

常用指令排行榜(二)

上一篇文章我們提到 MCS-51 指令集，也就是 8051 指令集的準備與用法，接下來我們要從一些簡單的範例開始。

一般來說，最適合初學者的範例，莫過於霹靂燈的控制，因為同樣一種燈號可以用好幾種不同的寫法表現，最重要的是，他不需要花費很高的費用即可看到自己辛苦的成果，一個炫目的霹靂燈程式只要稍加裝飾，就可以是很好的聖誕燈或是霓虹燈的控制系統，這在日常生活中的應用是相當多的。基於這種種的優點，所以本文將從燈號控制作為起點，為您介紹常用的 8051 指令。

我們先看一小段程式範例，這是讓一個 P1.0 接腳上的 LED 產生明滅效果最簡單的寫法：

```
START  MOV      R1,#00H      ;系統啟始時先延遲一小段時間使其穩定
$      DJNZ     R1,$
      MOV      SP,#60H      ;堆疊從 60H 開始(常用設定範圍是 3FH~7FH)
;
LOOP   CLR      P1.0        ;也可以寫成 CLR  90H.0
      CALL     DELAY        ;延遲上一個狀態一小段時間
      SETB    P1.0        ;也可以寫成 SETB  90H.0
      CALL     DELAY        ;延遲上一個狀態一小段時間
      SJMP    LOOP        ;回到 LOOP 再循環
;
DELAY  MOV      R0,#00H      ;延遲時間的副程式
$1     MOV      R1,#00H
$2     DJNZ     R1,$2
      DJNZ     R0,$1
      RET
```

接下來要先介紹的指令是 SETB 與 CLR，很老生常談，不是嗎？但是這裡面卻透露了另一個訊息：他真的很重要。

SETB 與 CLR 都是控制『單一 bit』的指令，SETB 是將控制的該 bit 填成 1，而 CLR 則是填成 0。許多人在撰寫程式時會不小心用了 SETB P3，或是 CLR P1 這樣的錯誤寫法，導致晶片無法動作，原因錯在 P3 與 P1 埠都是一個 Byte (8 bits) 的儲存空間，而這兩個指令只能控制一個 bit。同樣的，SETB 2BH 或是 CLR C0H 都是錯誤的寫法，**正確的寫法是 SETB P3.0 或是 CLR C0H.4** 這樣的寫法才正確。在 MCS-51 裡提供給 SETB 與 CLR 的寫法有兩種，一種是 SETB bit (2B 1C) 及 CLR bit (2B 1C)，另一種是 SETB C (1B 1C) 及 CLR C (1B 1C)。唯一的例外是『CLR A』，這個指令可以把 Acc 累加器的 8 個 bits 全部都清為 0，比起『MOV A,#00H』還要少一個 Byte 的空間，當你對程式空間有斤斤計較的需要時，你會發現這樣的指令是很實用的！以上這些指令可以在查表中看到，而括號中所寫的 B 與 C 分別代表所佔用的 Byte 數與所需要的機械週期 (Cycle)，一個機械週期在 MCS-51 的規範裡指的是振盪器振盪 12 次，這有什麼意義嗎？

當然是有的啦！不但有意義，而且還很重要呢！假設我們在電路設計上使用了 12M 的石英振盪器（也就是一秒鐘振盪 12 百萬次），每振盪 12 次代表一個機械週期，那麼一個機械週期所要花費的時間就是 1 uS，看到這裡有沒有一種恍然大悟、茅塞頓開的感覺？是了，這是我們用來計算系統運作時間的依據！在較高階的 8051 CPU 裡可能有執行一個週期只需花費四次振盪的高速 CPU，在原始程式完全不改的情況下可以讓執行速度就立刻快上三倍，當然其價格自然也是三級跳的囉！

到此為止，您是否對 8051 有更進一步的瞭解了呢？我們的程式還沒介紹完唷，下一篇，我們要來探討一下 JMP（跳躍指令）跟位址的關係，請密切鎖定 8051 初學者講座。