

如果到現在還在利用AD590做溫度感測，那可就落伍了，現在有結合半導體溫度感測器，運算放大器電路和AD轉換電路成一顆晶片，只有三隻接腳，外觀就如同一顆電晶體，可將溫度直接轉換成12位元的數位值，量測範圍從+125度到-55度，最高解析度可達0.0652度，相關基本資料如下圖所示。

如果要參考詳細資料請自行下載[資料手冊](#)。(另有一顆功能相近，解析度較低的DS1820簡體中文版[使用手冊](#)，亦可自行下載參考)



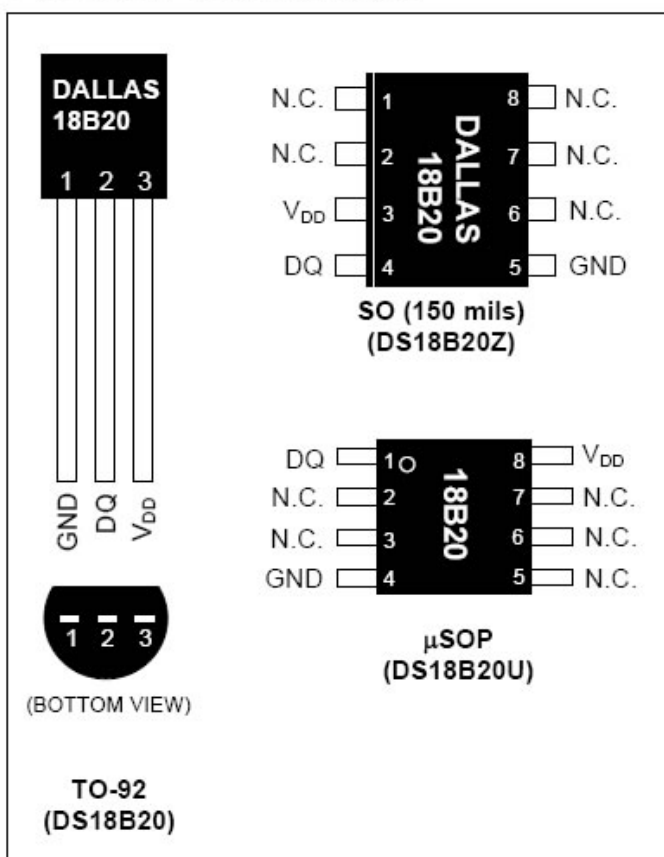
DS18B20 Programmable Resolution 1-Wire Digital Thermometer

www.maxim-ic.com

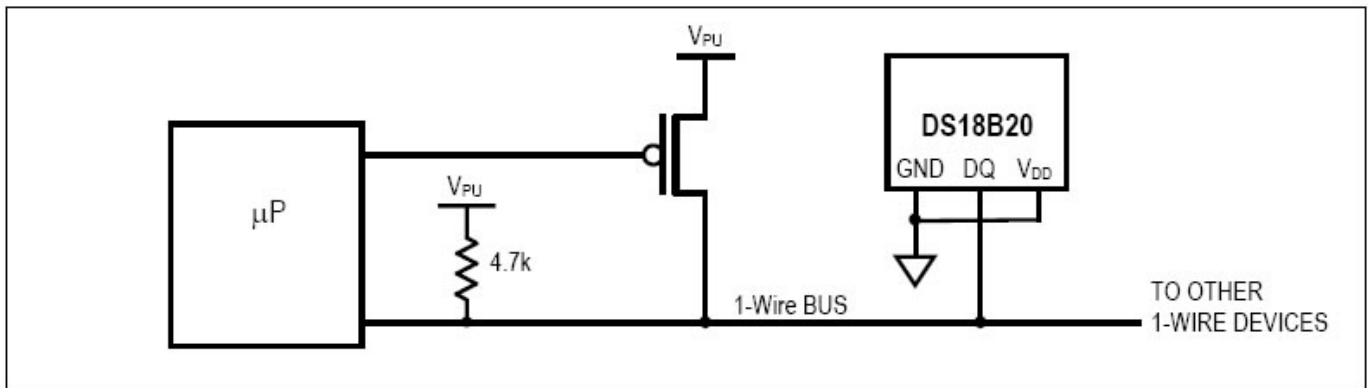
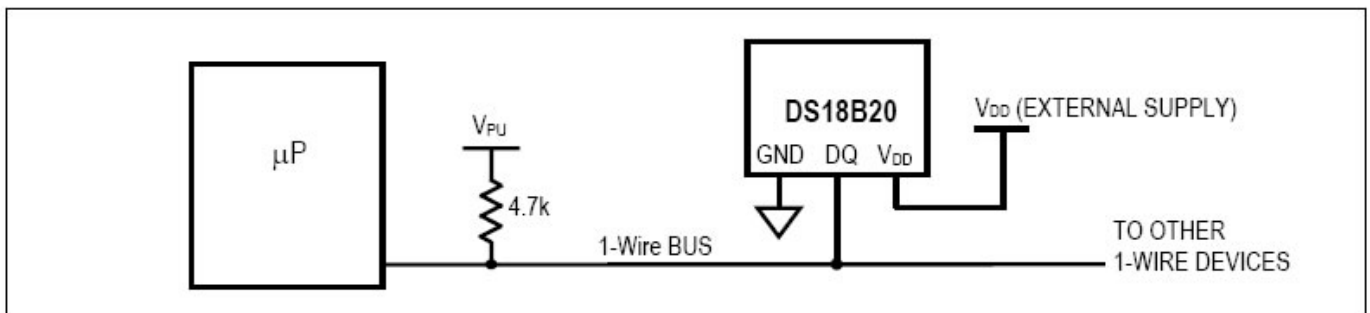
FEATURES

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line; Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C Accuracy from -10°C to +85°C
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)
- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Available in 8-Pin SO (150 mils), 8-Pin μ SOP, and 3-Pin TO-92 Packages
- Software Compatible with the DS1822
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System

PIN CONFIGURATIONS

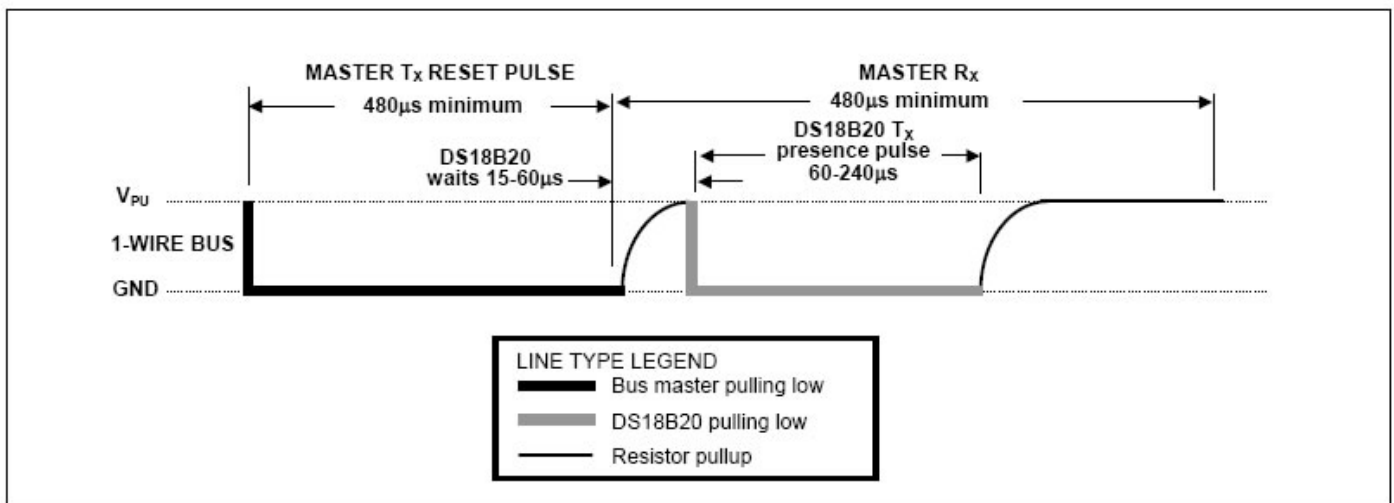


由於此顆晶片只有三隻腳，扣掉電源線，只剩一隻接腳可用來做資料傳輸線，透過這隻腳，可將控制命令寫入DS18B20，也透過它把溫度讀出來，這是如何辦到的？

Figure 4. Supplying the Parasite-Powered DS18B20 During Temperature Conversions**Figure 5. Powering the DS18B20 with an External Supply**

首先由其接線方式說起，此晶片與單晶片連接的方式有以上2種，圖4的接法連電源線都省了，但控制時需考慮的條件較多，因此通常會使用圖5的接法，利用一個4.7k歐姆的電阻將資料線提升至高電位。

當CPU要與DS18B20溝通時，先輸出0，將此接線電壓拉至0V，持續至少480 us，再將輸出訊號轉為1。DS18B20感應到此起始訊號後，也會產生持續60~240 us的低電壓做回應，CPU只要從有無此回應訊號，就知道DS18B20有沒有接上。

Figure 13. Initialization Timing

依照圖13之初始規劃之時序，可先寫一簡單程式做練習，偵測DS18B20是否存在

```

DQ      EQU      P3.3      ;DS18B20資料線

CLR     DQ
MOV     R5,#50             ;持續 500 us, 起始訊號
CALL    DELAY10
SETB    DQ
MOV     R5,#8              ;持續 80 us, 等待DS18B20回應
CALL    DELAY10

JNB     DQ,OK_18B20

```

```

MOV    P2,#55H      ;DS18B20不存在時,P2顯示01010101
JMP    $
OK_18B20:

MOV    P2,#0FH      ;DS18B20存在時,P2顯示00001111
JMP    $

DELAY10:      ;延遲時間 = R5 * 10 us, 此時間算法是以笙泉之82G516為準
MOV     R7,#30
DJNZ   R7,$
DJNZ   R5,DELAY10
RET

```

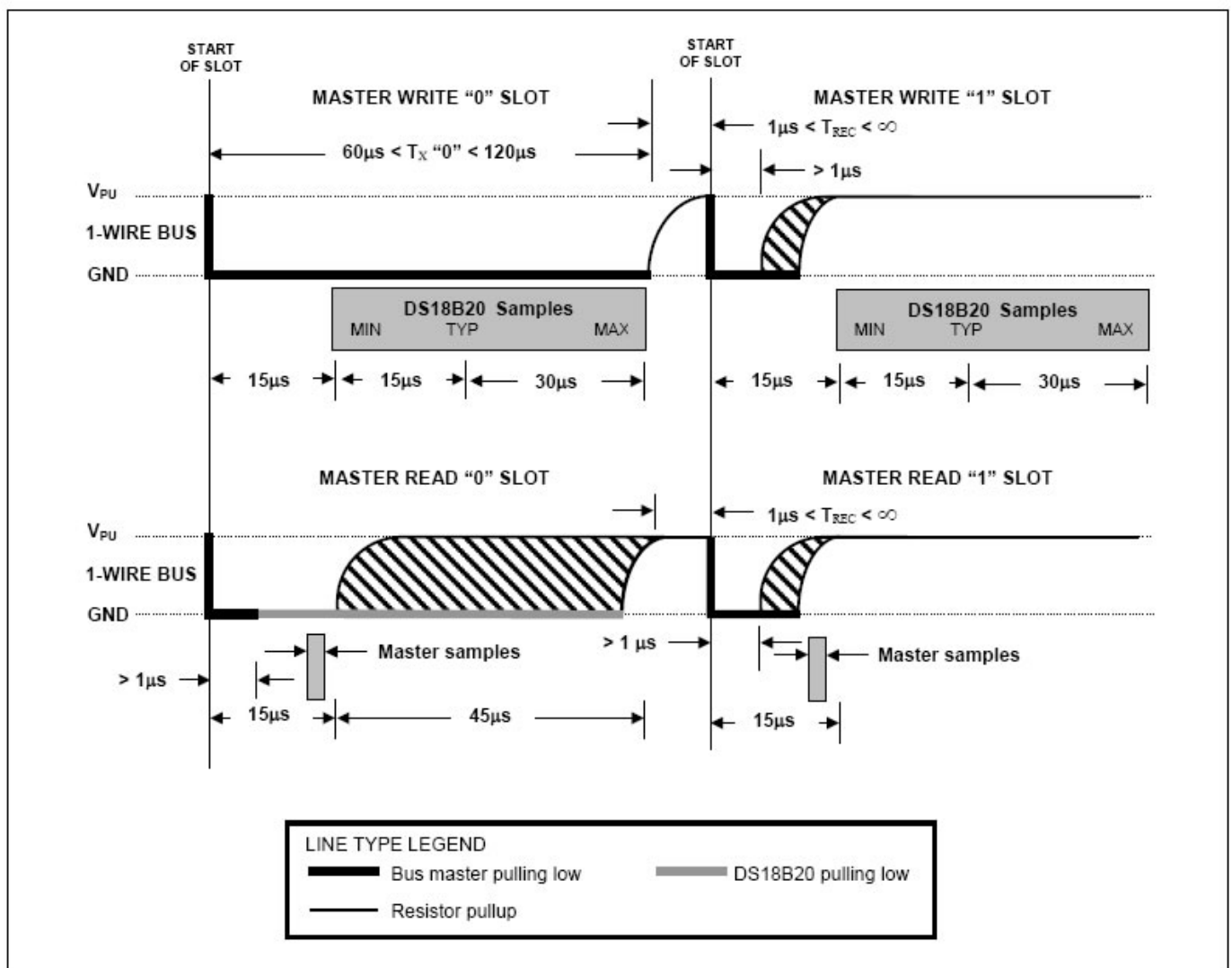
DS18B20規定要先初始規劃才可以對它下後續之命令，如果確定DS18B20是接好了，可直接寫成以下副程式即可。

```

INIT_18B20:      ;喚醒DS18B20
CLR     DQ
MOV     R5,#50      ;持續 500 us, 起始訊號
CALL    DELAY10
SETB    DQ
MOV     R5,#50      ;持續 500 us, 等待DS18B20回應結束
CALL    DELAY10
RET

```

Figure 14. Read/Write Time Slot Timing Diagram



對DS18B20初始規劃完畢後，便可開始對它下命令，問題是如何將0或1寫入此顆晶片？圖11上方波形為0與1之寫入時序，如果要寫入0，只要CPU輸出0，維持約60us就可以了，如果要寫入1，則CPU要先輸出0，在15us內，將此接腳提升至1，整個動作也是要維持超過60us。

以下副程式已經調整成可寫入一個位元組，將要寫到DS18B20的資料先放在累加器A，透過右旋將低位元傳至C，再將C的資料傳給DS18B20的DQ，如果C=0，則DQ就維持著0，如果C=1，則DQ維持著1，再透過迴圈，如此便可連續將8位元資料寫入DS18B20。

```

WR_BYTE:                ;寫入一位元組訊號
    MOV    R6,#8
WR_1BIT:
    RRC     A
    CLR     DQ
    MOV     R7,#45        ;持續至多15 us,起始訊號，此時間算法是以筆泉之82G516為準
    DJNZ    R7,$
    MOV     DQ,C          ;將要寫入DS18B20的資料送出
    MOV     R7,#60        ;持續20 us,等待ds18b20讀取
    DJNZ    R7,$
    SETB    DQ
    MOV     R7,#100       ;持續35 us,等待ds18b20回復
    DJNZ    R7,$
    DJNZ    R6,WR_1BIT

    RET

```

圖11下方波形為0與1之讀取時序，CPU要讀取DS18B20時，須先將此腳先拉至0，然後馬上回復至1，再來偵測這支接腳，如果DS18B20輸出為1，它會在15微秒內將此腳提升至1，如果DS18B20輸出為0，它會將此接腳維持著0，因此CPU在約15微秒時，便可讀取DS18B20的輸出值。整個動作也是要維持約60us。

```

RD_BYTE:                ;讀取一位元組訊號
    MOV     R6,#8
RD_1BIT:
    CLR     DQ
    MOV     R7,#2         ;起始訊號，持續 2 us
    DJNZ    R7,$
    SETB    DQ
    MOV     R7,#30        ;等待 10 us後再讀訊號訊號
    DJNZ    R7,$
    MOV     C,DQ          ;從DQ接腳讀回DS18B20送出之訊號
    RRC     A
    MOV     R7,#150       ;持續50 us,等待ds18b20回復
    DJNZ    R7,$
    DJNZ    R6,RD_1BIT

    RET

```

有辦法對DS18B20讀取或寫入1位元組的訊號後，該對它下什麼命令呢？

DS18B20範例(使用時請將JP1上方2隻接腳短路)

在七段顯示器上顯示：[範例\(ASM\)](#)

在16x2 LCD上顯示：[範例\(ASM\)](#)