DS18B20 资料整理私货

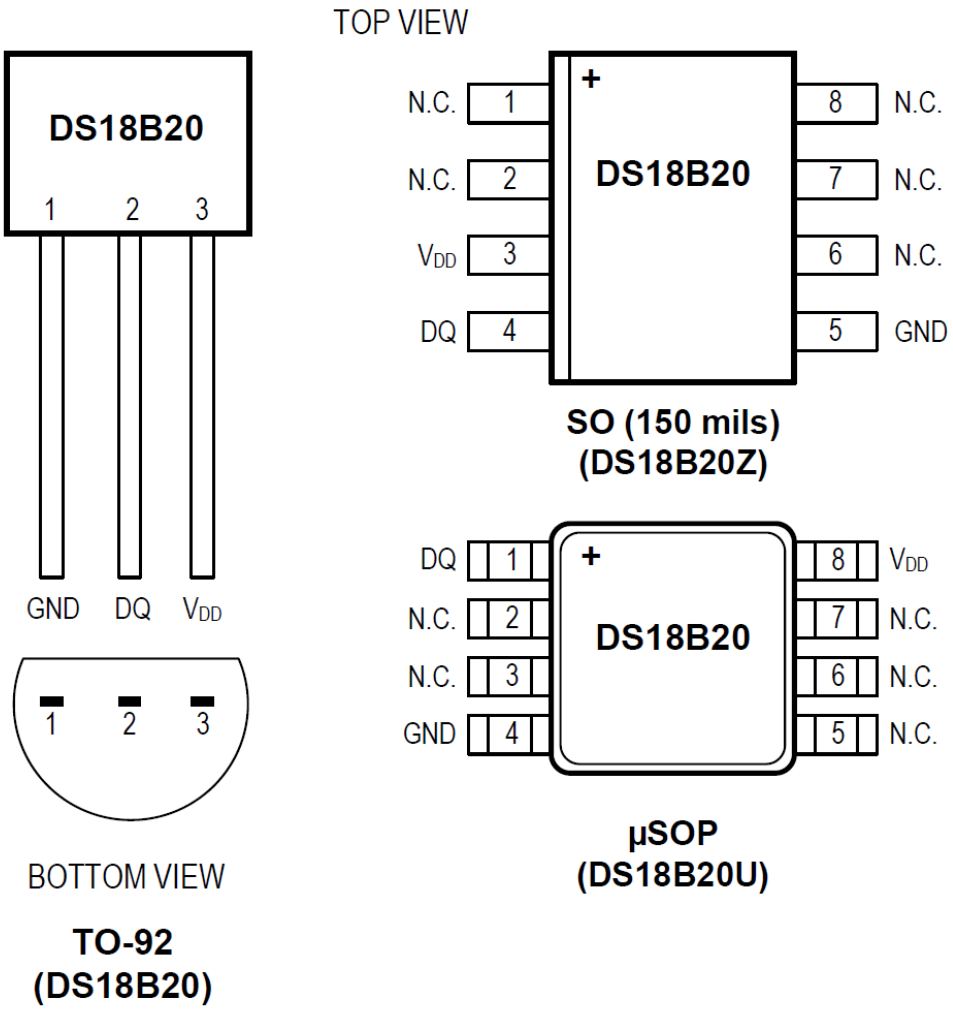
Version	Date
1.0	2018-05

说明：本文档主要参考互联网资料、《DS18B20 官方文档(2015-01 修订版)》、杜洋先生的《18B20 **温度传感器应用解析 V2.0**》，野火秉火提供的 DS18B20 示例代码与视频 进行整理关于温度获取方面相关资料，内容仅供参考且不保证绝对无误，错了你也打不了我，欧力给~^^

特性

- 单线接口实现双向通信
- DS18B20 数字温度计提供 9 至 12 位摄氏温度测量，分别以 0.5℃，0.25℃，0.125℃和 0.0625℃增量递增。
- 在上电状态下默认的精度为 12 位（所以最后获取的数据要乘以 0.0625 得到实际温度）
- 测量-55°C 至+ 125°C 的温度（-67°F 至+ 257°F）
- -10°C 至+ 85°C 范围内时，精度为±0.5°C
- 每个设备都有在板载 ROM 中存储一个唯一的 64 位序列码
- 待补充。。。。。

Pin 口


TOP VIEW
DS18B20
1 2 3
GND DQ VDD
1 2 3
BOTTOM VIEW
TO-92
(DS18B20)
DS18B20
N.C. 1
N.C. 2
VDD 3
DQ 4
8 N.C.
7 N.C.
6 N.C.
5 GND
SO (150 mils)
(DS18B20Z)
DS18B20
DQ 1
N.C. 2
N.C. 3
GND 4
8 VDD
7 N.C.
6 N.C.
5 N.C.
µSOP
(DS18B20U)

ROM & RAM

64 位 ROM 代码

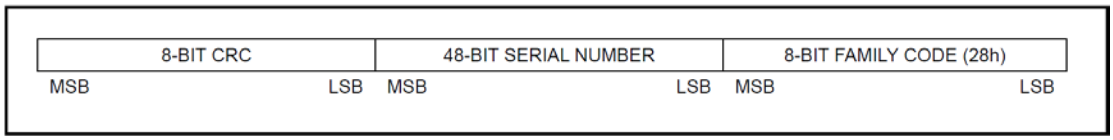
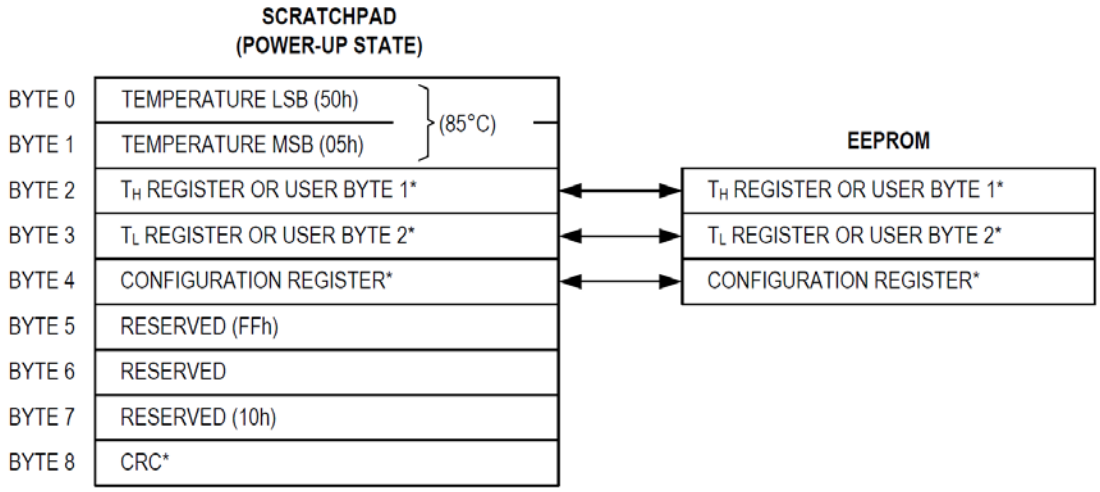


Figure 8. 64-Bit Lasered ROM Code

ROM 代码的最低 8 位包含 DS18B20 的单总线家族代码：28h。 接下来的 48 位包含一个唯一的序列号。 最重要的 8 位包含循环冗余校验（CRC）字节，该字节从 ROM 代码的前 56 位计算得出。 CRC 生成部分提供了 CRC 位的详细说明。 64 位 ROM 代码和相关的 ROM 功能控制逻辑允许 DS18B20 作为 1-Wire 器件使用 1-Wire 总线系统部分详述的协议运行。

RAM 结构



*POWER-UP STATE DEPENDS ON VALUE(S) STORED IN EEPROM.

共 9 个字节;
Byte 0-1 ： 存储温度。分别是温度的低 8 位、高 8 位
Byte 2-3 ： 用户 EEPROM（常用于温度报警值储存）的镜像
Byte 4 ： 用户第 3 个 EEPROM 的镜像。
Byte 5-7 ： 计数寄存器，内部温度转换、计算的暂存单元
Byte 8 ： 前 8 个字节的 CRC 码

DS28B20 芯片 ROM 指令

Read ROM（读 ROM）**[33H]** （方括号中的为 16 进制的命令字）

这个命令允许总线控制器读到 DS18B20 的 64 位 ROM。只有当总线上只存在一个

DS18B20 的时候才可以使用此指令，如果挂接不只一个，当通信时将会发生数据冲突。

Match ROM（指定匹配芯片）**[55H]**

这个指令后面紧跟着由控制器发出了 64 位序列号，当总线上有多只 DS18B20 时，只有与控制发出的序列号相同的芯片才可以做出反应，其它芯片将等待下一次复位。这条指令适应单芯片和多芯片挂接。

Skip ROM（跳跃 ROM 指令）**[CCH]**

这条指令使芯片不对 ROM 编码做出反应，在单总线的情况之下，为了节省时间则可以选用此指令。如果在多芯片挂接时使用此指令将会出现数据冲突，导致错误出现。

Search ROM（搜索芯片）**[F0H]**

在芯片初始化后，搜索指令允许总线上挂接多芯片时用排除法识别所有器件的 64 位 ROM。

Alarm Search（报警芯片搜索）**[ECH]**

在多芯片挂接的情况下，报警芯片搜索指令只对附合温度高于 TH 或小于 TL 报警条件的芯片做出反应。只要芯片不掉电，报警状态将被保持，直到再一次测得温度达不到报警条件为止。

DS28B20 芯片存储器操作指令

Write Scratchpad （向 RAM 中写数据）**[4EH]**

这是向 RAM 中写入数据的指令，随后写入的两个字节的数据将会被存到地址 2（报警 RAM 之 TH）和地址 3（报警 RAM 之 TL）。写入过程中可以用复位信号中止写入。

Read Scratchpad （从 RAM 中读数据）**[BEH]**

此指令将从 RAM 中读数据，读地址从地址 0 开始，一直可以读到地址 9，完成整个 RAM 数据的读出。芯片允许在读过程中用复位信号中止读取，即可以不读后面不需要的字节以减少读取时间。

Copy Scratchpad （将 RAM 数据复制到 EEPROM 中）**[48H]**

此指令将 RAM 中的数据存入 EEPROM 中，以使数据掉电不丢失。此后由于芯片忙于 EEPROM 储存处理，当控制器发一个读时间隙时，总线上输出“0”，当储存工作完成时，总线将输出“1”。在寄生工作方式时必须在发出此指令后立刻超用强上拉并至少保持 10MS，来维持芯片工作。

Convert T（温度转换）**[44H]**

收到此指令后芯片将进行一次温度转换，将转换的温度值放入 RAM 的第 1、2 地址。此后由于芯片忙于温度转换处理，当控制器发一个读时间隙时，总线上输出“0”，当储存工作完成时，总线将输出“1”。在寄生工作方式时必须在发出此指令后立刻超用强上拉并至少保持 500MS，来维持芯片工作。

Recall EEPROM（将 EEPROM 中的报警值复制到 RAM）**[B8H]**

此指令将 EEPROM 中的报警值复制到 RAM 中的第 3、4 个字节里。由于芯片忙于复制处理，当控制器发一个读时间隙时，总线上输出“0”，当储存工作完成时，总线将输出“1”。另外，此指令将在芯片上电复位时将被自动执行。这样 RAM 中的两个报警字节位将始终为 EEPROM 中数据的镜像。

Read Power Supply（工作方式切换）**[B4H]**

此指令发出后发出读时间隙，芯片会返回它的电源状态字，“0”为寄生电源状态，“1”为外部电源状态。

DS18B20 功能命令集

命令	描述	命令号	写入后	备注
Convert T	启动温度转换	44h	DS18B20 将转换状态传送给主机（不适用于寄生供电的 DS18B20）。	1
Read Scratchpad	读取包含 CRC 字节的整个暂存器。	BEh	DS18B20 最多可将 9 个数据字节传输至主设备。	2
Write Scratchpad	将数据写入暂存器字节 2，3 和 4（TH，TL 和配置寄存器）。	4Eh	主机发送 3 个数据字节到 DS18B20。	3
Copy Scratchpad	将 TH，TL 和配置寄存器数据从暂存器复制到 EEPROM。	48h	无。	1
Recall E^2	从 EEPROM 中恢复 TH，TL 和配置寄存器数据到暂存器。	B8h	DS18B20 将调用状态发送给主设备。	
Read Power Supply	向主设备发送 DS18B20 供电模式信号。	B4h	DS18B20 将电源状态传输给主设备。	

注 1：对于由寄生供电的 DS18B20，主控制器必须在温度转换期间在 1-Wire 总线上启用强上拉，并从暂存器复制到 EEPROM。 在这段时间没有其他总线活动发生。
注 2：主机可以随时通过发出复位来中断数据传输。
注 3：发出复位前，这三个字节必须都被写入。

温度数据存储

格式

低 8 位

|----------整数----------||---------- 小数 ----------|

2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}

高 8 位

|------符号位：全为 0➔负；全为 1➔正-------||----------整数----------|

S	S	S	S	S	2^6	2^5	2^4

S = 符号(SIGN)

计算公式

存储温度是 16 位含符号位的二进制补码形式存储的。
温度 = （16 位数据补码）* 0.0625

例:

① 输出二进制值为 0B1111 1110 0110 1111，可以判断出该值为负数，算得其原码（相当于其补码）为 0B1000 0001 1001 0001 即十进制数-401，十进制数乘以 0.0625，合计温度为-25.0625℃。(在编程时注意不要用无符号类型去取反，最高符号位会被忽略)

② 输出二进制值为 0B0000 0001 1001 0001，可以判断出该值为正数，算的其原码（相当于其自身）即十进制数 401，十进制数乘以 0.0625，合计温度为 25.0625℃。

温度（℃）	输出（二进制）	输出（八进制）
+125	0000 0111 1101 0000	07D0h
+85	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	000h

-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

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

（*温度寄存器的上电复位值为+ 85°C。）

报警寄存器格式

bit7	bit 6	bit5	bit4	bit3	bit2	bit1	bit0
S	2^6	2^5	2^4	2^3	2^2	2^1	2^0

*T_H and T_L Register Format

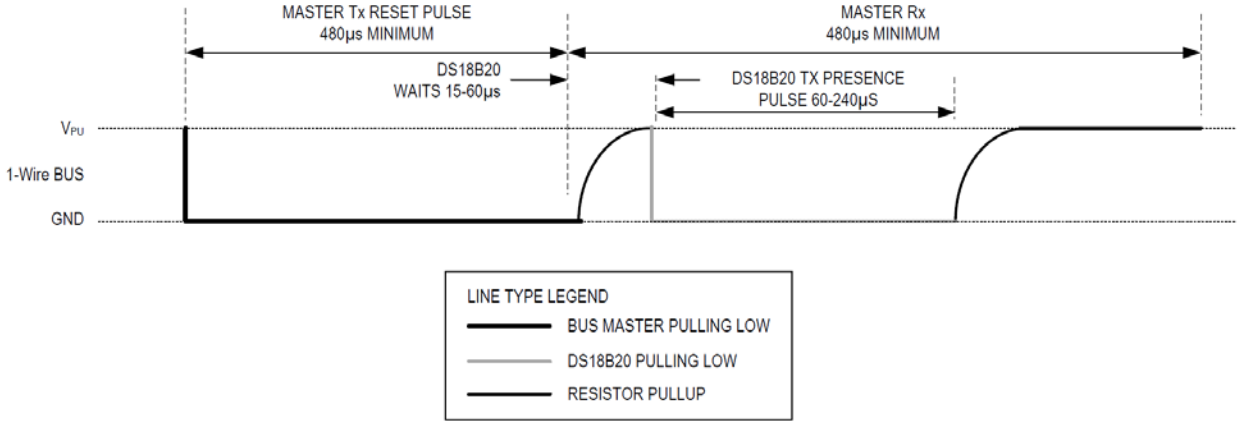
（报警值 T_H 和 T_L在寄存器中格式）

时序

获取温度流程

复位 → 检测存在脉冲 → 发送命令(0XCC) → 发送命令(0x44) → 复位 → 检测存在脉冲 → 发送命令(0XCC) → 发送命令(0xBE) → 获取字节(低8位温度数据) → 获取字节(高8位温度数据)

初始化时序

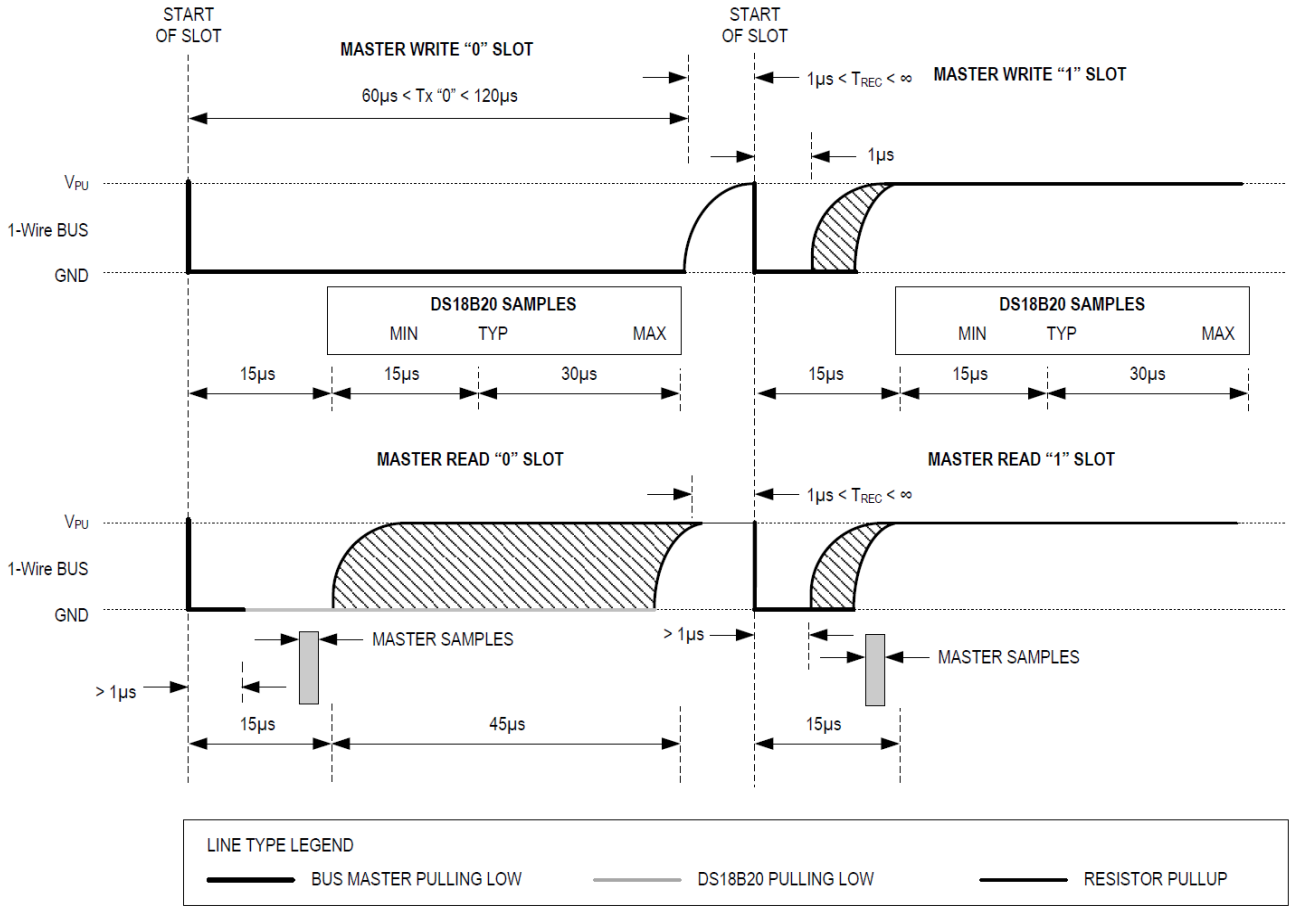


首先控制器(单片机)设置为推挽输出模式，向从机(DS18B20)写入至少持续 480µs 低电平复位脉冲;

从高电平设置到低电平，持续最少 480µs，等待 15-60µs，再将电平拉高;

接收 60-240µs 的低电平存在脉冲，从电平（被）拉高开始，到检测存在脉冲完毕后至少持续 480µs。

读写数据时序



写0:

输出低电平启动写时序，该时序时间：1µs< Time <15µs;

继续输出低电平（从机在这个窗口期进行采样），15µs<从机采样时间<45µs，整个写0过程必须在控制在60µs< Time <120µs;

写0完毕后输出高电平以表示本次写数据结束，且1µs<Time<∞;

写1:

输出低电平启动写时序，1µs< Time <15µs;

输出高电平（从机在这个窗口期进行采样），15µs<从机采样时间<45µs，上图整个无表示写1过程时间控制要求，根据手册原文（"The DS18B20 samples the 1-Wire bus during a window that lasts from 15µs to 60µs after the master initiates the write time slot. If the bus is high during the sampling window, a 1 is written to the DS18B20. If the line is low, a 0 is written to the DS18B20. "）可知全程时间安排与写0相同为最佳选择。

结束时序:

写0完毕后输出高电平以表示本次写时序结束，且1µs<Time<∞；写1完毕后则直接延迟同样时间

读0、读1:

输出低电平启动读时序，该时序时间：1μs< Time <15μs，此时从机会发送数据位；

设置为上拉输入模式，在启动读时序中15μs内开始读值，等待45μs（期间模块会自动上拉电阻变成高电平，读1同样没有时间要求，但最好和读0时间保持一致）；

模块自动上拉电阻，恢复高电平；

读数据完毕后输出高电平以表示本次读数据结束，且维持1μs<Time<∞；

后记

遇到问题：

1. 读出两个数据都是0xFF，查知检测不到合格的存在脉冲（时间太短只有0μs，按照时序图是说大于60μs的，真实原因还是个谜。。。），删除检测时间不低于60μs即可。
2. 第一次读值会出错，属于正常情况

```
DS18B20 is initializing......Wait a moment Plz ^v^
DS18B20 is Initialized
writing byte:0xcc       0       0       1       1       0       0       1       1
writing byte:0x44       0       0       1       0       0       0       1       0
Skip ROM:
writing byte:0xcc       0       0       1       1       0       0       1       1
Read Scratchpad:
writing byte:0xbe       0       1       1       1       1       1       0       1
low temp: 0xf1
high temp: 0x1
zhenshu
******
447.81250 Degree Celsius
******

writing byte:0xcc       0       0       1       1       0       0       1       1
writing byte:0x44       0       0       1       0       0       0       1       0
Skip ROM:
writing byte:0xcc       0       0       1       1       0       0       1       1
Read Scratchpad:
writing byte:0xbe       0       1       1       1       1       1       0       1
low temp: 0xf1
high temp: 0x1
zhenshu
******
31.06250 Degree Celsius
******
```

第一次读值

第二次读值