DS18B20 概述

1）DS18B20 是 DALLAS 公司生产的一线制数字温度传感器；

2）具有 3 引脚 TO－92 小体积封装形式；

3）温度测量范围为－55℃～＋125℃；

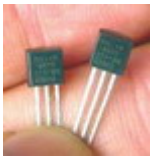
4）电源供电范围为 3V~5.5V ；

5）可编程为 9 位～12 位数字表示；

6）测温分辨率可达 0.0625℃，被测温度用符号扩展的 16 位数字量方式串行输出；

7）其工作电源既可在远端引入，也可采用寄生电源方式产生；

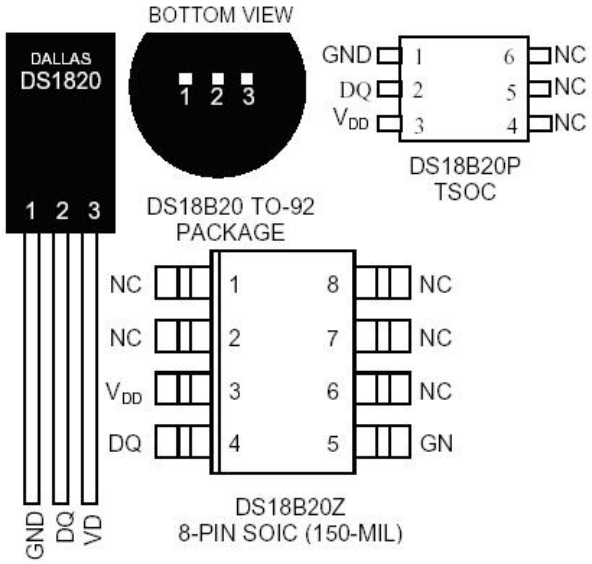
8）多个 DS18B20 可以并联到 3 根（VDD、DQ 和 GND）或 2 根（利用 DQ 线供电、GND）线上，CPU 只需一根端口线就能与总线上的多个串联的 DS18B20 通信，占用微处理器的端口较少，可节省大量的引线和逻辑电路。一线总线独特而且经济的特点，使用户可轻松地组建传感器网络，为测量系统的构建引入全新概念。



TO-92 封装的 DS18B20

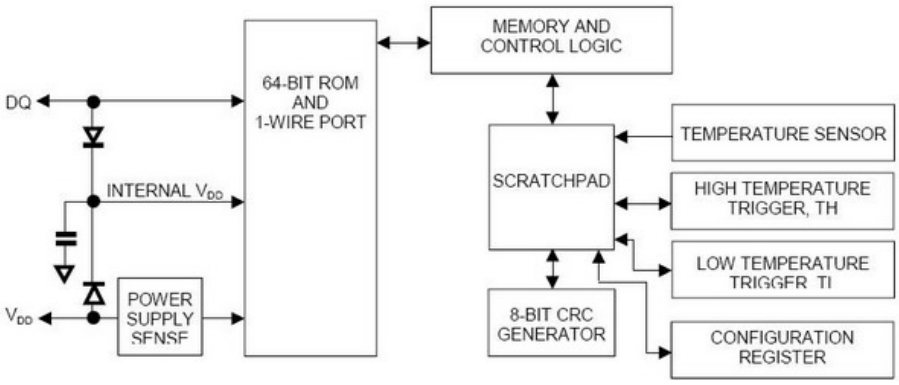
DS18B20 管脚

DS18B20 的管脚排列及不同封装形式如图 2 所示，DQ 为数字信号输入/输出端；GND 为电源地；VDD 为外接供电电源输入端（在寄生电源接线方式时接地，见），NC 表示无连接。



管脚图

DS18B20 内部结构如图 3 所示，主要由 4 部分组成：64 位 ROM、温度传感器、非易失性存储的温度报警触发器 TH 和 TL、配置寄存器。

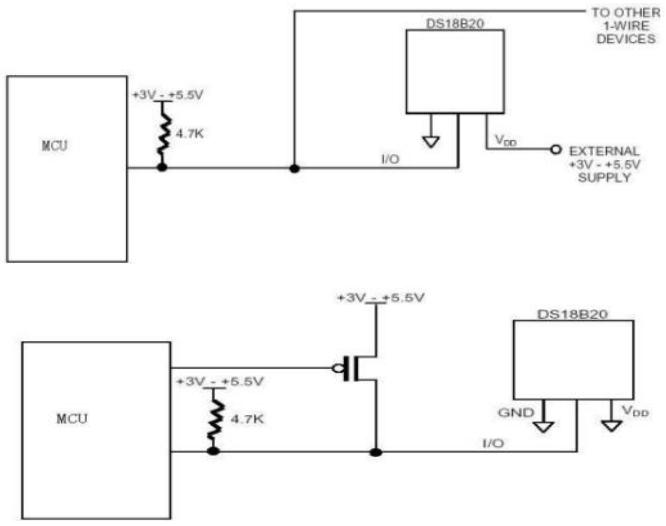


DS18B20 内部结构图

应用领域

- 非常适用于远距离多点温度检测系统。

连接图



DQ-数据输入输出。漏极开路 1 线接口。也在寄生电源模式时给设备提供电源。

DS18B20 读写

访问 DS18B20 的顺序如理

- 初始化；

Tel：010-62669059
Website：www.61mcu.com
E-mail：fae_61mcu@163.com

- ROM 命令（接着是任何需要的数据交换）；
- DS18B20 函数命令（接着是任何需要的数据交换）。

每一次访问 DS18B20 时必须遵循这一顺序，如果其中的任何一步缺少或打乱它们的顺序，DS18B20 将不会响应。

（1）初始化时序

所有与 DS18B20 的通信首先必须初始化：控制器发出复位脉冲，DS18B20 以存在脉冲响应。在图 14-9 中给出了描述。当 DS18B20 发出存在脉冲对复位响应时，它指示控制器该 DS18B20 已经在总线上并准备好操作。

（2）读/写时序

控制器在写时序写数据到 DS18B20，在读时序从 DS18B20 中读数据。每一总线时序传送 1 位数据。

读/写时序图见图 14-10。

（3）写流程时序

有两种类型的写时序：写 1 时序和写 0 时序。控制器用写“1”时序写逻辑“1”到 DS18B20，用写“0”时序写逻辑“0”到 DS18B20。

所有写时序必须持续最少 60μs，每个写时序之间必须有至少 1μs 的恢复时间。两种类型的写时序都从控制器把总线拉低开始。为产生写“1”时序，在将总线拉低后，总线控制器必须在 15μs 内释放总线。

总线释放后，5K 的上拉电阻将总线电平抬高。为产生写“0”时序，在将总线拉低后，控制器在整个时序内必须持续控制总线为低电平（至少 60μs）。DS18B20 在控制器发出写时序后的 15~60μs 的时间窗口内采样总线。如果在采样窗口期间总线为高，“1”就被写入 DS18B20；如果在采样窗口期间总线为低，“0”就被写入 DS18B20。

（4）读时序

当控制器发出读时序时，DS18B20 可以发送数据到控制器。所有读时序必须持续最少 60μs，每个读时序之间必须有至少 1μs 的恢复时间。读时序从控制设备将总线拉低至少 1μs 后释放总线开始。控制器启动读时序后，DS18B20 开始在总线上传送“1”或“0”时，DS18B20 通过保持总线为高发送“1”，将总线拉低发送“0”发送“0”时，DS18B20 在时序结束时释放总线，总线被上拉电阻拉回到高电平空闲状态。

从 DS18B20 输出的数据在启动读时序的下降沿后 15μs 内有效。因此，控制器必须在时序开始的 15μs 内释放总线然后采样总线状态。通过读写时序，控制器可以发出控制命令，对 DS18B20 进行读写操作。

常用命令

（1） SKIP ROM [CCH]

控制器可以用这一命令同时访问总线上的所有设备而不需要送出 ROM 序列码信息。例如：发出 Skip ROM 命令后接着送出 Convert T [44H] 命令,控制器可以使总线上的所有 DS18B20 同时进行温度转换。请注意，仅当总线上只有一个从属设备时，Skip ROM 命令后才可跟着 Read Scratchpad [BEH] 命令。如果总线上有多于一个的从属设备，由于多设备企图同时送出数据，将引起数据冲突。

（2） SEARCH ROM [F0H]

当系统开始上电时，控制器必须识别总线上所有从机的 ROM 序列码，以确定从机的数目和它们的类别。控制器需要执行 Search ROM 循环（如 Search ROM 命令后接着数据交换）足够多次以识别所有的从属设备。如果只有一个从属设备在总线上，

可使用简单的 ReadROM 命令取代 Search ROM。每一个 Search ROM 命令之后必须返回到事务序列的步骤 1（初始化）。

（3） READ ROM [33H]

这一命令在总线上只有一个从属设备时使用。它使得控制器可以不用 Search ROM 命令就可读出从机的 64 位 ROM 序列码。当多于一个从属设备在总线上时，如果使用该命令，由于所有设备都企图响应该命令将产生数据冲突。

（4） CONVERT T [44H]

这一命令开始一次温度转换。变换结束后，数据保存在暂存器的 2 字节温度寄存器中，DS18B20 回到低功耗空闲状态。如果设备工作在寄生电源模式，则这一命令发出后 10μs 之内，在整个变换期间（tconv）控制器必须在总线上能够有强的上拉。如果 DS18B20 由外部电源供电，那么 Convert T 命令之后控制器可以发出读时序。如果温度变换正在进行，那么 DS18B20 返回“0”；如果已经完毕，则返回“1”。在寄生供电模式下，不能使用这一技术，因为在变换期间总线被抬高。

（5） WRITE SCRATCHPAD [4EH]

这一命令使得控制器可以写 3 字节数据到 DS18B20 的暂存器中。第一字节数据写到 TH 寄存器（暂存器的字节 2），第 2 字节写到 TL 寄存器（字节 3），第 3 字节写到配置寄存器（字节 4）。数据以最低有效位先发送。所有 3 字节必须在控制器发出复位或数据可能丢失之前写完。

（6） READ SCRTCHPAD [BEH]

这一命令使控制器可以读暂存器的内容。数据传送开始于字节 0 的最低位，直到暂存器的第 9 字节（字节 8CRC）被读取。任何时候，如果只需部分暂存器数据，控制器可以使用复位结束读操作。

代码

温度传感器 ds1820 的 51 单片机汇编程序(不保证程序从编译器 copy 过程中的准确无误性)

```
TEMPER_L    EQU    36H
TEMPER_H    EQU    35H
TEMPER_NUM  EQU    60H
FLAG1       BIT    00H
DQ          BIT    P3.3

AAA: MOV    SP,#70H
     LCALL GET_TEMPER
     LCALL TEMPER_COV
     LJMP AAA
     NOP

     ;------------------读出转换后的温度值
GET_TEMPER:
     SETB DQ ; 定时入口
     BCD: LCALL INIT_1820
     JB FLAG1,S22
```


Tel：010-62669059
Website：www.61mcu.com
E-mail：fae_61mcu@163.com

```
        LJMP BCD ; 若 DS18B20 不存在则返回

S22: LCALL DELAY1
        MOV A,#0CCH ; 跳过 ROM 匹配------0CC
        LCALL WRITE_1820
        MOV A,#44H ; 发出温度转换命令
        LCALL WRITE_1820
        NOP
        LCALL DELAY
        LCALL DELAY
        CBA: LCALL INIT_1820
        JB FLAG1,ABC
        LJMP CBA

ABC: LCALL DELAY1
        MOV A,#0CCH ; 跳过 ROM 匹配
        LCALL WRITE_1820
        MOV A,#0BEH ; 发出读温度命令
        LCALL WRITE_1820
        LCALL READ_18200 ;READ_1820
        RET

        ;------------------读 DS18B20 的程序,从 DS18B20 中读出一个字节的数据
READ_1820:
        MOV R2,#8
RE1:
        CLR C
        SETB DQ
        NOP
        NOP
        CLR DQ
        NOP
        NOP
        NOP
        SETB DQ
        MOV R3,#7
        DJNZ R3,$
        MOV C,DQ
        MOV R3,#23
        DJNZ R3,$
        RRC A
        DJNZ R2,RE1
```

```
        RET

        ;--------------------写 DS18B20 的程序
WRITE_1820:
        MOV R2,#8
        CLR C
WR1:
        CLR DQ
        MOV R3,#6
        DJNZ R3,$
        RRC A
        MOV DQ,C
        MOV R3,#23
        DJNZ R3,$
        SETB DQ
        NOP
        DJNZ R2,WR1
        SETB DQ
        RET
        ;--------------------读 DS18B20 的程序,从 DS18B20 中读出两个字节的温度数据
READ_18200:
        MOV R4,#2 ; 将温度高位和低位从 DS18B20 中读出
        MOV R1,#36H ; 低位存入 36H(TEMPER_L),高位存入 35H(TEMPER_H)
RE00:
        MOV R2,#8
RE01:
        CLR C
        SETB DQ
        NOP
        NOP
        CLR DQ
        NOP
        NOP
        NOP
        SETB DQ
        MOV R3,#7
        DJNZ R3,$
        MOV C,DQ
        MOV R3,#23
        DJNZ R3,$
        RRC A
        DJNZ R2,RE01
```


Tel：010-62669059
Website：www.61mcu.com
E-mail：fae_61mcu@163.com

```
        MOV @R1,A
        DEC R1
        DJNZ R4,RE00
        RET
;-------------------将从 DS18B20 中读出的温度数据进行转换
TEMPER_COV:
        MOV A,#0F0H
        ANL A,TEMPER_L ; 舍去温度低位中小数点后的四位温度数值
        SWAP A
        MOV TEMPER_NUM,A
        MOV A,TEMPER_L
        JNB ACC.3,TEMPER_COV1 ; 四舍五入去温度值
        INC TEMPER_NUM
TEMPER_COV1:
        MOV A,TEMPER_H
        ANL A,#07H
        SWAP A
        ORL A,TEMPER_NUM
        MOV TEMPER_NUM,A ; 保存变换后的温度数据
        LCALL BIN_BCD
        RET
        ;-------------------将 16 进制的温度数据转换成压缩 BCD 码
BIN_BCD:
        MOV DPTR,#TEMP_TAB
        MOV A,TEMPER_NUM
        MOVC A,@A+DPTR
        MOV TEMPER_NUM,A
        RET
TEMP_TAB:
        DB 00H,01H,02H,03H,04H,05H,06H,07H
        DB 08H,09H,10H,11H,12H,13H,14H,15H
        DB 16H,17H,18H,19H,20H,21H,22H,23H
        DB 24H,25H,26H,27H,28H,29H,30H,31H
        DB 32H,33H,34H,35H,36H,37H,38H,39H
        DB 40H,41H,42H,43H,44H,45H,46H,47H
        DB 48H,49H,50H,51H,52H,53H,54H,55H
        DB 56H,57H,58H,59H,60H,61H,62H,63H
        DB 64H,65H,66H,67H,68H,69H,70H,71H
        DB 72H,73H,74H,75H,76H,77H,78H,79H
        DB 80H,81H,82H,83H,84H,85H,86H,87H
        DB 88H,89H,90H,91H,92H,93H,94H,95H
        DB 96H,97H,98H,99H
```

```
;-------------------DS18B20 初始化程序
INIT_1820:
    SETB DQ
    NOP
    CLR DQ
    MOV R0,#80H
TSR1:
    DJNZ R0,TSR1 ; 延时
    SETB DQ
    MOV R0,#25H ;96US-25H
TSR2:
    DJNZ R0,TSR2
    JNB DQ,TSR3
    LJMP TSR4 ; 延时
TSR3:
    SETB FLAG1 ; 置标志位,表示 DS1820 存在
    LJMP TSR5
TSR4:
    CLR FLAG1 ; 清标志位,表示 DS1820 不存在
    LJMP TSR7
TSR5:
    MOV R0,#06BH ;200US
    TSR6:
    DJNZ R0,TSR6 ; 延时
    TSR7:
    SETB DQ
    RET
    ;-------------------重新写 DS18B20 暂存存储器设定值
RE_CONFIG:
    JB FLAG1,RE_CONFIG1 ; 若 DS18B20 存在,转 RE_CONFIG1
    RET
RE_CONFIG1:
    MOV A,#0CCH ; 发 SKIP ROM 命令
    LCALL WRITE_1820
    MOV A,#4EH ; 发写暂存存储器命令
    LCALL WRITE_1820
    MOV A,#00H ; TH(报警上限)中写入 00H
    LCALL WRITE_1820
    MOV A,#00H ; TL(报警下限)中写入 00H
    LCALL WRITE_1820
    MOV A,#7FH ; 选择 12 位温度分辨率
    LCALL WRITE_1820
```


Tel：010-62669059
Website：www.61mcu.com
E-mail：fae_61mcu@163.com

```
        RET
        ;------------------延时子程序
        DELAY: MOV R7,#00H
        MIN: DJNZ R7,YS500
        RET
    YS500: LCALL YS500US
        LJMP MIN
        YS500US:MOV R6,#00H
        DJNZ R6,$
        RET
        DELAY1: MOV R7,#20H
        DJNZ R7,$
        RET
```