

# 第 8 章

## MCS-51 的內部計時/計數器

6-1	Timer 0 和 Timer 1 的工作模式	6-2
6-2	計時/計數器的控制邏輯分析	6-3
6-3	如何啟動計時/計數器	6-5
6-4	Timer 0 和 Timer 1 的 Mode 0 工作模式	6-6
6-5	Timer 0 和 Timer 1 的 Mode 1 工作模式	6-9
6-6	Timer 0 和 Timer 1 的 Mode 2 工作模式	6-10
6-7	Timer 0 和 Timer 1 的 Mode 3 工作模式	6-11
6-8	讓 MCS-51 的 Timer 正確工作的程式設計	6-12

MCS-51 內部有兩個 16 位元計時/計數器稱為 Timer 0 和 Timer 1，Timer 0 和 Timer 1 實際上是由兩個 8bit 的暫存器所組成。構成 Timer 0 的兩個暫存器稱 TH 0 和 TL 0；Timer 1 的兩個暫存器名稱為 TH 1 及 TL 1。因為這兩組暫存器的特性都一樣，因此底下我們以 THX 代表 TH 0 及 TH 1，以 TLX 代表 TL 0 及 TL 1。



## 6-1 Timer 0 和 Timer 1 的工作模式

MCS-51 內部的兩個 16 位元計時/計數器 Timer 0、Timer 1 各有 4 種工作模式，除模式 3 在 Timer 0 與 Timer 1 中的用法不同外，模式 0、模式 1 及模式 2 在 Timer 0 與 Timer 1 的用法皆相同。

Mode 0：13 位元計時/計數模式

Mode 1：16 位元計時/計數模式

Mode 2：附有自動載入功能的 8 位元計時/計數器

Mode 3：Timer 0 被拆成兩個獨立的 8 位元計時/計數器 TL 0 和 TH 0。

Timer 1 在 Mode 3 時停止計時/計數的工作。

### • 計時模式

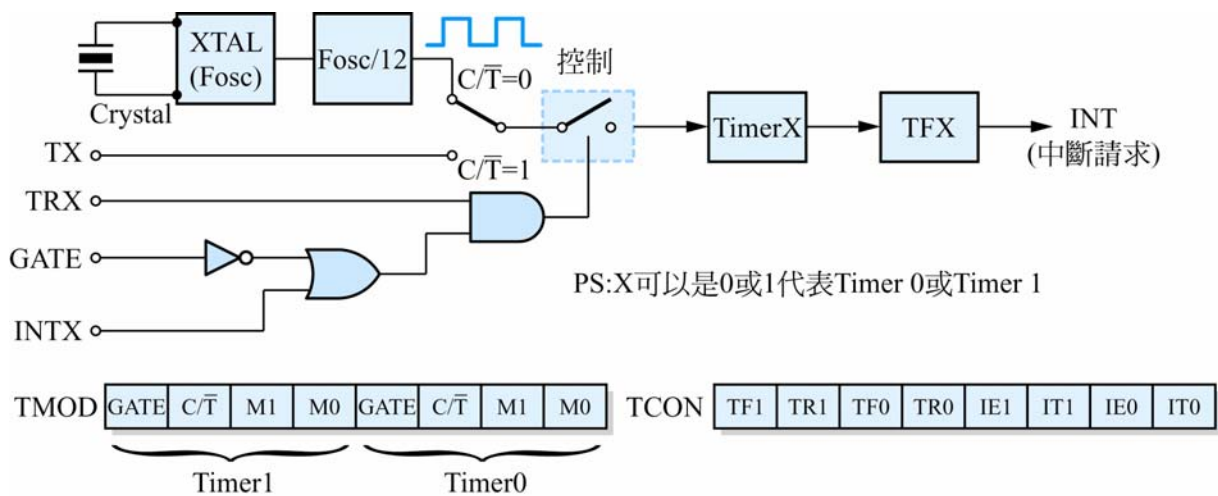
在計時模式下，計時暫存器（Timer 0 或 Timer 1）的內容值，隨著每個振盪脈波( $@F_{osc}/12$ )上數加 1，如果使用 $@F_{osc}=12\text{MHz}$ ，則每個計時脈波為 1 微秒 ( $1\mu\text{s}$ )，在計時模式下，可使用 TMOD 暫存器的 M 1 及 M 0 位元設定計時模式，共有 4 種模式可設定，即模式 0、模式 1、模式 2、模式 3。

### • 計數模式

在計數模式下，當其對應的輸入腳(T0、T1)有負緣信號( $\downarrow$ )時，計數器暫存器（Timer 0 或 Timer 1）內容加 1，由於需要 2 個機械週期的時間檢知負緣信號變化，所以可輸入的最大計數頻率為 $@F_{osc}/24$ ，如果我們使用 $@F_{osc}=12\text{MHz}$ ，則可輸入的最大計數頻率為 500kHz。在計數模式下，可使用 TMOD 暫存器的 M 1 及 M 0 位元設定計數模式，共有 4 種模式可設定，即模式 0、模式 1、模式 2、模式 3。

## 6-2 計時/計數器的控制邏輯分析

MCS-51 內部的計時/計數器 Timer 0 和 Timer 1 都有獨立的四種計時/計數模式。其控制與執行模式均操作在計時/計數控制暫存器(TCON)與計時計數模式暫存器(TMOD)。



● 圖 6-1 計時/計數器控制邏輯分析圖

### 1. 計時/計數器控制暫存器 (TIMER/COUNTER CONTROL REGISTER)



#### TF1：Timer 1 之溢位旗號

當 Timer 1 溢位時，TF1 會被硬體設定為 1。當處理器執行中斷服務程式時，硬體會自動清除此位元。

#### TR1：Timer 1 之啟動位元

由軟體設定/清除以啟動/停止計時/計數。  
設為"1"時啟動，"0"時停止。

#### TF0：Timer 0 之溢位旗號

當 Timer 0 溢位時，會被硬體設為 1。  
當處理器執行中斷服務程式時，硬體會自動清除此位元。

**TR0**：Timer 0 之啟動位元

軟體設為 1 時啟動，0 時停止。

**IE1**：外部中斷 INT1 之中斷旗號

當外部中斷被檢知時，硬體會設定此位元為"1"。

當中斷被處理時，硬體會自動清除此位元。

**IT1**：外部中斷 INT1 之觸發型態控制

當設定為 1 時，中斷觸發型態為負緣觸發。

當此位元為 0 時，則為低準位觸發。

**IE0**：外部中斷 INT0 之邊緣旗號

當檢知外部中斷時，此位元會被硬體設為 1。

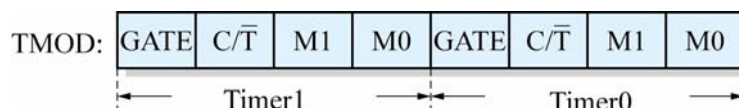
執行中斷服務程式時，硬體會將之清除。

**IT0**：外部中斷 INT0 之觸發型態控制

當設定為 1 時中斷觸發型態為負緣觸發。

當此位元為 0 時，則為低準位觸發。

## 2. 計時/計數模式控制暫存器 (TIMER/COUNTER MODE CONTROL REGISTER)

**GATE**：閘控制位元

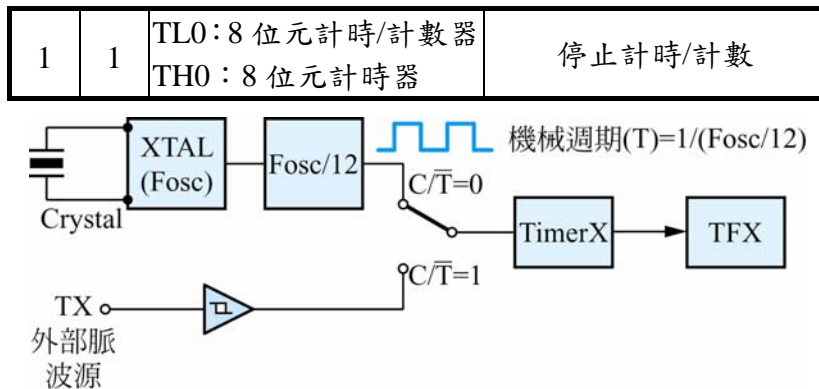
1. GATE=0 時，只要 TRX=1，計時/計數器就能動作。
2. GATE=1 時，必須 TRX=1 且 INTX=1 時，計時/計數器才能動作。

**C/ $\bar{T}$** ：計時器或計數器之選擇位元。

C/ $\bar{T}$ =0 時為計時模式，C/ $\bar{T}$ =1 時為計數模式。

**M1、M0**：計時/計數模式控制選擇位元

M1	M0	計時/計數器 0(Timer 0)	計時/計數器 1(Timer 1)
0	0	13 位元計時/計數器	13 位元計時/計數器
0	1	16 位元計時/計數器	16 位元計時/計數器
1	0	8 位元計時/計數器 (可自動載入)	8 位元計時/計數器 (可自動載入)



● 圖 6-2 Timer 0 和 Timer 1 的脈波來源

如圖 6-2 所示，送到計時/計數器 Timer X（也就是 TLX 和 THX）的脈波源有兩個，其中一個是 MCS-51 外部晶體振盪器頻率除以 12 後的脈波源，另一個是由 TX（T0 是 Timer 0 的外部輸入脈波腳，T1 是 Timer 1 的外部輸入脈波腳）接腳輸入的脈波源。這兩個脈波源究竟何者會被送入計時/計數器，這是由 TMOD 暫存器中  $C/\bar{T}$  位元來決定。當  $C/\bar{T}=1$  時，就選擇外部 TX 接腳所輸入的脈波源。當  $C/\bar{T}=0$  時就選擇內部脈波源。

這裏所指的內部脈波源 XTAL，是指加在 OSC1 及 OSC2 兩支腳上的晶體振盪器 (Crystal)。

■ 表 6-1 常用晶體振盪頻率與機械週期關係一覽表

一般常用的晶體振盪頻率	Fosc÷12 後的頻率	Fosc÷12 後的週期
12MHz	1MHz	1μs
11.0592MHz	921.6kHz	1.08507μs
6MHz	0.5MHz	2μs

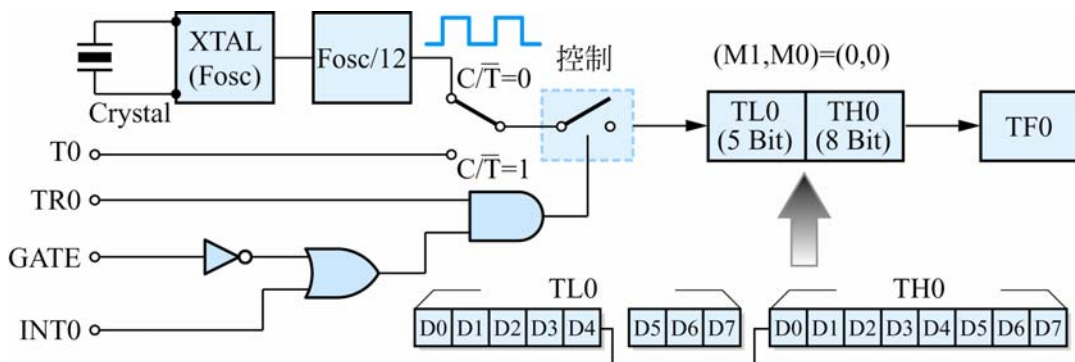
### 6-3 如何啟動計時/計數器

由圖 6-1 所示的 Timer 0、Timer 1 控制邏輯分析圖可知，若要計時/計數器開始計時或計數，則您必須以程式去設定 TMOD 及 TCON 中的相關位元。啟動計時/計數器的方法如下：

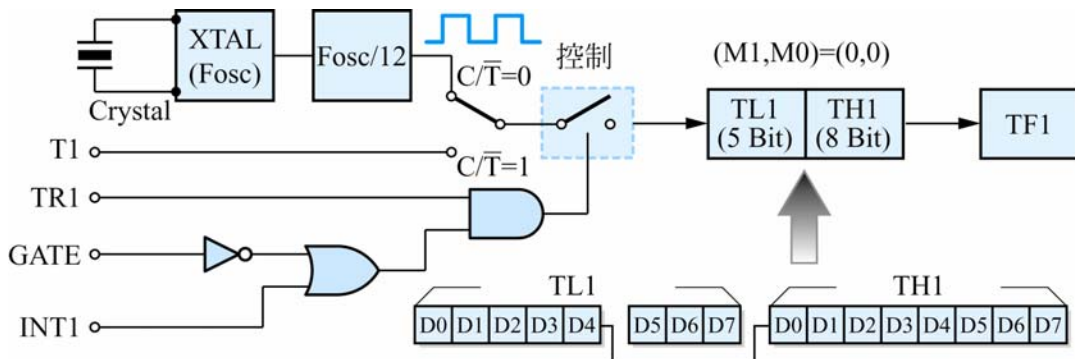
- 方法一 若 GATE=0 及 TRX=1 時
- 當  $C/\bar{T}=0$  啟動內部計時器
- $C/\bar{T}=1$  啟動外部計數器。

- 方法二** 若  $GATE=1$  及  $TRX=1$  且  $INTX=1$  時  
 當  $C/\bar{T}=0$  啟動內部計時器  
 $C/\bar{T}=1$  啟動外部計數器。  
 (提示：建議採用方法 1)

### 6-4 Timer 0 和 Timer 1 的 Mode 0 工作模式



● 圖 6-3 Timer 0 Mode 0 的控制邏輯圖



● 圖 6-4 Timer 1 Mode 0 的控制邏輯圖

#### 6-4-1 計時時間/計數次數的計算及設定

Timer X 的計時/計數方式屬於上數型，而 Mode 0 工作模式是由 TLX(5bit) 及 THX(8bit) 所組成，總共是 13 位元，因此其最多計數值為  $2^{13}=8192$  個脈波。假如我們所需要的只是要計數 5000 個脈波就好，這時我們可以設 Timer X 的初值為  $8192-5000=3192$ ，然後啟動 Timer X，當 Timer X 數到 8192 產生溢位而去設

定 TFX 旗標時，正好數了 5000 個脈波。

要將 3192 寫到 Timer X 中的 TLX 和 THX 中可由以下計算公式獲得：

$$TLX=3192 \text{ MOD } 32=24 \quad (\text{MOD 為取除法的餘數})$$

$$TH0=3192/32=99 \quad (\text{取商數})$$

$$\text{提示 } 3192 \div 32 = 99 \dots 24 \quad (\text{商數寫入 THX 中，餘數寫入 TLX 中})$$

實際應用的程式寫法：

```
MOV TL0,#(2**13-5000).MOD.32
```

```
MOV TH0,#(2**13-5000)/32
```

提示  $2^{13}$  即  $2^{13}=8192$

如果是要計時/計數 n 個脈波的程式寫法：

```
MOV TL0,#(2**13-n).MOD.32
```

```
MOV TH0,#(2**13-n)/32
```

#### 6-4-2 CPU 如何知道已計時/計數完成

Timer X 是一個上數計數器，不管加給這上數器的脈波源是來自晶體振盪器所產生的脈波，或是由 T0 或 T1 所外加的脈波，當您把初值設給 Timer X 後，Timer X 根據初值，每送入一個脈波，就往上加 1，一直到溢位，而 Timer X 產生溢位的這動作，就會將旗號 TFX (Timer 0 為 TF0、Timer 1 為 TF1) 設定為 1。因此 CPU 只要檢查 TFX 的狀態就可以知道計時/計數是否已終了。

Mode 0 的工作模式，最高計數值為  $2^{13}=8192$ ，假設我們寫入 TLX、THX 的初值為 3192，每送入一脈波就往上加 1，最後加到 13 位元所能容納的最大值 1111111111111(8191)，則再送入一個脈波，就會產生溢位去設定 TFX 為 1。

#### 6-4-3 寫一個完整計時/計數程式

假設我們要利用 Timer 0，工作模式為 Mode 0，寫一個計時 0.5 秒的延遲副程式，參考程式如下：

```

Delay :   MOV  R7,#100                ; 設定 Timer 0 計數 100 次
          MOV  TMOD,#00000000B      ; 設定 Timer 0 的工作模式為 Mode 0
Ready :   MOV  TL0,#(8192-5000).MOD.32 ;
          MOV  TH0,#(8192-5000)/32   ; 設定 Timer 0 計數 5000 個脈波
          SETB TR0                    ; 啟動 Timer 0 開始計時
Wait :    JNB  TF0,Wait              ; 等待 Timer 0 計時是否到了(判斷 TF0
                                     ; 是否等於 1)
          CLR  TF0                    ; 計時到就將 TF0 清除為 0
          DJNZ R7,Ready               ; 執行計數 100 次
          RET

```

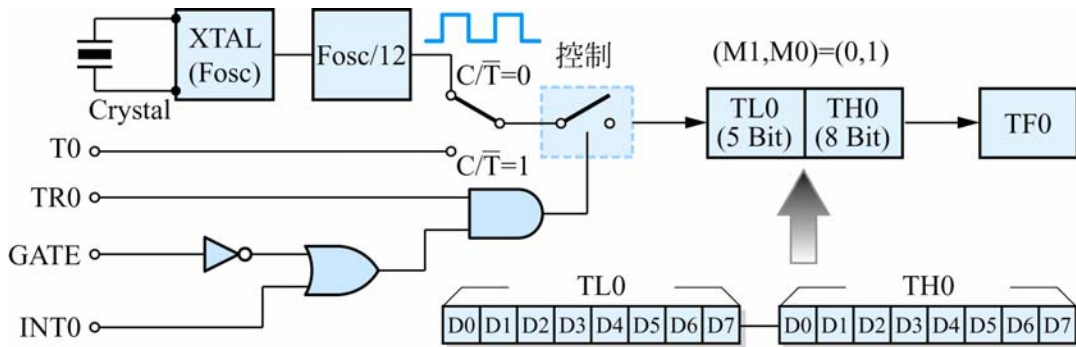


提示

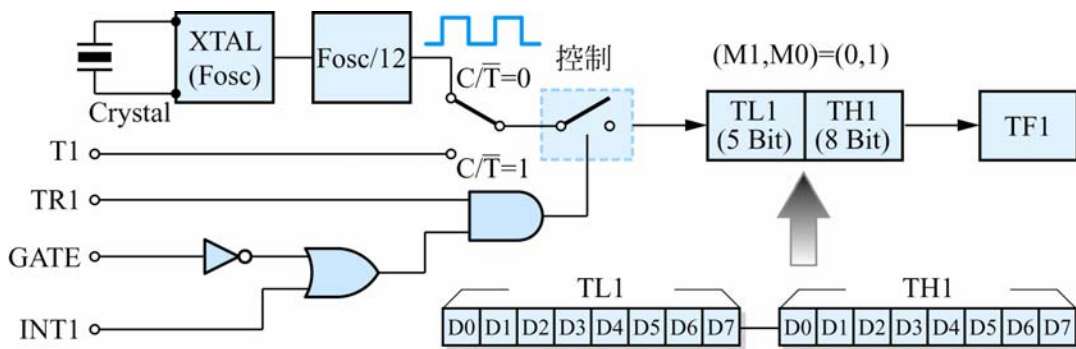
1. 程式中的晶體振盪頻率是以 12MHz 來計算的，因此 Timer 0 所接受到的脈波是  $1\mu s$  的脈波，指令 SETB TR0 啟動了 Timer 0 開始計時後，Timer 0 就開始接受由晶體振盪器所傳來的  $1\mu s$  脈波。每送一個脈波，Timer 0 就上數 1，數超過 5000 時，Timer 0 就會產生溢位，而去設定旗號 TF0=1。
2. 指令 JNB TF0,Wait 就是用來偵測 Timer 0 計時溢位否，若溢位了，也就是 TF0=1 時，指令 JNB TF0,Wait，才會繼續往下執行，否則還是一直在執行自己的迴圈。因此從開始計時 (SETB TR0)，到計時完畢 (TF0=1) 離開 JNB TF0,Wait，總共大約花了  $5000 \times 1\mu s = 5000\mu s = 5ms$  的時間。而 R7=100 共讓此"計時 5ms 的動作" 工作了 100 次，因此整個 Delay 副程式執行完共花費大約  $R7 \times 5ms$  的時間，若 R7=100，則花費  $100 \times 5ms$  秒=0.5 秒的時間。
3. 當 TF0 被設定為 1 後，CPU 並不會自動將它清除為 0，因此我們在程式中加了 CLR TF0，就是用來清除 TF0=0，以便使指令 JNB TF0,Wait 不會有誤動作出現。



## 6-5 Timer 0 和 Timer 1 的 Mode 1 工作模式



● 圖 6-5 Timer 0 Mode 1 的控制邏輯



● 圖 6-6 Timer 1 Mode 1 的控制邏輯

Mode 1 是由 TLX(8bit) 和 THX(8bit) 組成一個 16 位元計時/計數器。Mode 1 的動作及使用方法与 Mode 0 完全一樣。只不過 Mode 1 是 16 位元計時/計數器而 Mode 0 是 13 位元，因此其最大計時/計數脈波值為  $2^{16} = 65536$ 。

要讓 Mode 1 計時/計數 n 個脈波的程式寫法：

$$TLX = (2^{16-n}) \text{MOD} 256$$

$$THX = (2^{16-n}) / 256$$

提示： $2^{16} = 65536$

實際應用上的程式寫法 (Timer 0 中的舉例)

```
MOV TL0,#(2**16-n).MOD.256
```

```
MOV TH0,#(2**16-n)/256
```

或

```
MOV TL0,#<(2**16-n)
```

```
MOV TH0,#>(2**16-n)
```

提示：“<” 取 16 位元組的低 8 位元組(Low Byte)  
“>” 取 16 位元組的高 8 位元組(High Byte)

## 6-6 Timer 0 和 Timer 1 的 Mode 2 工作模式

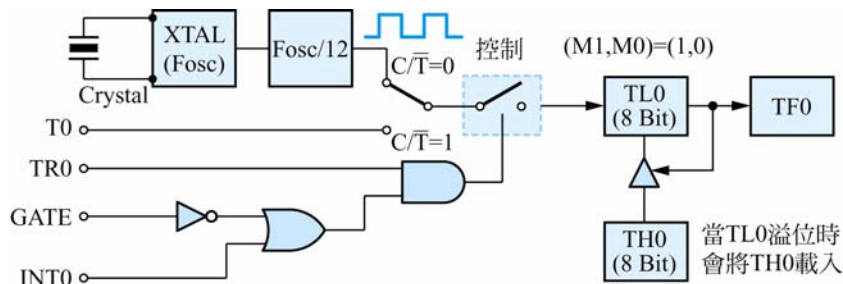
Timer X 工作在 Mode 2 時，變成一個具有自動重新載入 (Auto-ReLoad) 功能的 8 位元計時/計數器。它是將原來 16 位元的計時/計數器，拆成兩個 8 位元暫存器 (THX 和 TLX)，其中 TLX 為真正計時/計數脈波暫存器，THX 則變成放置重新載入值的暫存器。

所謂的自動重新載入功能，就是 TLX 計時/計數產生溢位時，除了設定 TFX 旗號等於 1 以外，還會將 THX 的內容載入 TLX 中，TLX 就可以根據這個載入值繼續往上數。

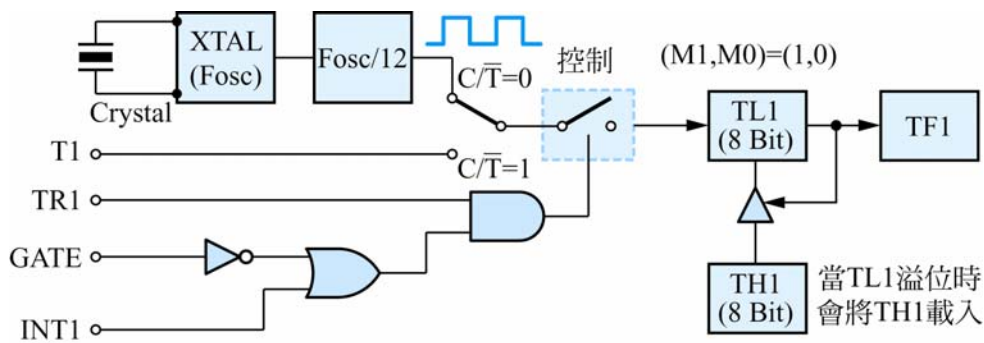
因為自動載入模式是將 THX 的內容載入至 TLX，因此要改變計時/計數時間，就必須改變 THX 的內容。Mode 2 是一個 8 位元計時/計數器，因此最多可以計時/計數  $2^8=256$  脈波。

在 Mode 2 中計時/計數 n 個脈波程式寫法：(在 Timer 0 中)

```
MOV TH0,#256-n
```



● 圖 6-7 Timer 0 Mode 2 的控制邏輯圖

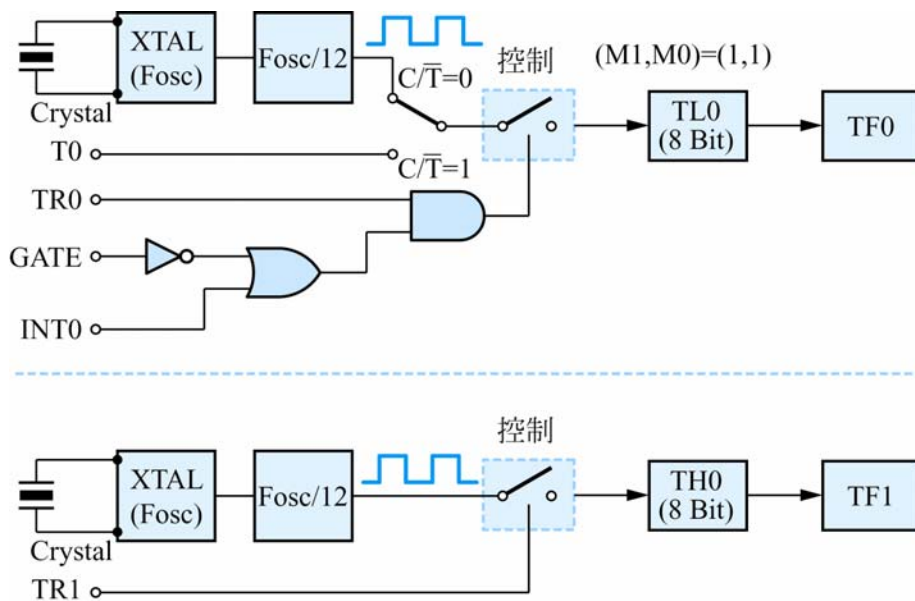


● 圖 6-8 Timer 1 Mode 2 的控制邏輯圖



### 6-7 Timer 0 和 Timer 1 的 Mode 3 工作模式

在 Mode 3 的工作模式中，Timer 0 和 Timer 1 是不同的。若 Timer 1 工作在 Mode 3 時，Timer 1 就停止工作。而 Timer 0 工作在 Mode 3 模式時，原來 16 位元的計時/計數器被拆成兩個獨立的 8 位元計時/計數器 (TL0, TH0)。其中 TL0 為 Timer 0 的 8 位元計時/計數器，其用法與 Mode 2 完全一樣，只是沒有自動載入的功能。另外一個 8 位元 TH0，它固定用來計時內部脈波，並且它所使用的控制邏輯旗號是 TR1 和 TF1，也就是如要啟用 TH0，您必須以軟體設定 TR1=1，當 TH0 產生溢位後會將 TF1 設為 1，如圖 6-9 所示。



● 圖 6-9 Timer 0 Mode 3 的控制邏輯圖



## 6-8 讓 MCS-51 的 Timer 正確工作的程式設計

要使 Timer 正確工作，則程式上有下列三點必須配合：

1. 設定工作模式 (規劃 TMOD 暫存器)
2. 設定計時計數時間 (規劃 THX 和 TLX)
3. 啟動計時器 (設定 TRX=1)

若計時/計數工作中途要停止，可清除 TRX=0，即可停止目前正在進行的計時/計數工作。

### 採用 Timer 0 Mode 0


```
MOV    TMOD,#00000000B    ;設定 Timer 0 的工作模式為
                                ; Mode 0 計時器
MOV    TL0,#(2**13 - 5000).MOD.32    ;設定 Timer 0 計時 5000 個
MOV    TH0,#(2**13 - 5000)/32    ;脈波
SETB   TR0                ;啟動 Timer 0 開始計時
```

### 採用 Timer 0 Mode 1

```
MOV    TMOD,#00000001B    ;設定 Timer 0 的工作模式為
                                ; Mode 1 計時器
MOV    TL0,#<(2**16 - 5000)    ;設定 Timer 0 計時 5000 個
MOV    HE0,#>(2**16 - 5000)    ;脈波
SETB   TR0                ;啟動 Timer 0 開始計時
```

### 採用 Timer 0 Mode 2

```
MOV    TMOD , #00000010B    ;設定 Timer 0 的工作模式為
                                ; Mode 2 計時器
MOV    TL0,#(256 - 250)    ;設定 Timer 0 計時 250 個脈波
MOV    TH0,#(256 - 250)    ;自動載入數值
SETB   TR0                ;啟動 Timer 0 開始計時
```

 **練習 1 ... Timer 0 的 Mode 0 實習-跑馬燈**

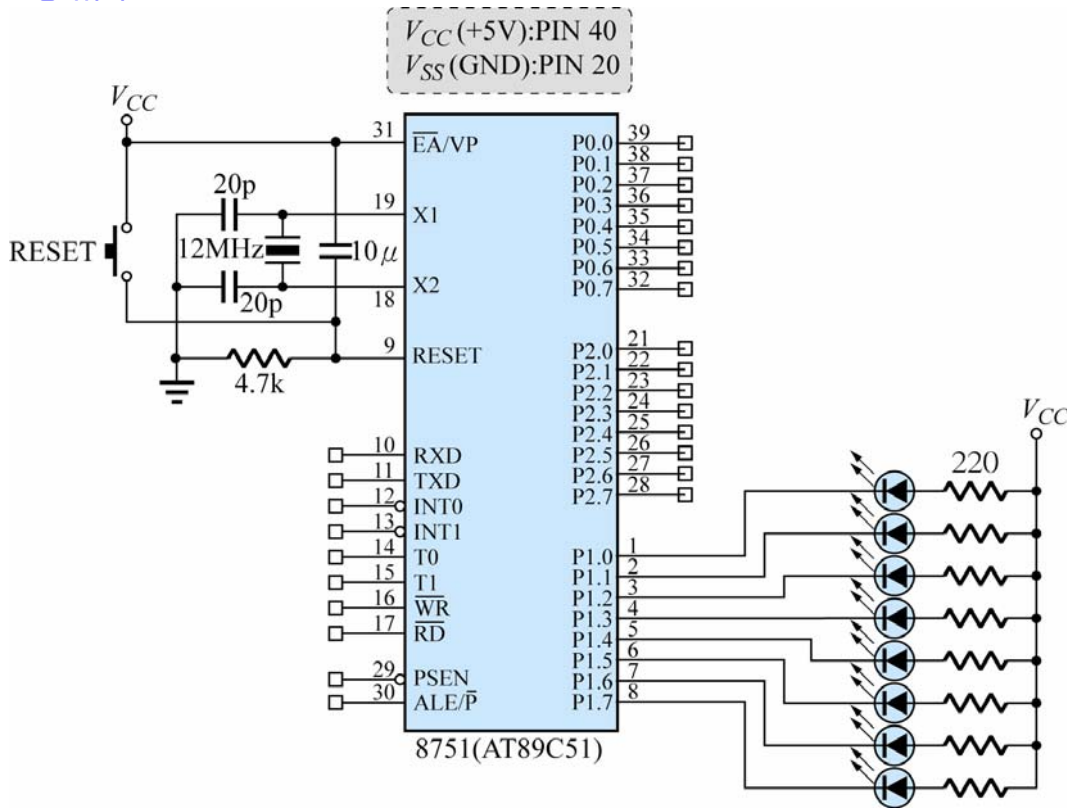
1. **學習目的：**練習 Timer 0 的 Mode 0 工作模式的使用方法
2. **功能說明**

P1 控制 8 個 LED 作跑馬燈旋轉動作（向左），每隔一秒旋轉一個 LED，延時一秒的工作由 MCS-51 內部的計時/計數器 Timer 0 負責，使用工作模式為 Mode 0。

P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
○	○	○	○	○	○	○	●
○	○	○	○	○	○	●	○
○	○	○	○	○	●	○	○
○	○	○	○	●	○	○	○
○	○	○	●	○	○	○	○
○	○	●	○	○	○	○	○
○	●	○	○	○	○	○	○
●	○	○	○	○	○	○	○
○	○	○	○	○	○	○	●
○	○	○	○	○	○	●	○
●	●	●	●	●	●	●	●
●	●	●	●	●	●	●	●

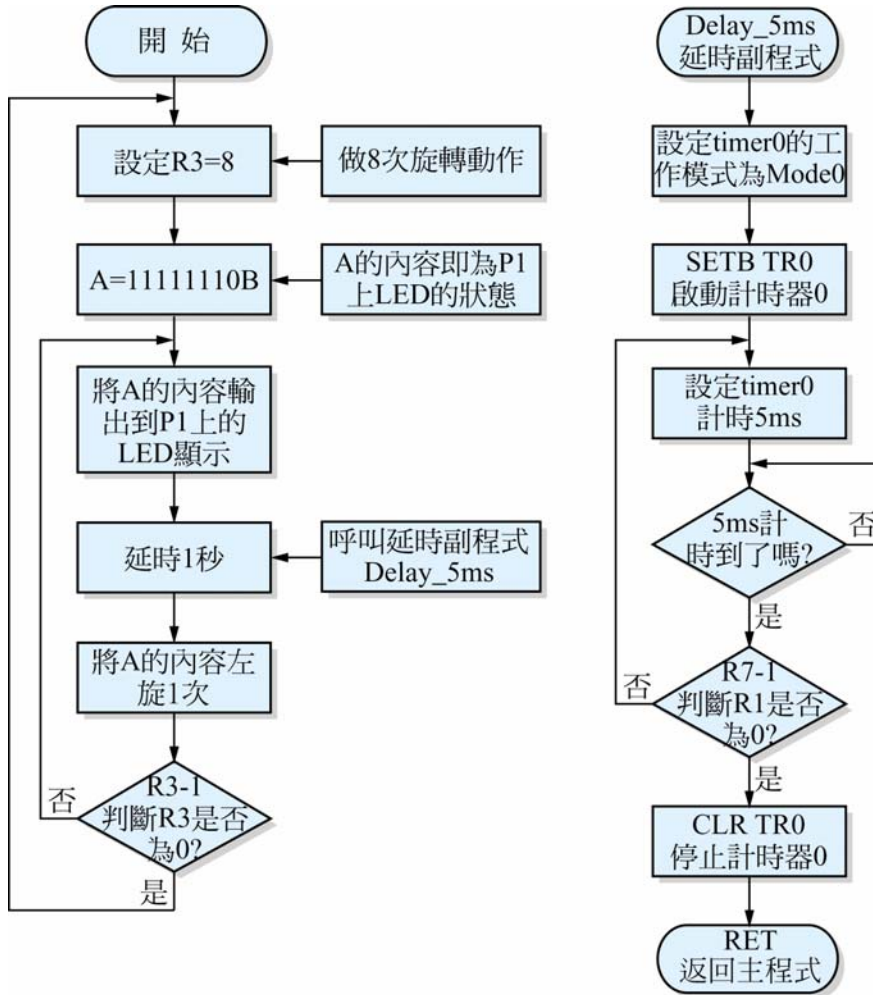
註：○代表 LED 滅 ●代表 LED 亮

3. 電路圖



● 圖 6-10 Timer 0 的 Mode 0 實習-跑馬燈

4. 流程圖



● 圖 6-11 程式 P06-1 流程圖

## 5. 程式及程式說明

；程式檔名：P06-1.asm

```

org    0
jmp    Start
Start: mov    sp,#6fh
      mov    r3,#8          ; 欲左移 8 次
      mov    a,#11111110b  ; 開始時令右邊的燈亮
Left:  mov    p1,a          ; A 的內容即為 P1 控制 LED 的顯示狀態
      mov    r7,#200        ; 設定 Timer 0 執行 200 次計時 5ms 的工作
      call   Delay_5ms      ; 因此延時時間為 200 * 5ms = 1sec
      rl     a              ; 8 個 LED 要依序亮，因此將 A 的內容旋轉
      djnz  r3,Left        ; 重複執行 8 次
      jmp   Start          ; 從頭重複執行程式

```

；延時副程式（每呼叫本副程式一次，可延時 $(r7)*5ms$ ）

；採用計時器 0 的 Mode 0

Delay\_5ms:

```

mov    tmod,#00000000b    ; 設定 Timer 0 的工作模式為 Mode 0
setb   tr0                ; 啟動計時器，開始計時

```

Timer\_Loop:

```

mov    tl0,#(2**13-5000).mod.32 ; 設定 Timer 0 計時 5000 個脈波
mov    th0,#(2**13-5000)/32     ; 計時 5000 個脈波共花
                                      ; 5000*1 $\mu$ s=5ms 時間

```

Timer\_Wait:

```

Jbc    tf0,Timer_Ok        ; Timer 0 計數完 5000 個脈波產生溢位時，會設
                                      ; 定 tf0=1，否則 tf0 維持為 0。指令 jbc tf0,
                                      ; Timer_Ok 則檢查 tf0 是否為 1，若是則先將 tf0
                                      ; 清除為 0，再跳至 Timer_Ok 位址
      jmp   Timer_Wait      ; 若未計時完(tf0=0)則跳回 Timer_Wait，繼續檢查

```

Timer\_Ok:



```

djnz   r7,Timer_Loop    ; 讓 Timer 0 執行計時 5ms 工作(r7)次
clr     tr0              ; 執行完計時 5ms 工作(r7)次後，就停止計時工作
ret                                           ; 回到主程式
.end                                           ; 程式結束

```

## 6. 討 論

程式 P06-1 中的延遲時間技巧，是利用檢查 TF0 旗號是否被設定為 1 的方法。若 Timer 0 計時尚未到達，則 TF0=0，若 TF0=1 則表示計時時間已到。而 JBC 指令是負責檢查 TF0 是否被設為 1 的工作。若檢查到 TF0=1，則 JBC 指令會先清除 TF0 旗號為 0，然後再做跳至 Timer\_Ok 的工作。清除 TF0=0，以便繼續下一次計時。

### 練習 2... Timer 0 的 Mode 1 實習-跑馬燈

1. 學習目的：練習 Timer 0 工作在 Mode 1 的用法
2. 動作說明：與練習 1 相同
3. 流程圖：參考練習 1
4. 程式及程式說明

；程式檔名：P06-2.asm

```

org     0
jmp     Start
Start:   mov     sp,#6fh
        mov     r3,#8                ; 欲左移 8 次
        mov     a,#11111110b        ; 開始時令右邊的燈亮
Left:    mov     p1,a                ; A 的內容即為 P1 控制 LED 的顯示狀態
        mov     r7,#200              ; 設定 Timer 0 執行 200 次計時 5ms 的工作
        call    Delay_5ms           ; 因此延時時間為 200 * 5ms = 1sec
        rl      a                    ; 8 個 LED 要依序亮，因此將 A 的內容旋轉
        djnz    r3,Left              ; 重複執行 8 次
        jmp     Start                ; 從頭重複執行程式

```

；延時副程式（每呼叫本副程式一次，可延時(r7)\*5ms）

；採用計時器 0(Timer 0)的 Mode 1

Delay\_5ms:

```
mov    tmod,#00000001b    ;設定 Timer 0 的工作模式為 Mode 1
setb   tr0                ;啓動計時器，開始計時
```

Timer\_Loop:

```
mov    tl0,#<(2**16-5000) ;設定 Timer 0 計時 5000 個脈波
mov    th0,#>(2**16-5000) ;計時 5000 個脈波共花 5000*1μs=5ms 時間
```

Timer\_Wait:

```
jbc    tf0,Timer_Ok      ;說明同程式 6-1
jmp    Timer_Wait        ;
```

Timer\_Ok:

```
djnz   r7,Timer_Loop    ;
clr    tr0               ;
ret    ;
.end    ;
```

## 5. 討 論

Timer X 的 Mode 1 與 Mode 0 的工作原理其實完全一樣，所差別的是 Mode 0 是一個 13 位元上數計數器，因此最大計數值是從 0000H~1FFFH，共可數 8192 個脈波。而 Mode 1 是一個 16 位元上數計數器，最大計數值是從 0000H~FFFFH，共可數 65536 個脈波。因此在實際應用上，大都採用 Mode 1 較實用。

### 練習 3...Timer 0 的 Mode 2 實習-跑馬燈

1. 學習目的：學習 Timer 0 工作在 Mode 2 的用法
2. 動作說明：與練習 1 相同
3. 流程圖：參考練習 1

#### 4. 程式及程式說明

；程式檔名：P06-3.asm

```

    org    0
    jmp    Start
Start:  mov    sp,#6fh
        mov    r3,#8           ; 欲左移 8 次
        mov    a,#11111110b   ; 開始時令右邊的燈亮
Left:   mov    p1,a           ; A 的內容即為 P1 控制 LED 的顯示狀態
        mov    r7,#200        ; 設定 Timer 0 執行 200 次計時 5ms 的工作
        call   Delay_5ms      ; 因此延時時間為 200 * 5ms = 1sec
        rl     a               ; 8 個 LED 要依序亮，因此將 A 的內容旋轉
        djnz   r3,Left        ; 重複執行 8 次
        jmp    Start          ; 從頭重複執行程式

```

；延時副程式（每呼叫本副程式一次，可延時(r7)\*5ms）

；採用計時器 0(Timer 0)的 Mode 2

Delay\_5ms:

```

    mov    tmod,#00000010b    ; 設定 Timer 0 的工作模式為 Mode 2
    setb   tr0                ; 啓動計時器，開始計時
    mov    tl0,#(2**8-250)    ; 設定 Timer 0 計時 250 個脈波

```

Timer\_Loop1:

```

    mov    r6,#20             ; 設定 Timer 0 執行 20 次計時 250 $\mu$ s 的工作
                                ; 因此計時完 20 次共花 20*250 $\mu$ s=5ms 的時間

```

Timer\_Loop2:

```

    mov    th0,#(2**8-250)    ; 計時 250 個脈波共花 250*1 $\mu$ s=250 $\mu$ s 時間

```

Timer\_Wait:

```

    jbc    tf0,Timer_Ok       ; 說明同程式 P06-1.asm
    jmp    Timer_Wait         ;

```

Timer\_Ok:

```

    djnz   r6,Timer_Loop2     ; 執行計時 250 $\mu$ s 工作(r6)次
    djnz   r7,Timer_Loop1     ; 執行計時 5ms 工作(r7)次
    clr    tr0                 ; 停止計時工作
    ret                                ;
    .end                          ;

```

## 5. 討 論

Timer X 的 Mode 2 是 "自動重新載入" 模式。當您將計時時間預設給 TL0 後，TL0 計時計數器則根據這個值而開始計時，當計時到了，產生溢位後，TL0 的內容變成 00H，也就是先前您預先設給 TL0 的預設值，已經不見了。如果您希望計時時間都固定的話，那就必須將您所希望的值再重新放入 TL0 中。而 "自動重新載入" 模式，就是具有上述可幫您自動重新設定 TL0 內容值工作的能力，只要您將欲加給 TL0 的值，預先放入 TH0 中，則當 TL0 計時時間到了之後，就會自動地將 TH0 的內容值載入到 TL0 中，然後 TL0 再根據此值繼續計時。

自動重新載入有個好處，就是可以讓計時器計時更準確，例如在練習 2 中，Timer 0 每次計時 5000 個脈波，若要延時 0.5 秒，必須要 Timer 0 執行 100 次（每次 5ms）。但是因為 Mode 0、Mode 1 並不具備計時到了重新載入設定值的功能，因此重新載入的工作，必須由程式做，如此一來就會造成軟體干擾硬體的計時工作，而使計時器計時不準確，如果重新載入的動作由硬體致能自動載入，程式就不需做載入的工作，而使得計時時間更準確。

Mode 2 是將 TLX 用來計時，另一個 THX 用來存放設定值。因此 Mode 2 只有一個 8 位元的計時器，最大可計數脈波數只有 00~FFH 個。如此的計時計數範圍多半是太小了。在 MCS-52 裡的 Timer 2 則為了彌補此缺點，增加有一個 16bit 的自動重新載入功能。

## 練習 4... 由外部 T1 腳輸入計數脈波實習

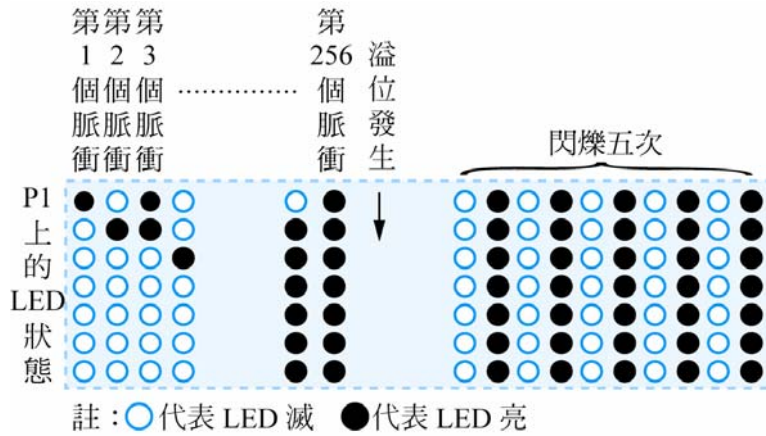
### 1. 學習目的

學習在程式中，同時啟用 Timer 0 和 Timer 1 的用法及由 T1 腳輸入計數脈波的方法。

### 2. 動作說明

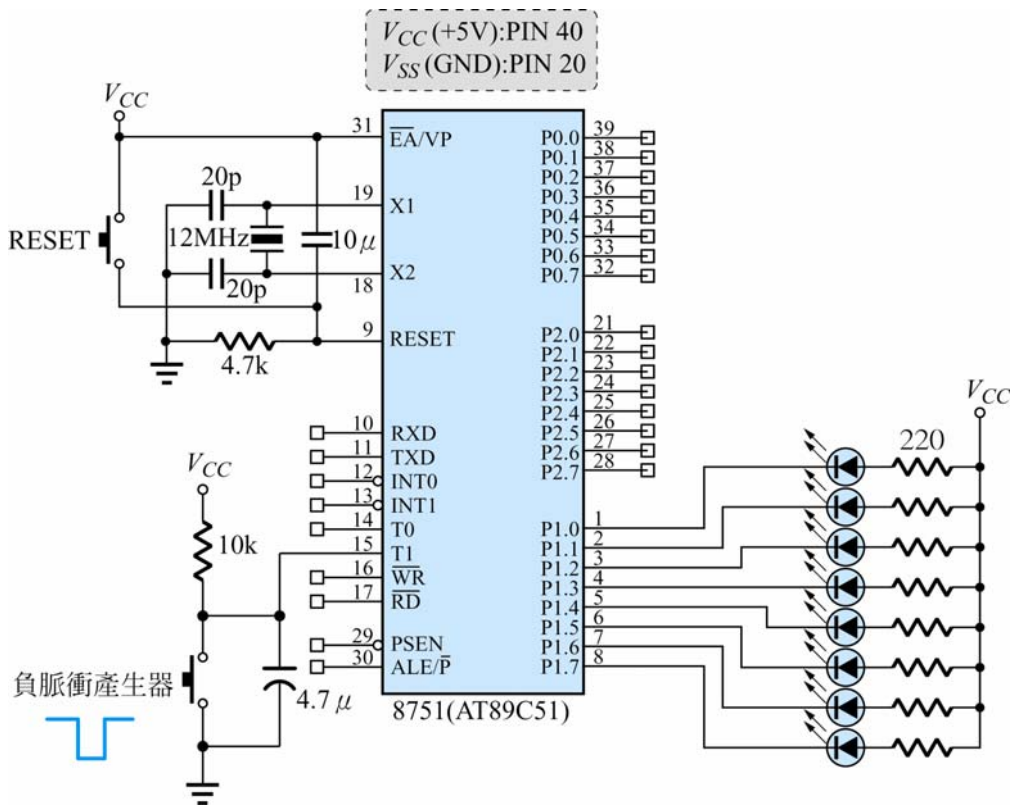
參考電路圖，本練習是利用 Timer 1 的 Mode 2 計數功能，接受自外部 T1 輸入的脈衝信號，當 T1 腳產生一個負脈衝時，在 Timer 1 的計數暫存器會增加 1，並且將 TL1 的內容（目前已累計的計數脈波數）由 P1 上的 LED 顯示出來，一直到 Timer 1 的計數脈波數超過 256 個之後產生

溢位，則停止 Timer 1 的計數工作 (CLR TR1)。並且由 P1 上的 8 個 LED 同時顯示亮→滅→亮→滅等閃爍五次，每次間隔 0.5 秒的時間，此 0.5 秒的延時工作，則交由 Timer 0 的 Mode 0 負責執行。



● 圖 6-12 P1 腳上的 LED 顯示 T1 腳輸入脈波數顯示狀態圖

### 3. 電路圖



● 圖 6-13 由外部 T1 腳輸入計數脈波實習

#### 4. 程式及程式說明

；程式檔名：P06-4.asm

```

    org      0
    jmp      Start
Start:  mov    sp,#6fh
        mov    tmod,#01100000b    ;設定 Timer 1 的工作模式為 Mode 2，且脈波
                                    ;源來自外部 T1 腳的輸入脈波，並且設定
                                    ;Timer 0 的工作模式為 Mode 0，且脈波源來
                                    ;自內部晶體振盪 Fosc/12

Loop:
        mov    th1,#0              ;設定 th1 初始值由 0 開始計數
        mov    tl1,#0              ;設定 tl1 初始值由 0 開始計數
        setb   tr1                  ;開始接受來自 T1(P3.5)腳的輸入脈波

Wait:
        Jbc    tf1,OverFlow        ;判斷 tf1 是否為 1，若 tf1=1 代表 Timer 1 計
                                    ;數器已計數溢位（256 個脈波輸入）則跳至
                                    ;OverFlow

        mov    a,tl1                ;將目前的計數個數 TL1 的內容值，分別輸出
        cpl    a                    ;到 P1 的 LED 上顯示出來，因為 LED 亮
                                    ;代表"0"，滅代表"1"，因此 TL1 輸出之前必
                                    ;須先反相
        mov    p1,a                 ;須先反相
        jmp    Wait

OverFlow:
        clr    tr1                  ;停下 T1 腳上脈波輸入計數工作
        mov    r3,#5                ;令 P1 腳上的 LED 亮、滅閃爍 5 次

Flash_Led:
        mov    p1,#0h               ;令 P1 腳上的 LED 全亮
        mov    r7,#100              ;設定 Timer 0 執行 100 次計時 5ms 的工作
        call   Delay_5ms            ;因此延時時間為 100 * 5ms = 0.5sec

```

```
mov    p1,#0ffh          ; 令 P1 腳上的 LED 全滅
mov    r7,#100           ;
call   Delay_5ms        ; 延時 0.5 秒
djnz   r3,Flash_Led     ;
jmp    Loop              ; 回到 Loop 重新計數

; ==    延時副程式    ==
; 每呼叫本副程式一次，可延時(r7)*5ms
; 採用計時器 0(Timer 0)的 Mode 0

Delay_5ms:
    setb    tr0          ; 啓動計時器，開始計時

Timer_Loop:
    mov    tl0,#(2**13-5000).mod.32 ; 設定 Timer 0 計時 5000 個脈波
    mov    th0,#(2**13-5000)/32    ; 計時 5000 個脈波共花 5000*1 $\mu$ s=5ms 時間

Timer_Wait:
    jbc    tf0,Timer_Ok      ; Timer 0 若計數完 5000 個脈波產生溢位，
                            ; 會設定旗號 tf0=1，否則 tf0 維持為 0，
                            ; 指令 jbc tf0,Timer_Ok 則檢查 tf0 是否為
                            ; 1，若是則先將 tf0 清除為 0，再跳至
                            ; Timer_Ok 位址
    jmp    Timer_Wait        ; 若未計時完(tf0=0)則跳回 Timer_Wait，繼
                            ; 續檢查

Timer_Ok:
    djnz   r7,Timer_Loop    ; 讓 Timer 0 執行計時 5ms 工作(r7)次
    clr    tr0              ; 當 Timer 0 執行完計時 5ms 工作(r7)次後
                            ; 就停止計時工作
    ret                               ; 回到主程式
.end                               ; 程式結束
```



## 學後評量 ●●●

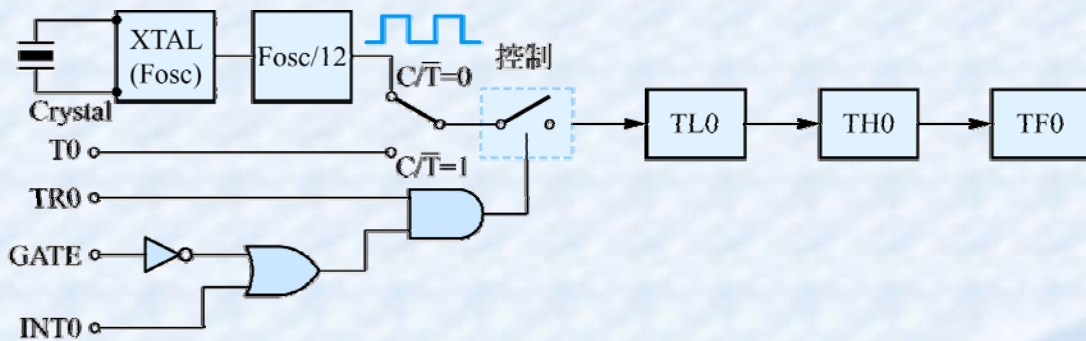
### 一、選擇題

- ( ) 1. MCS-51 內部有二個多少位元計時/計數器？ (A)7 位元 (B)8 位元 (C)16 位元 (D)32 位。
- ( ) 2. MCS-51 的計時/計數模式中模式 1(mode1)是 (A)13 位元計時/計數模式 (B)16 位元計時/計數模式 (C)自動載入功能 (D)兩個獨立的 8 位元計時/計數器。
- ( ) 3. 根據圖(1)timer0 計時/計數器控制圖，下列那一個條件可以啟動計時器？  
 (A)GATE=0，TR0=1 及  $C/\bar{T}=1$   
 (B)GATE=1，TR0=1 及  $C/\bar{T}=1$   
 (C)GATE=1，TR0=1，INT0=1 及  $C/\bar{T}=0$   
 (D)GATE=1，TR0=1，INT0=1 及  $C/\bar{T}=1$
- ( ) 4. 根據圖(1)timer0 計時/計數器控制圖，下列那一個條件可以啟動計數器？  
 (A)GATE=0，TR0=1 及  $C/\bar{T}=1$   
 (B)GATE=1，TR0=1 及  $C/\bar{T}=1$   
 (C)GATE=0，TR0=1，INT0=1 及  $C/\bar{T}=0$   
 (D)GATE=1，TR0=1，INT0=1 及  $C/\bar{T}=0$
- ( ) 5. 根據圖(1)timer0 計時/計數器控制圖，Crystal 接 12MHz，如何設計程式讓 timer0 計時 5ms？  
 (A) 

```
mov    tmod,#00000000b
mov    tl0,#(2**13-5000).mod.256
mov    th0,#(2**13-5000) / 256
setb   tr0
```



- (B) `mov tmod,#00000000b`  
`mov tl0,#(2**16-5000).mod.32`  
`mov th0,#(2**16-5000) / 32`  
`setb tr0`
- (C) `mov tmod,#00000000b`  
`mov tl0,#(2**8-5000).mod.32`  
`mov th0,#(2**8-5000) / 32`  
`setb tr0`
- (D) `mov tmod,#00000000b`  
`mov tl0,#(2**13-5000).mod.32`  
`mov th0,#(2**13-5000) / 32`  
`setb tr0`



● 圖(1)

( ) 6. 根據圖(1)timer0 計時/計數器控制圖，如何設計程式讓 timer0 計數 5000 脈波？

- (A) `mov tmod,#00000001b`  
`mov tl0,#(2**13-5000).mod.32`  
`mov th0,#(2**13-5000) / 32`  
`setb tr0`
- (B) `mov tmod,#00000001b`  
`mov tl0,#(2**16-5000).mod.256`  
`mov th0,#(2**16-5000) / 256`

```

    setb    tr0
(C)  mov    tmod,#00000001b
    mov    tl0,#(2**16-5000).mod.32
    mov    th0,#(2**16-5000) / 32
    setb    tr0
(D)  mov    tmod,#00000000b
    mov    tl0,#(2**8-5000).mod.16
    mov    th0,#(2**8-5000) / 16
    setb    tr0

```

( ) 7. 設計一個延時 0.5 秒的延時副程式

```

(A) delay:  mov    r7,#200
           mov    tmod,#00000000b
ready:     mov    tl0,#(8192-5000).mod.32
           mov    th0,#(8192-5000)/32
           setb    tr0
wait:      jnb    tf0,wait
           clr    tf0
           djnz   r7,ready
           ret
(B) delay:  mov    r7,#100
           mov    tmod,#00000000b
ready:     mov    tl0,#(65536-5000).mod.256
           mov    th0,#(65536-5000)/256
           setb    tr0
wait:      jnb    tf0,wait
           clr    tf0
           djnz   r7,ready
           ret

```

```

(C) delay:  mov    r7,#100
            mov    tmod,#00000000b
ready:     mov    tl0,#(8192-5000).mod.32
            mov    th0,#(8192-5000)/32
            setb   tr0
wait:     jnb    tf0,wait
            clr    tf0
            djnz  r7,ready
            ret

(D) delay:  mov    r7,#200
            mov    tmod,#00000000b
ready:     mov    tl0,#(65536-5000).mod.256
            mov    th0,#(65536-5000)/256
            setb   tr0
wait:     jnb    tf0,wait
            clr    tf0
            djnz  r7,ready
            ret

```

相關暫存器名稱

暫存器	位元 7	位元 6	位元 5	位元 4	位元 3	位元 2	位元 1	位元 0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0
SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
PCON	SMOD	—	—	—	GF1	GF0	PD	IDC
IE	EA	—	ET2	ES	ET1	EX1	ET0	EX0
IP	—	—	PT2	PS	PT1	PX1	PT0	PX0

## 二、實作題

1. 設計一個移動速度為 0.5 秒跑馬燈，延時時間的工作利用 Timer 1 的 Mode 1 來計時產生。
2. 利用 Timer 0 的 Mode 0 設計一個產生 1kHz 聲音，聲音則由 P1.0 接 BUZZER 產生。
3. 利用 Timer 0 的 Mode 1 設計一個 19 秒倒數計時器，其中十位數以 LED 亮滅代表 0 或 1，個位數則以七段顯示器顯示。
4. 使用兩個七段顯示器，設計一個顯示目前按鍵次數的計數器，計數範圍為 00~99，其按鍵次數脈波由 T0 腳輸入產生。
5. 利用 Timer 1 的 Mode 1 設計一個量測按鍵押下時間(以秒為單位)的計時器，由 P1 的 LED 顯示出來，不足一秒以一秒為單位。

