数据表

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	0000 0111 1101 0000	07D0h
+85°C*	0000 0101 0101 0000	0550h
+25.0625°C	0000 0001 1001 0001	0191h
+10.125°C	0000 0000 1010 0010	00A2h
-0.5°C	0000 0000 0000 1000	0008h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1000	FF8h
-10.125°C	1111 1111 0101 1110	FF50h
-25.0625°C	1111 1110 0110 1111	FE6fh
-55°C	1111 1100 1001 0000	FC90h

*The power-on reset value of the temperature register is +85°C

(3) DS18B20 温度传感器的存储器 DS18B20 温度传感器的内部存储器包括一个高速暂存 RAM 和一个非易失性的可电擦除的 EEPROM,后者存放高温度和低温触发器 TH、TL 和结构寄存器。

(4) 配置寄存器 该字节各位的意义如下：

表 3: 配置寄存器结构

TM	R1	R0	1	1	1	1	1
----	----	----	---	---	---	---	---

低五位一直都是“1”，TM 是测试模式位，用于设置 DS18B20 在工作模式还是在测试模式。在 DS18B20 出厂时该位被设置为 0，用户不要去改动。R1 和 R0 用来设置分辨率，如下表所示：（DS18B20 出厂时被设置为 12 位）

表 4: 温度分辨率设置表

R1	R0	分辨率	温度最大转换时间
0	0	9 位	93.75ms
0	1	10 位	187.5ms
1	0	11 位	375ms
1	1	12 位	750ms

4、高速暂存存储器高速暂存存储器由 9 个字节组成，其分配如表 5 所示。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。单片机可通过单线接口读到该数据，读取时低位在前，高位在后，数据格式如表 1 所示。对应的温度计算：当符号位 S=0 时，直接将二进制位转换为十进制；当 S=1 时，先将补码变为原码，再计算十进制值。表 2 是对应的一部分温度值。第九个字节是冗余检验字节。

表 5: DS18B20 暂存寄存器分布

寄存器内容	字节地址
温度值低位 (LS Byte)	0
温度值高位 (MS Byte)	1
高温限值 (TH)	2
低温限值 (TL)	3
配置寄存器	4
保留	5

保留	6
保留	7
CRC 校验值	8

根据 DS18B20 的通讯协议，主机（单片机）控制 DS18B20 完成温度转换必须经过三个步骤：每一次读写之前都要对 DS18B20 进行复位操作，复位成功后发送一条 ROM 指令，最后发送 RAM 指令，这样才能对 DS18B20 进行预定的操作。复位要求主 CPU 将数据线下拉 500 微秒，然后释放，当 DS18B20 收到信号后等待 16~60 微秒左右，后发出 60~240 微秒的存在低脉冲，主 CPU 收到此信号表示复位成功。

表 6：ROM 指令表

指令	约定代码	功能
读 ROM	33H	读 DS1820 温度传感器 ROM 中的编码（即 64 位地址）
符合 ROM	55H	发出此命令之后，接着发出 64 位 ROM 编码，访问单总线上与该编码相对应的 DS1820 使之作出响应，为下一步对该 DS1820 的读写作准备。
搜索 ROM	0FOH	用于确定挂接在同一总线上 DS1820 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 DS1820 发温度变换命令。适用于单片工作。
告警搜索命令	0ECH	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。

表 6：RAM 指令表

指令	约定代码	功能
温度变换	44H	启动 DS1820 进行温度转换，12 位转换时最长为 750ms（9 位为 93.75ms）。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 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	将 EEPROM 中内容恢复到 RAM 中的第 3、4 字节。
读供电方式	0B4H	读 DS1820 的供电模式。寄生供电时 DS1820 发送“0”，外接电源供电 DS1820 发送“1”。

5、DS18B20 的应用电路 DS18B20 测温系统具有测温系统简单、测温精度高、连接方便、占用口线少等优点。下面就是 DS18B20 几个不同应用方式下的测温电路图：

5.1、DS18B20 寄生电源供电方式电路图如下面图 4 所示，在寄生电源供电方式下，DS18B20 从单线信号线上汲取能量：在信号线 DQ 处于高电平期间把能量储存在内部电容里，在信号线处于低电平期间消耗电容上的电能工作，直到高电平到来再给寄生电源（电容）充电。

独特的寄生电源方式有三个好处：

- 1) 进行远距离测温时，无需本地电源
- 2) 可以在没有常规电源的条件下读取 ROM
- 3) 电路更加简洁，仅用一根 I/O 口实现测温

要想使 DS18B20 进行精确的温度转换，I/O 线必须保证在温度转换期间提供足够的能量，由于每个 DS18B20 在温度转换期间工作电流达到 1mA，当几个温度传感器挂在一根 I/O 线上进行多点测温时，只靠 4.7K 上拉电阻就无法提供足够的能量，会造成无法转换温度或温度误差极大。

因此，图 4 电路只适应于单一温度传感器测温情况下使用，不适宜采用电池供电系统中。并且工作电源 VCC 必须保证在 5V，当电源电压下降时，寄生电源能够汲取的能量也降低，会使温度误差变大。

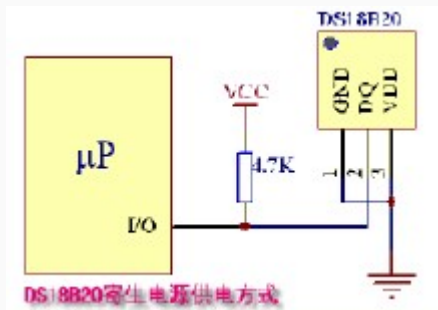


图 4

5.2、DS18B20 寄生电源强上拉供电方式电路图改进的寄生电源供电方式如下面图 5 所示，为了使 DS18B20 在动态转换周期中获得足够的电流供应，当进行温度转换或拷贝到 E2 存储器操作时，用 MOSFET 把 I/O 线直接拉到 VCC 就可提供足够的电流，在发出任何涉及到拷贝到 E2 存储器或启动温度转换的指令后，必须在最多 10 μ S 内把 I/O 线转换到强上拉状态。在强上拉方式下可以解决电流供应不走的问题，因此也适合于多点测温应用，缺点就是要多占用一根 I/O 口线进行强上拉切换。

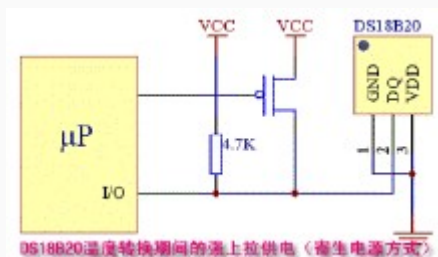


图 5

图 5

注意：在图 4 和图 5 寄生电源供电方式中，DS18B20 的 VDD 引脚必须接地

5.3、DS18B20 的外部电源供电方式

在外部电源供电方式下，DS18B20 工作电源由 VDD 引脚接入，此时 I/O 线不需要强上拉，不存在电源电流不足的问题，可以保证转换精度，同时在总线上理论可以挂接任意多个 DS18B20 传感器，组成多点测温系统。注意：在外部供电的方式下，DS18B20 的 GND 引脚不能悬空，否则不能转换温度，读取的温度总是 85 $^{\circ}$ C。

图 6：外部供电方式单点测温电路

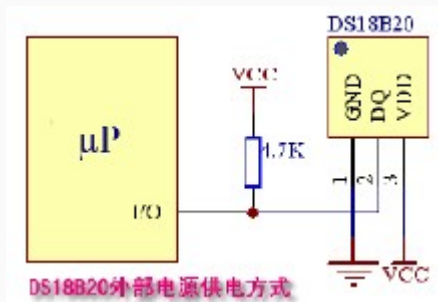


图 6

图 7：外部供电方式的多点测温电路图

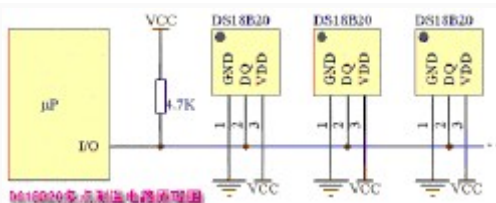


图 7

外部电源供电方式是 DS18B20 最佳的工作方式，工作稳定可靠，抗干扰能力强，而且电路也比较简单，可以开发出稳定可靠的多点温度监控系统。站长推荐大家在开发中使用外部电源供电方式，毕竟比寄生电源方式只多接一根 VCC 引线。

在外接电源方式下，可以充分发挥 DS18B20 宽电源电压范围的优点，即使电源电压 VCC 降到 3V 时，依然能够保证温度测量精度。

6、DS1820 使用中注意事项

DS1820 虽然具有测温系统简单、测温精度高、连接方便、占用口线少等优点，但在实际应用中也应注意以下几方面的问题：

6.1、较小的硬件开销需要相对复杂的软件进行补偿，由于 DS1820 与微处理器间采用串行数据传送，因此，在对 DS1820 进行读写编程时，必须严格的保证读写时序，否则将无法读取测温结果。在使用 PL/M、C 等高级语言进行系统程序设计时，对 DS1820 操作部分最好采用汇编语言实现。

6.2、在 DS1820 的有关资料中均未提及单总线上所挂 DS1820 数量问题，容易使人误认为可以挂任意多个 DS1820，在实际应用中并非如此。当单总线上所挂 DS1820 超过 8 个时，就需要解决微处理器的总线驱动问题，这一点在进行多点测温系统设计时要加以注意。

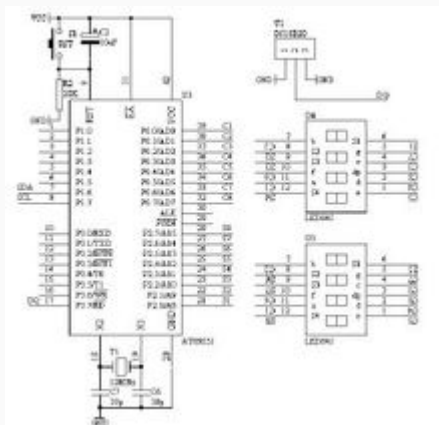
6.3、连接 DS1820 的总线电缆是有长度限制的。试验中，当采用普通信号电缆传输长度超过 50m 时，读取的测温数据将发生错误。当将总线电缆改为双绞线带屏蔽电缆时，正常通讯距离可达 150m，当采用每米绞合次数更多的双绞线带屏蔽电缆时，正常通讯距离进一步加长。这种情况主要是由总线分布电容使信号波形产生畸变造成的。因此，在用 DS1820 进行长距离测温系统设计时要充分考虑总线分布电容和阻抗匹配问题。

6.4、在 DS1820 测温程序设计中，向 DS1820 发出温度转换命令后，程序总要等待 DS1820 的返回信号，一旦某个 DS1820 接触不好或断线，当程序读该 DS1820 时，将没有返回信号，程序进入死循环。这一点在进行 DS1820 硬件连接和软件设计时也要给予一定的重视。测温电缆线建议采用屏蔽 4 芯双绞线，其中一对线接地线与信号线，另一组接 VCC 和地线，屏蔽层在源端单点接地。

应用举例及程序代码段

利用 DS18B20 构成的数字温度计

用一片 DS18B20 构成测温系统，测量的温度精度达到 0.1 度，测量的温度的范围在 -20 度到 +100 度之间，用 8 位



数码管显示出来

读取温度程序代码段

```
#include "reg51.h"
#include "d1820.h"

int t;
uint num;
uchar dat; //读写数据变量
uchar a=0;
uchar b=0;
float tep=0; //读一个温度时的温度转换中间间
uchar data tempbuff[4]=0; //温度字型显示中间变量
```

/******

函数名称: delay(uint num)

返回值: 无

参数: uint num 颜色数据值

作用: 延时

*****/

```
void delay(uint num)
```

```
{
```

```
while(num--);
```

```
}
```

/******

函数名称: void Init_DS18B20(void)

返回值: 无

参数: 无

作用: 初始化 18B20

*****/

```
void Init_DS18B20(void)
```

```
{
```

```
char x=0;
```

```
DQ=1;
```

```
delay(10);//稍作延时
```

```
DQ=0;
```

```
delay(80);//延时>480us 540us
```

```
DQ=1; //拉高总线 15-60us
```

```
delay(20);
```

```
x=DQ;//读总线状态 为 0 复位成功, 为 1 则不成功
```

```
delay(30);
```

```
DQ=1;//释放总线
```

```
}
```

/******

函数名称: uchar ReadOneChar(void)

返回值: uchar dat

参数: 无

作用: 读 1820 一个字节

*****/

```
uchar ReadOneChar(void)
```

```
{
```

```
uchar i;
```

```
uchar dat=0;
```

```
for(i=0;i<8;i++)
```

```
{
```

```
DQ=0;
```

```
dat>>=1;
```

```
DQ=1;//给脉冲
```

```
if(DQ) {dat|=0x80;}//读 1 /// 读 0 右移处理
```

```
delay(8);//15us 内读完一个数
```

```
}
```

```
return(dat);
```

```
}
```

```
/******
```

函数名称: void WriteOneChar(uchar dat)

返回值: 无

参数: uchar dat

作用: 写 1820 一个字节

```
*****/
```

```
////**写 DS18B20**//写 0 60us 读完, 写 1 30us 内读完
```

```
void WriteOneChar(uchar dat)
```

```
{
```

```
uchar i=0;
```

```
for(i=0;i<8;i++)
```

```
{
```

```
DQ=0;
```

```
DQ=dat&0x01;//写所给数据最低位
```

```
delay(10); //////////////
```

```
DQ=1;//给脉冲
```

```
dat>>=1;
```

```
}
```

```
delay(8);
```

```
}
```

```
/******
```

函数名称: int ReadOneTemperature(void)

返回值: int t

参数: 无

作用: 读温度值

```
*****/
```

```
int ReadOneTemperature(void) //****//**读取温度值*****// 每次读写均要先复位
```

```
{
```

```
Init_DS18B20();
```

```
WriteOneChar(0xcc);//发跳过 ROM 命令
```

```
WriteOneChar(0x44);//发读开始转换命令
```

```
Init_DS18B20();
```

```
WriteOneChar(0xcc);//发跳过 ROM 命令
```

```
WriteOneChar(0xbe);//读寄存器, 共九字节, 前两字节为转换值
```

```
a=ReadOneChar(); //a 存低字节
```

```
b=ReadOneChar(); //b 存高字节
```

```
t=b;
```

```
t<<=8;//高字节转换为 10 进制
```

```
t=t|a;
```

```
tep=t*0.0625;//转换精度为 0.0625/LSB
```

```
t=tep*10+0.5;//保留 1 位小数并四舍五入****后面除 10 还原正确温度值)
```

```
return(t);
```

```
}
```

```
/******
```

函数名称: uint Temperaturepro(void)

返回值: void

参数: void

作用: 温度处理

```
*****/
```

```
void Temperaturepro(void)
```

```
{
```

```
int temp;
```

```
temp=ReadOneTemperature();
```

```
tempbuf[3]=temp/1000; //百位
```

```
tempbuf[2]=temp/100%10; //十位
```

```
tempbuf[1]=temp%100/10; //个位
```

```
tempbuf[0]=temp%10; //小数
```

```
}
```