

步進馬達之控制

本章內容豐富，主要包括三部分：

● 硬體部分：

介紹了步進馬達的結構、驅動方式，及如何應用 8051 來控制步進馬達等。

● 程式與實作部分：

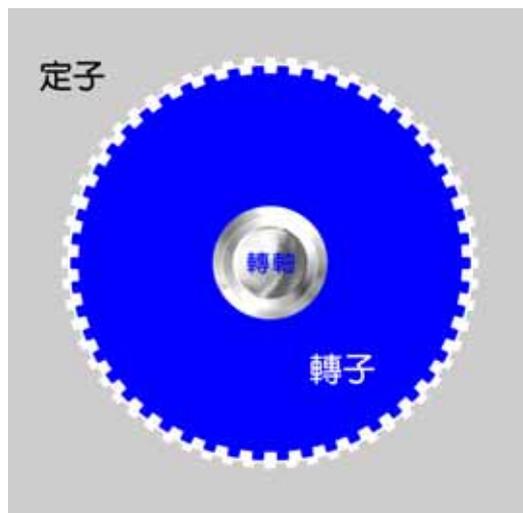
步進馬達的 1 相驅動、2 相驅動與 1-2 相驅動程式的應用。

步進馬達的角度控制、速度控制等。

9-1 認識步進馬達

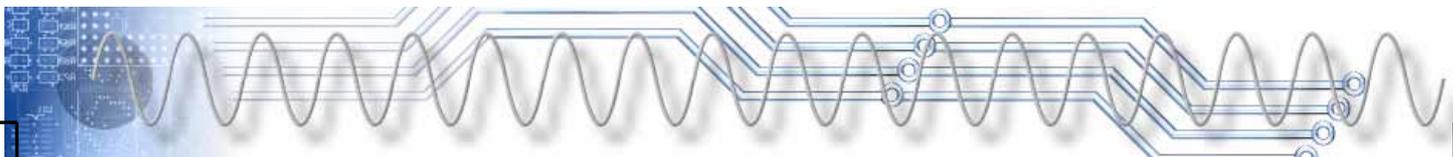
步進馬達 (stepping motor) 是一種以脈波控制的轉動裝置，由於是以脈波驅動，很適合以數位或微電腦來控制，所以我們又把它當成是一種數位裝置。

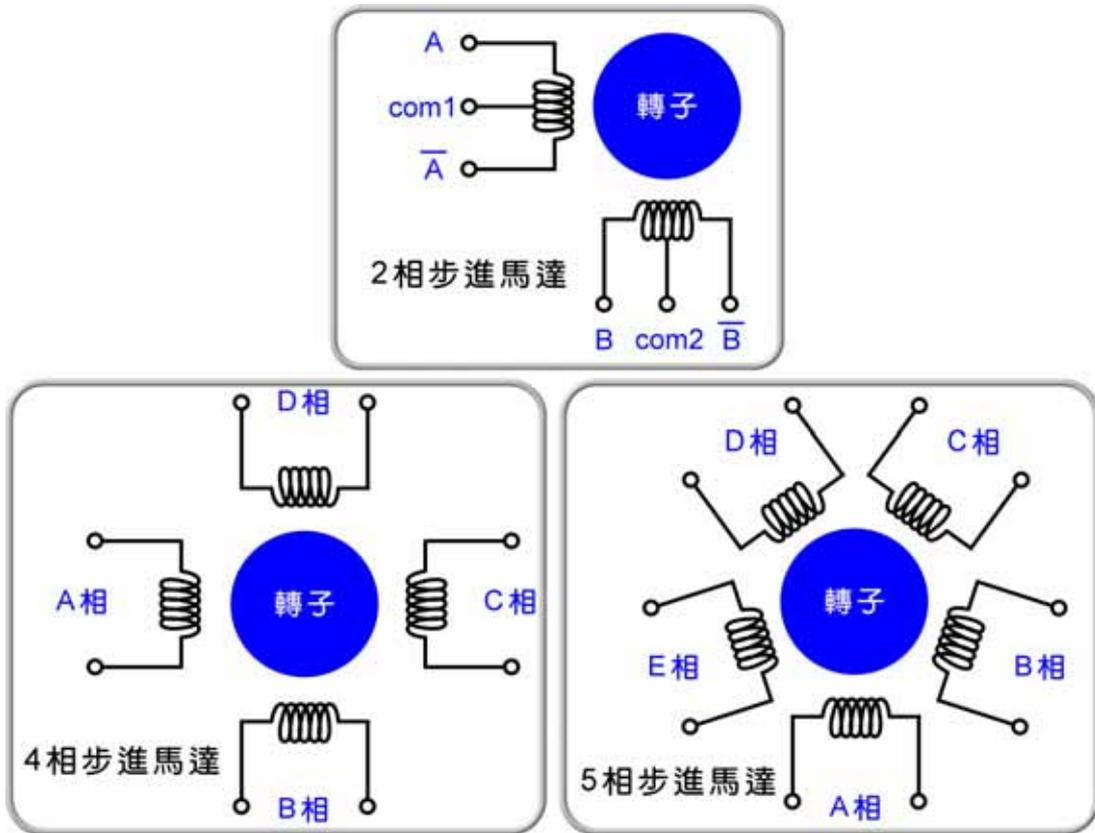
9-1-1 步進馬達的結構



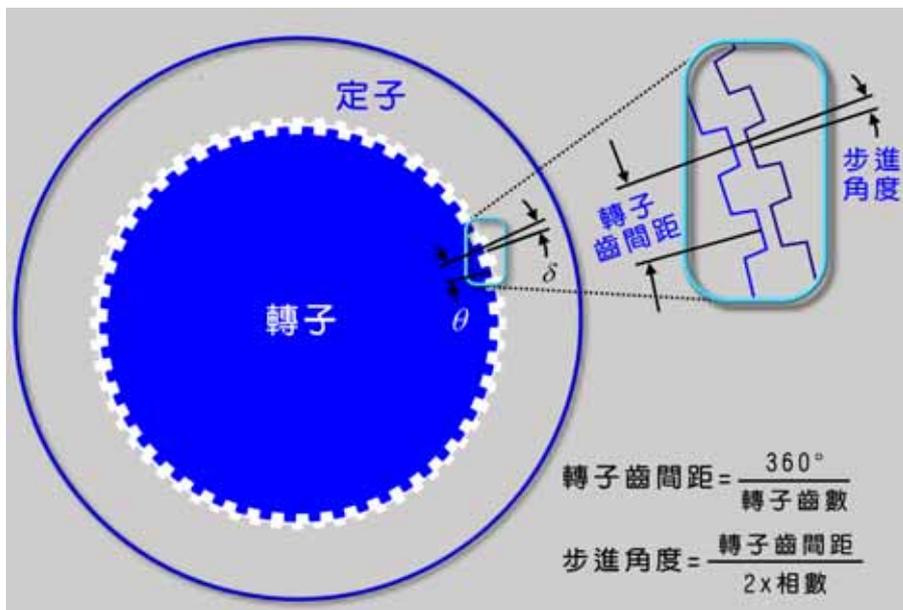
(圖1) 步進馬達之基本架構

步進馬達與一般馬達結構類似，除了托架、外殼之外，就是轉子與定子，比較特殊的是其轉子與定子上有許多細小的齒。而其轉子為永久磁鐵，線圈是繞在定子上。依線圈的配置，可分為 2 相、4 相、5 相等，如圖 2 所示。比較常用的是 2 相的步進馬達，其中包括兩組具有中間抽頭的線圈，A、com1 與 \bar{A} 為一組，B、com2 與 \bar{B} 為另一組。2 相 6 線式步進馬達，其連接線就是 A、com1、 \bar{A} 、B、com2 與 \bar{B} ；而 2 相 5 線式步進馬達是將其中的 com1 與 com2 連接。另外，4 相步進馬達是由四組線圈所構成，5 相步進馬達是由五組線圈所構成。



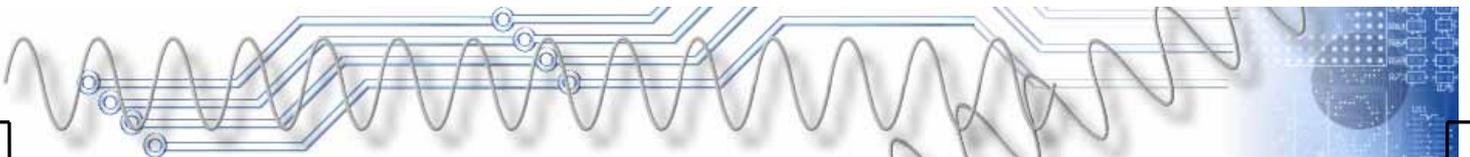


(圖2) 步進馬達的種類



(圖3) 步進馬達的齒間距

顧名思義，步進馬達就是一步步走的馬達，而其轉子與定子的齒，



決定其每步的間距，若轉子上有 N 個齒，則其齒間距 θ 為

$$\theta = \text{轉子齒間距} = \frac{360^\circ}{N}$$

而步進角度 δ 為

$$\delta = \frac{\text{轉子齒間距}}{2 \times \text{相數}} = \frac{\theta}{2P}$$

以常用的 2 相式 50 齒步進馬達為例，

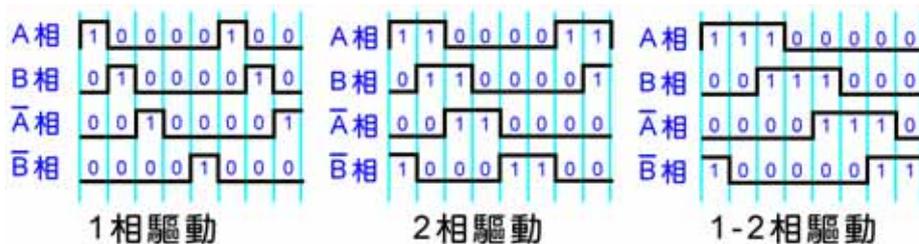
$$\theta = 360^\circ / 50 = 7.2^\circ$$

$$\delta = 7.2^\circ / (2 \times 2) = 1.8^\circ$$

另外一種比較簡便的說法，就是以步數來表示，以 200 步的步進馬達為例，200 步為一圈 (360°)，則每步 1.8° 。

9-1-2 步進馬達的動作

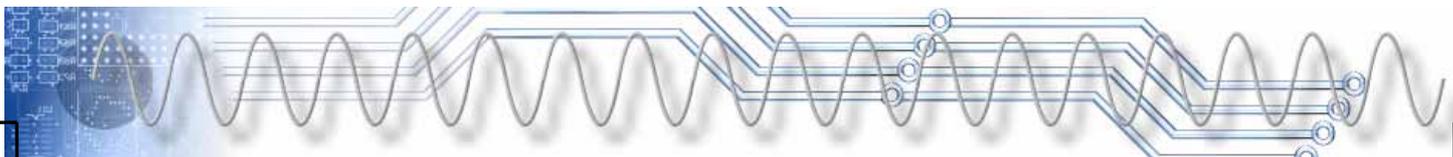
簡單講，步進馬達的動作是靠定子線圈激磁後，將鄰近轉子上相異磁極吸引過來。因此，線圈排列的順序，以及激磁信號的順序就很重要！以 2 相式步進馬達為例，其驅動信號有 1 相驅動、2 相驅動與 1-2 相驅動三種，如下圖所示：



(圖4) 步進馬達的驅動方式

1 相驅動

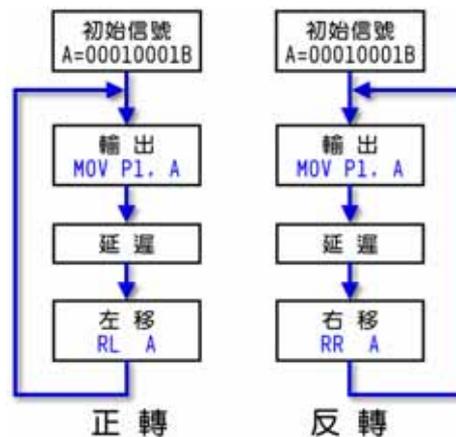
1 相驅動的方式是任何一個時間，只有一組線圈被激磁，其它線圈在休息，因此，其所產生的力矩較小。但，這種激磁方式最簡單，其信號依序為：



1000⇒0100⇒0010⇒0001⇒1000...(正轉)

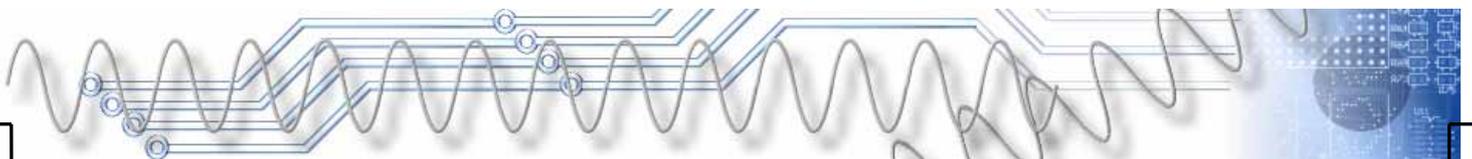
1000⇒0001⇒0010⇒0100⇒1000...(反轉)

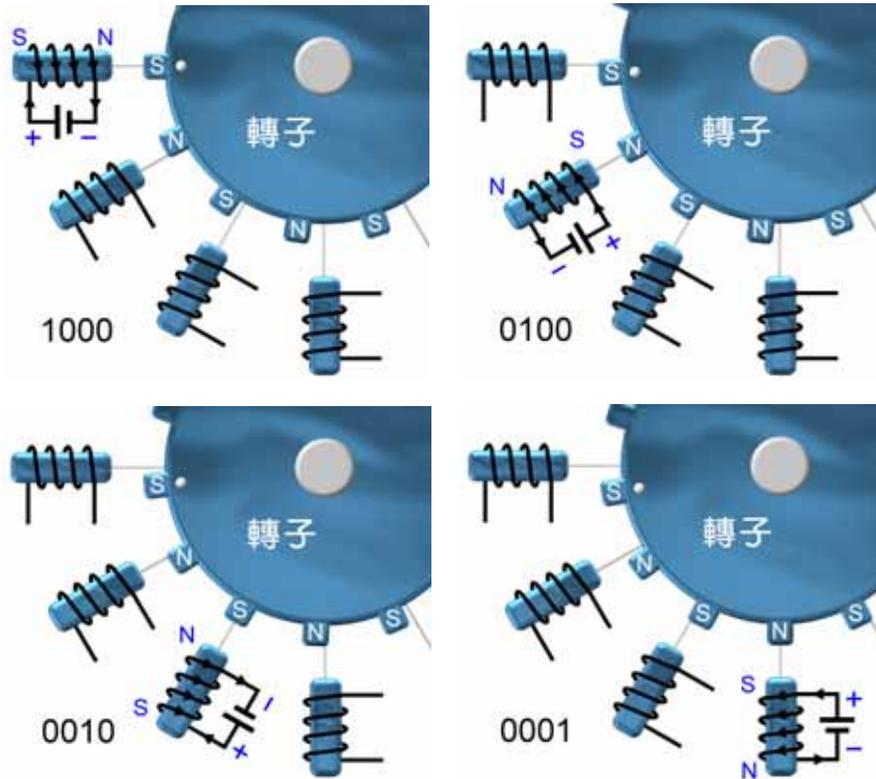
總共有四種不同的信號，呈現週期性的變化，在 8051 裡，若要產生這種信號，可先輸出「00010001B」，即「11H」。經過一小段的時間延遲，讓步進馬達有足夠的時間建立磁場及轉動，再以「RL A」或「RR A」指令將「00010001B」左移(或右移)，使之變為「00100010B」(或「10001000B」)，再輸出即可，如下流程圖是由 PORT 1 輸出。



(圖5) 1相驅動信號之產生

另外，四條信號線只要由 P1.0 到 P1.3(或 P1.4 到 P1.7)輸出就好了，為何初始信號是「00010001B」，而不是「00000001B」或「00010000B」呢？答案是為了簡化程式的，左移(或右移)時，不必判斷是否已超出範圍。為了方便起見，在此依序以 16 進位方式，列出這四個信號，即「11H、22H、44H、88H」。若將這四個信號，依序加入步進馬達，其反應如下圖所示：





(圖6) 1相驅動步進馬達之動作

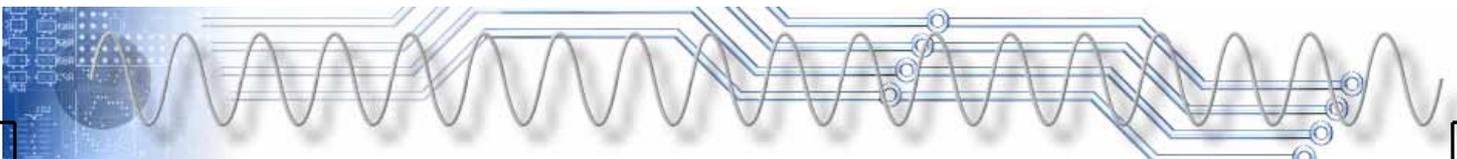
2相驅動

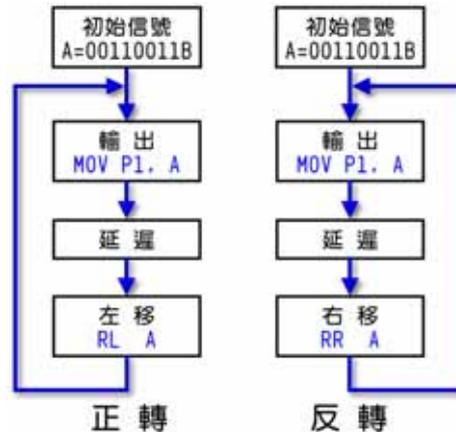
2相驅動的方式是任何一個時間，有兩組線圈同時被激磁，因此，其所產生的力矩比1相驅動來得大。儘管如此，這種激磁方式也很簡單，其信號依序為：

1100⇒0110⇒0011⇒1001⇒1100...(正轉)

1100⇒1001⇒0011⇒0110⇒1100...(反轉)

總共有四種不同的信號，呈現週期性的變化，在8051裡，若要產生這種信號，可先輸出「00110011B」，即「33H」。經過一小段的時間延遲，讓步進馬達有足夠的時間建立磁場及轉動，再以「RL A」或「RR A」指令將「00110011B」左移(或右移)，使之變為「01100110B」(或「10011001B」)，再輸出即可，如下流程圖是由PORT 1輸出。





(圖7) 2相驅動信號之產生

由上述可得知，1相驅動信號的產生與2相驅動信號的產生，除了初始信號外，完全一樣。為了方便起見，在此依序以16進位方式，列出這四個信號，即「33H、66H、CCH、99H」。

1-2 相驅動

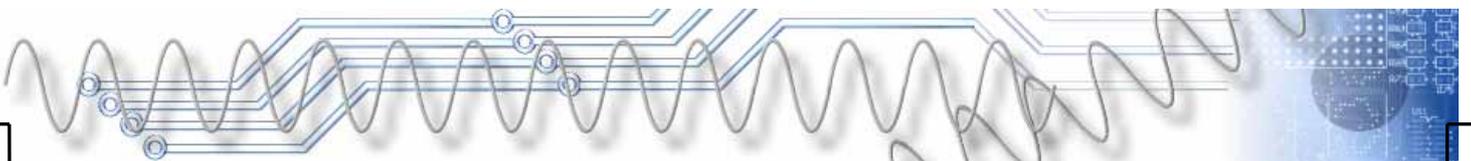
1-2相驅動的方式又稱為「半步驅動」，每個驅動信號只驅動半步。而其驅動信號依序為：

1001⇒1000⇒1100⇒0100⇒0110⇒0010⇒0011⇒0001 (正轉)

1001⇒0001⇒0011⇒0010⇒0110⇒0100⇒1100⇒1000 (反轉)

總共有八種不同的信號，呈現週期性的變化，仔細觀察可發現其中的信號是將1相驅動信號與2相驅動信號混合而成。為了方便起見，在此依序以16進位方式，列出這四個信號，即「99H、88H、CCH、44H、66H、22H、33H、11H」，其中藍色字為2相驅動信號、黑色字為1相驅動信號。

在8051裡，最好是利用查表法，將這八個信號放入表格，再依序讀出、輸出。當然，兩個信號輸出之間，還是須要經過一小段的時間延遲，讓步進馬達有足夠的時間建立磁場及轉動。



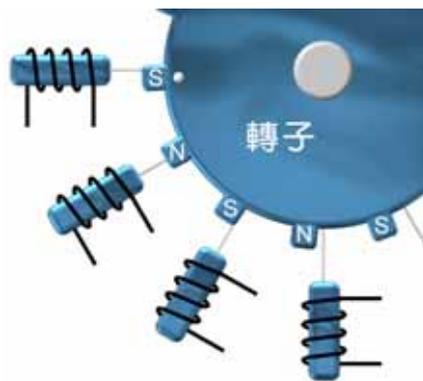


(圖8) 1-2 相驅動信號之產生

9-1-3 步進馬達之定位

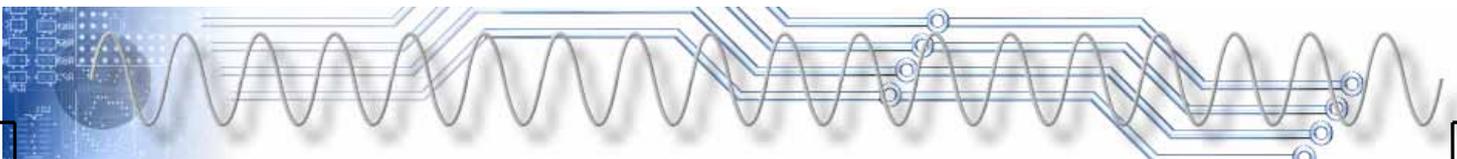
當我們開啓個人電腦時，則電腦上的軟磁機會動一下、連接該電腦的週邊設備，也會有所反應。不管是動一下、閃一下或反應一下，就是 RESET 的動作，讓所有裝置歸為一定的狀態。若說步進馬達是一種數位輸出裝置，則這種裝置在使用之前必須歸零或定位，才能精確的使用這個步進馬達。在圖 6 裡，將「1000、0100、0010、0001」信號加入步進馬達時，該步進馬達將逆時鐘轉動 4 步，若每步為 1.8° ，則總共轉了 7.2° 。同樣的驅動信號，如果一開始步進馬達的轉子位置不對，則可能發生下列兩種非預期狀態：

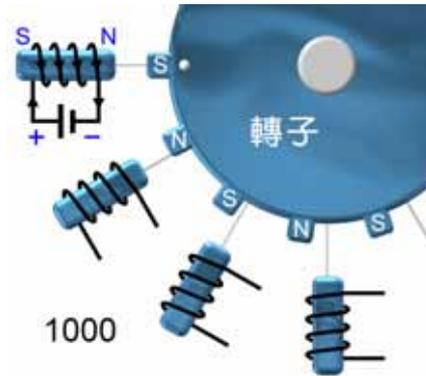
● 先順時鐘轉再逆時鐘



(圖9) 原始轉子位置

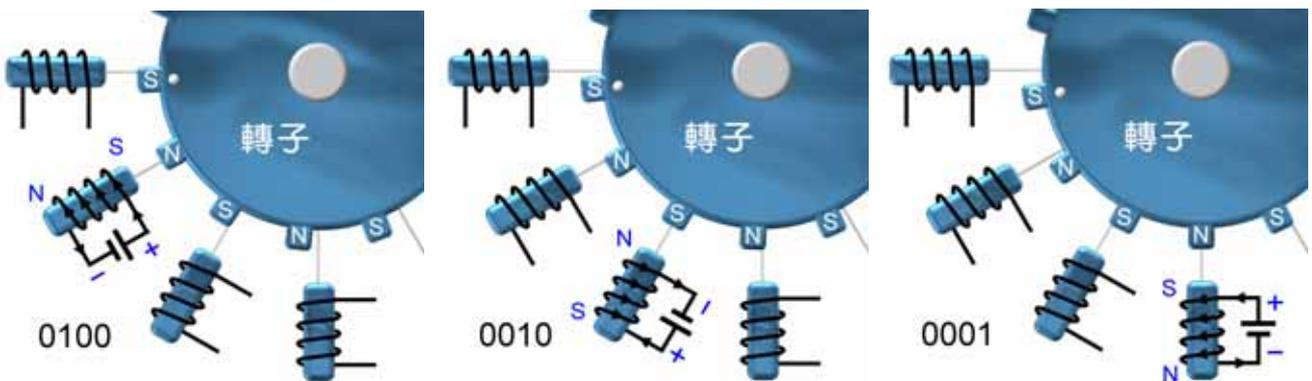
如上圖所示，送入「1000」信號時，步進馬達順時鐘旋轉 1.8° ，如下圖所示：





(圖10) 順時鐘旋轉 1.8°

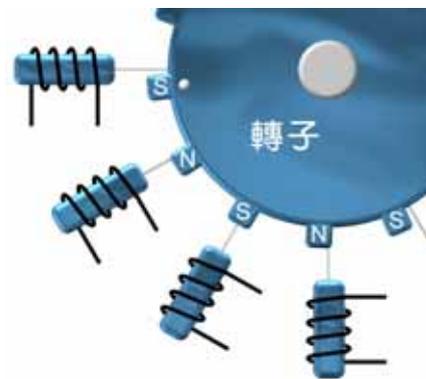
從第二組信號(「0100」、「0010」、「0001」)，才開始逆時鐘旋轉，如下圖所示：



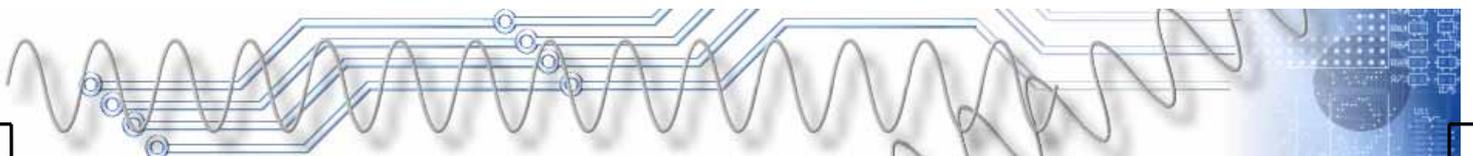
(圖11) 逆時鐘旋轉 5.4°

如此一來，總共逆時鐘旋轉 3.6°，而非預期的逆時鐘旋轉 7.2°。

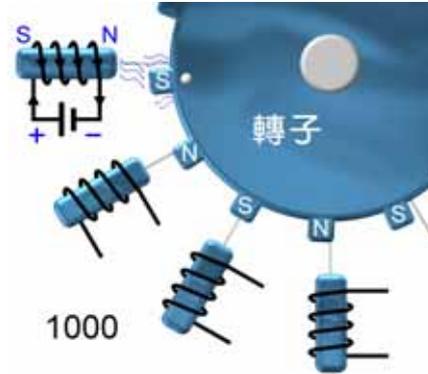
● 先抖動、順時鐘轉再逆時鐘



(圖12) 原始轉子位置

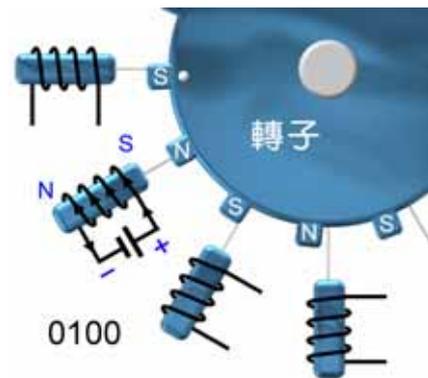


如上圖所示，送入「1000」信號時，步進馬達抖動，如下圖所示：



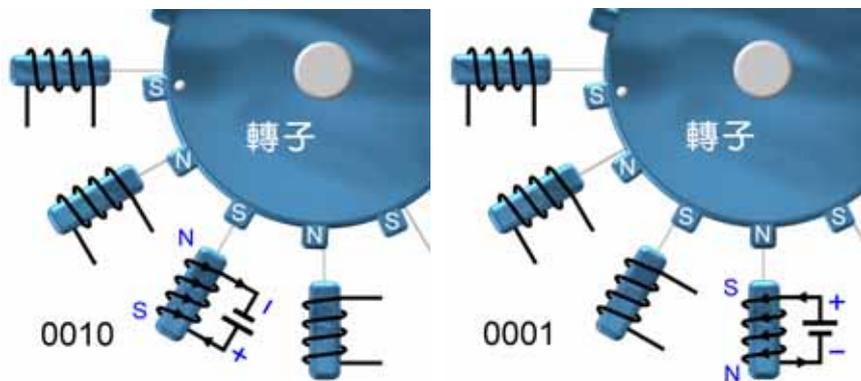
(圖13) 吸不過來

當第二組信號(「0100」)時，步進馬達順時鐘旋轉 1.8°，如下圖所示：

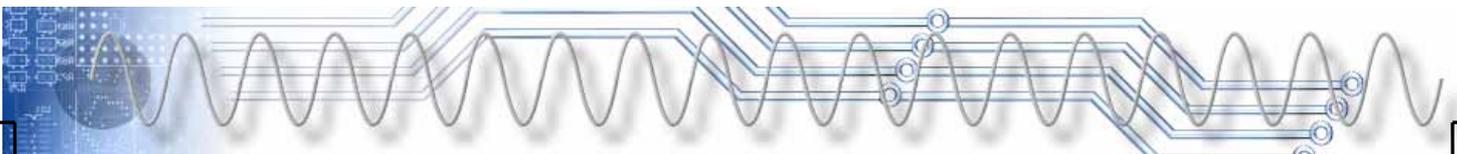


(圖14) 順時鐘旋轉 1.8°

從第三組信號(「0010」、「0001」)，才開始逆時鐘旋轉，如下圖所示：



(圖15) 逆時鐘旋轉 3.6°



如此一來，總共逆時鐘旋轉 1.8° ，而非預期的逆時鐘旋轉 7.2° 。

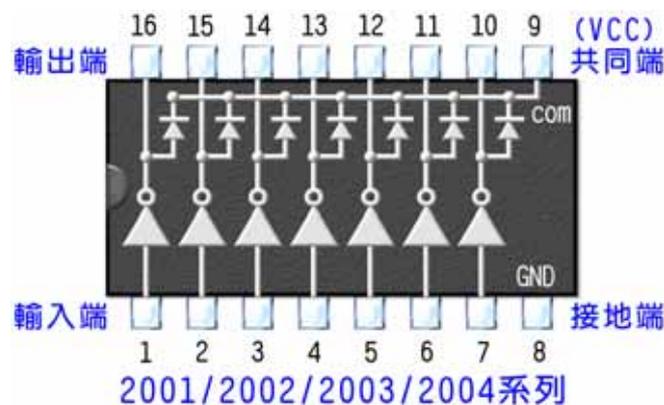
對於上述非預期狀態的產生，最簡單的防制方法是在開始運作之前，先送出一組信號，換言之，若是 1 相或 2 相驅動，則依序送出四個驅動信號；若是 1-2 相驅動，則依序送出八個驅動信號，即可正確地抓住此步進馬達的位置，稱之為定位或歸零。

9-2 步進馬達驅動電路

8051 之輸出電流很難驅動步進馬達，必須另外設置驅動電路才行，在此將介紹幾種常用的驅動電路。

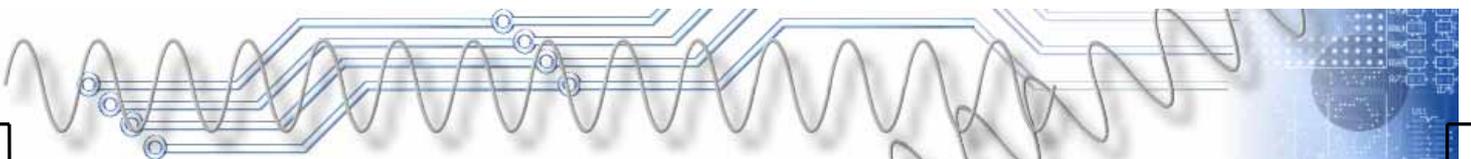
9-2-1 小型步進馬達之驅動電路

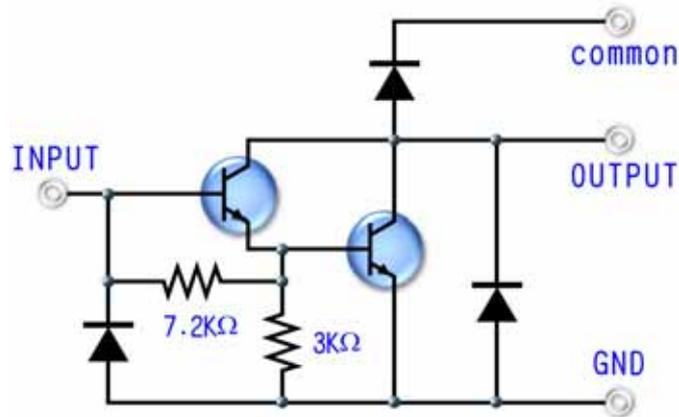
對於電流小於 0.5 安培的步進馬達，可以採用 UNL2003 之類的驅動 IC 是一種「小而美」的驅動裝置，可不要看它只是個一般包裝的 IC，它所提供的輸出電路(吸入)可達 0.5 安培呢！如下圖所示為 2003 系列驅動 IC 的接腳圖：



(圖16) 2003 系列驅動 IC 的接腳圖

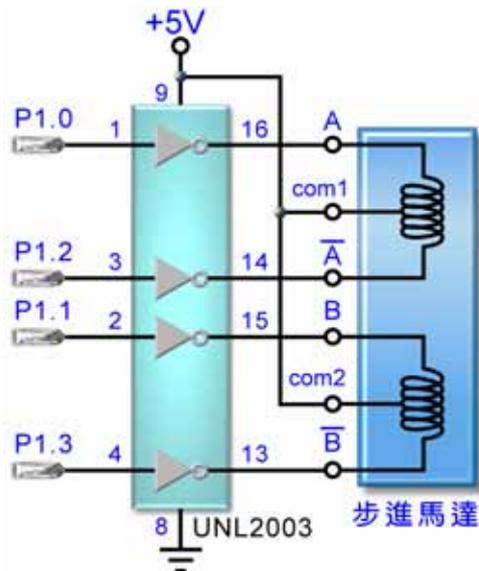
一顆 2003 系列 IC 包含 7 個開集極式輸出的反相器，而在每個輸出端都有一個連接到共同端(VCC)的二極體，做為放電保護電路，每組反相器的內部電路如下圖所示：



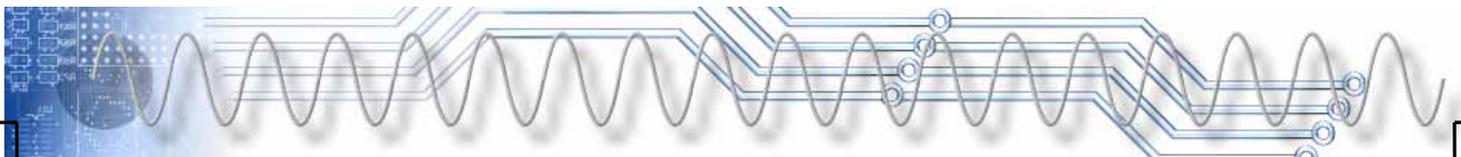


(圖17) 內部電路

如上圖所示，基本上這是一個附有保護二極體的達靈頓電路，它可提供最高 0.5 安培的電流。對於步進馬達而言，可能會有瞬間大電流，但每個驅動電路所提供的工作週期都很低，驅動 IC 應該不會過熱。如下圖所示，驅動信號由 8051 的 P1.0~P1.3 連接到 2003 的四個反相器輸入端，而其輸出端連接到步進馬達。2003 的 common 端與步進馬達的 com1、com2，連接到+5V 的電源上，而 2003 的第 8 腳接地。

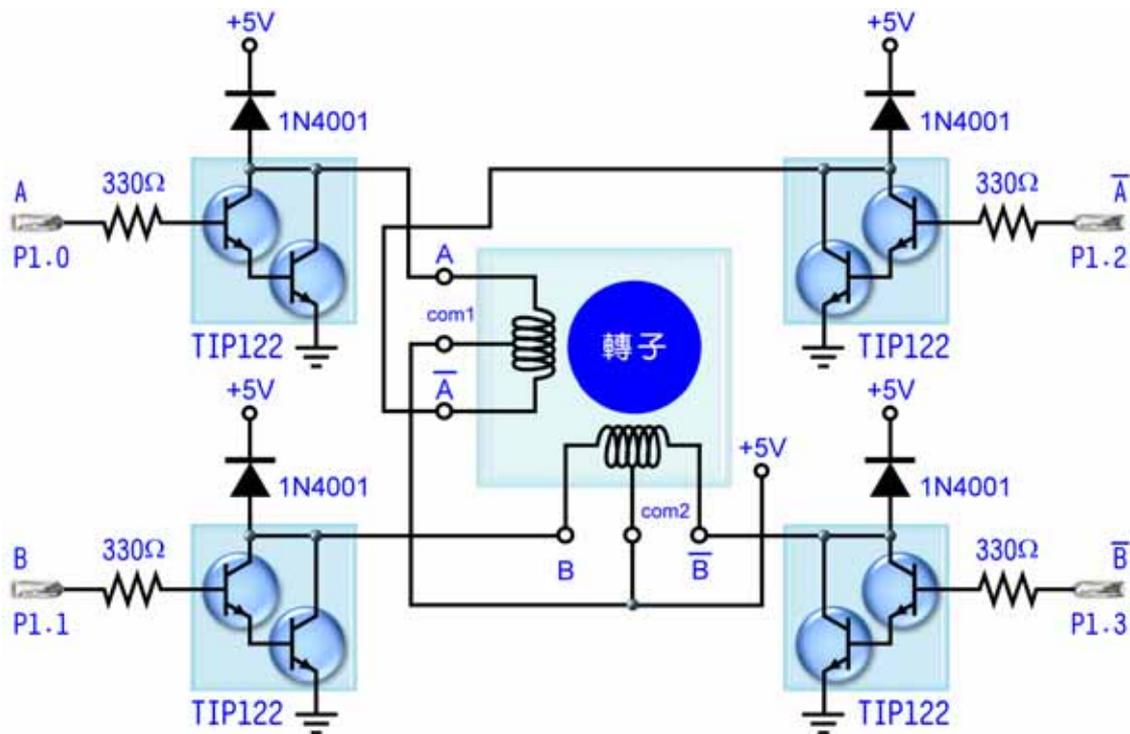


(圖18) 2003 驅動電路



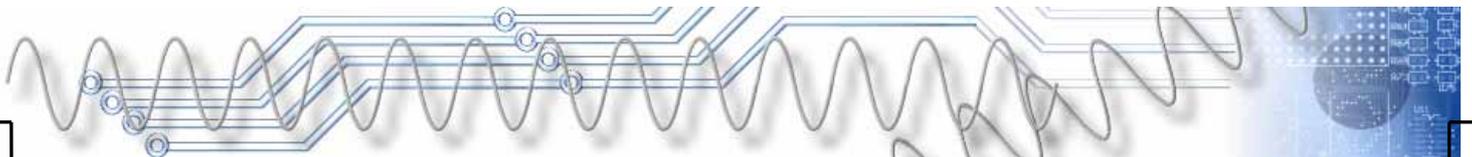
9-2-2 達靈頓電晶體驅動電路

如果不放心那顆小小的 IC，或要驅動較大的步進馬達，則可利用中功率包裝(TO-220)的達靈頓電晶體，如 TIP122 等，這種電晶體可瞬間放大而達到 1 到 3 安培，足以應付大多數的步進馬達。當然，在每相的驅動電路裡，還是需要一個基極電阻，以抑制過大的基極電流，而在輸出端，也連接一個 1N4001，提供電感器(步進馬達的線圈)的放電路徑，如下圖所示：



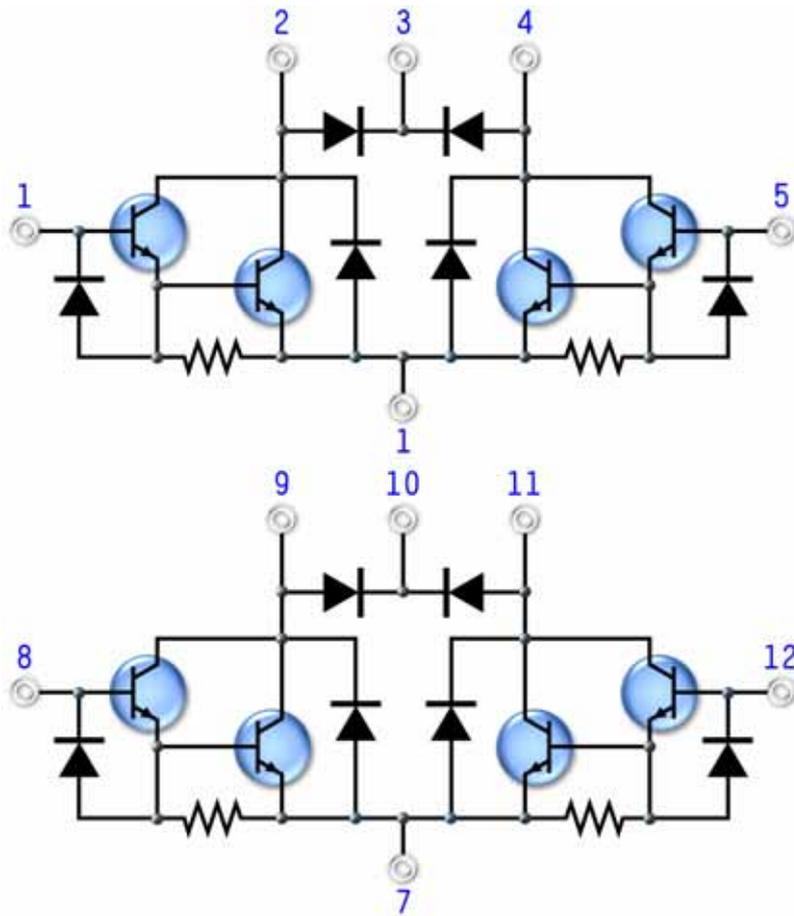
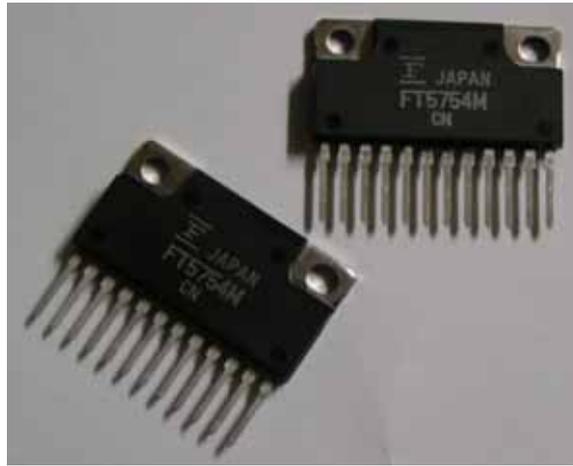
(圖19) 達靈頓電晶體驅動電路

若驅動電流仍然不夠，則可利用 2N2053 與 2N3055 搭接成達靈頓電路，以取代上圖中的 TIP122 達靈頓電晶體，即可提供更多的驅動電流。

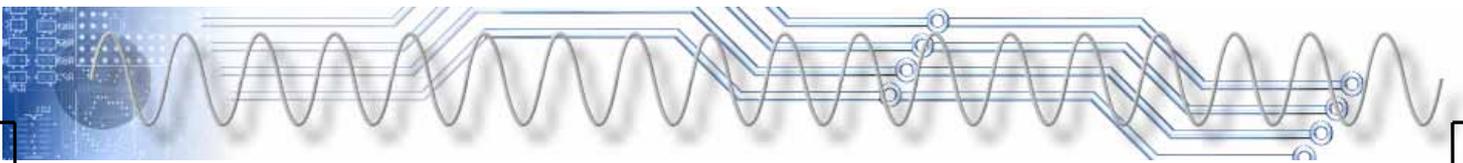


9-2-3 FT5754 驅動電路

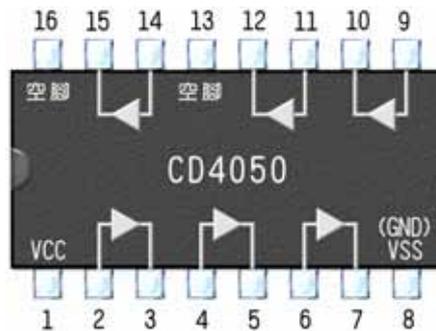
FT5754 是一顆專用的步進馬達驅動 IC，這是一顆 12 隻接腳的功率 IC，如下圖所示為其外觀與其內部電路：



(圖20) FT5754 電路

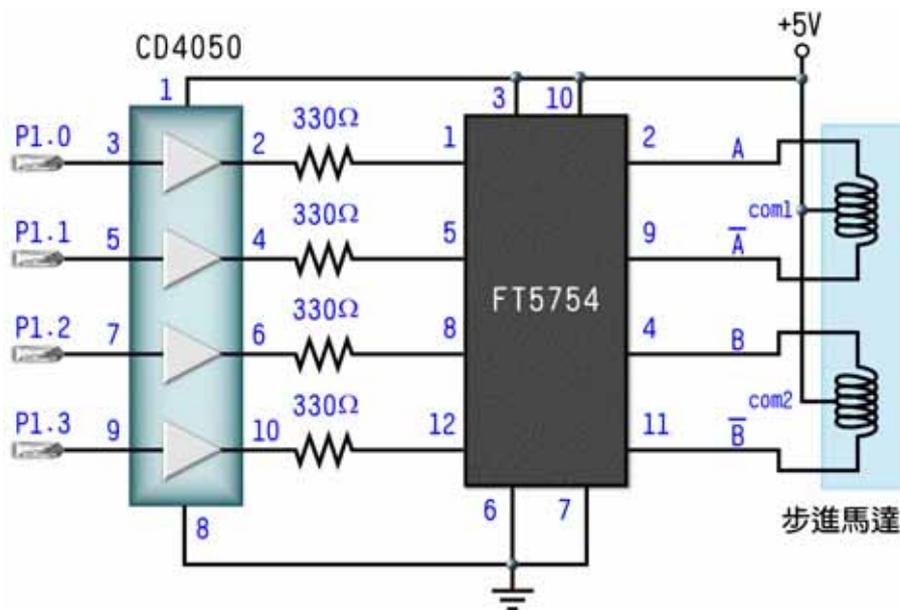


如上圖所示，其中包括四個相同的達靈頓模組，電路結構與 2003 系列 IC 的內部電路有點相似，不過，FT5754 所能提供的電路更大！所以，我們可以將它拿來驅動較大的步進馬達。通常在應用 FT5754 時，都會使用一個緩衝器，與微處理器連接，例如 CD4050 就是一個常用的 CMOS 緩衝器，其接腳圖如下所示：

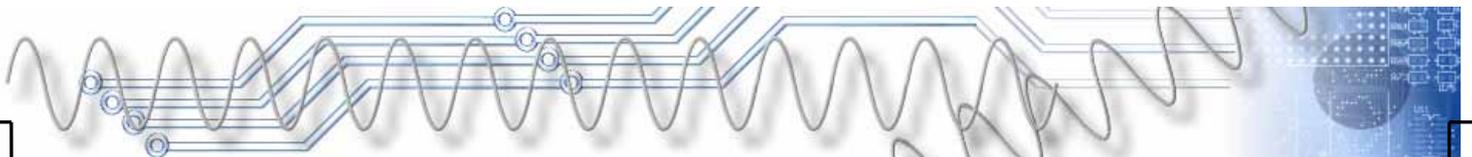


(圖21) CD4050 接腳圖

如下圖所示，則為 FT5754 步進馬達驅動電路



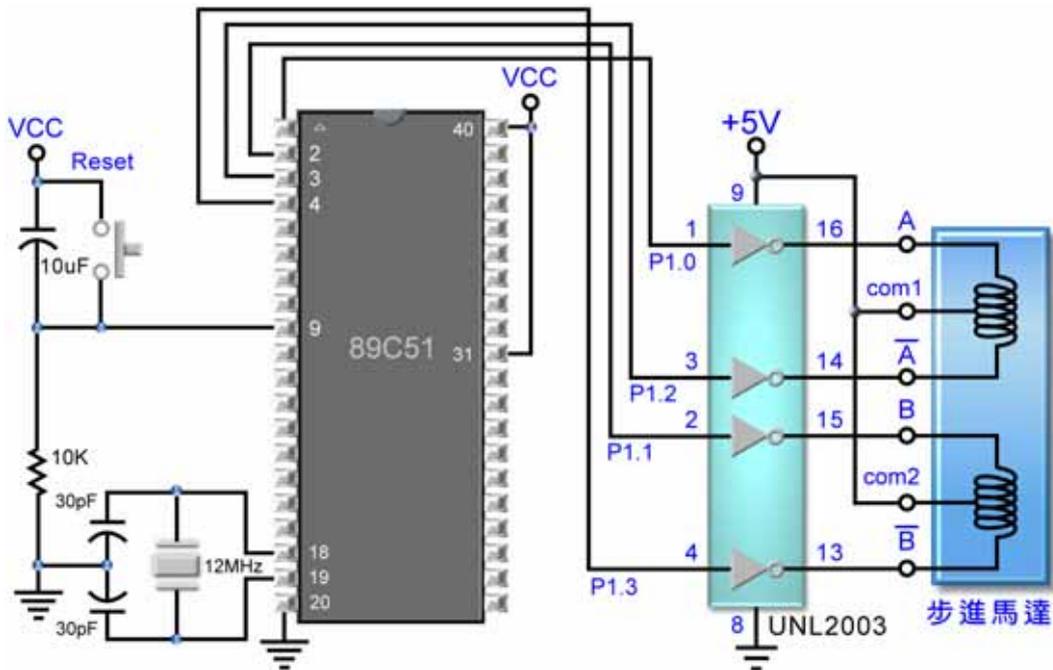
(圖22) FT5754 步進馬達驅動電路



9-3 實例演練

在本單元裡提供四個範例，以展示步進馬達的控制方法，如下所示：

9-3-1 以延遲副程式產生驅動信號實例演練



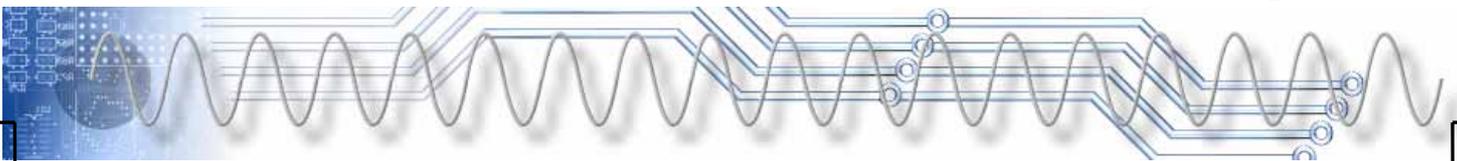
(圖23) 電路圖

● 功能說明

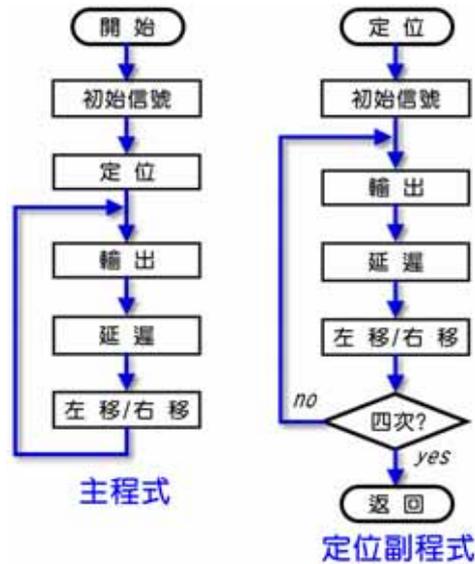
如圖 23 所示，在此使用小型的步進馬達，所以直接採用 UNL2003 驅動電路，若使用較大的步進馬達，則改用達靈頓電晶體而驅動電路，或 ET5754 驅動電路等。本單元將利用延遲副程式來控制旋轉的速度，而激磁信號為 1 相激磁或 2 相激磁。當程式開始時先定位，然後不斷地正轉或反轉。

● 參考程式

依功能需求與電路結構得知，其控制程式與單燈左移(1 相激磁)，或雙燈左移(2 相激磁)類似，而其中的延遲副程式將產生 0.05 秒倍數的延遲，其倍數則在程式開始的「TIMES EQU 10」虛擬指令中指定。同樣地，激磁方式也可在程式開始的「PHASE EQU 11H」虛擬指令中指定，若要改採 2 相激磁，則改為「PHASE EQU 22H」



即可。另外，在此由 P1.0 到 P1.3 連接到步進馬達驅動電路，若要改由其它輸出埠輸出，則可在「OUT REG P1」中定義。



```

;利用 DELAY 副程式(0.05 秒 XR5)，產生驅動信號
;由 P1 輸出
;#11H 為 1 相驅動，#33H 為 2 相驅動
;速度為 1/(0.05XTIMES) 步/秒
TIMES      EQU    10          ;重複次數
PHASE      EQU    11H        ;驅動方式
OUT        REG    P1         ;輸出埠
;=====
                ORG    0          ;
                CALL  POSITION      ;呼叫定位副程式
RL_1:        MOV    A, #PHASE     ;指定驅動信號
                MOV    OUT, A     ;輸出驅動信號
                MOV    R5, #TIMES ;指定重複數
                CALL  DELAY       ;呼叫延遲副程式
                RL    A           ;下一個驅動信號
                JMP   RL_1        ;跳至 RL_1 形成迴圈
;=====延遲副程式==R5x0.05 秒=====
DELAY:      MOV    R7, #100
D1:         MOV    R6, #250
                DJNZ  R6, $
                DJNZ  R7, D1
                DJNZ  R5, DELAY
                RET
;=====定位副程式=====
POSITION:   MOV    30H, #4        ;四個驅動信號
                MOV    A, #PHASE ;指定驅動信號
  
```

```

P_1:    MOV    OUT, A        ;輸出驅動信號
        MOV    R5, #TIMES   ;指定重複數
        CALL  DELAY        ;呼叫延遲副程式
        RL    A            ;下一個驅動信號
        DJNZ  30H, P_1      ;跳至 P_1 形成迴圈
        RET                ;返回
;=====
        END

```

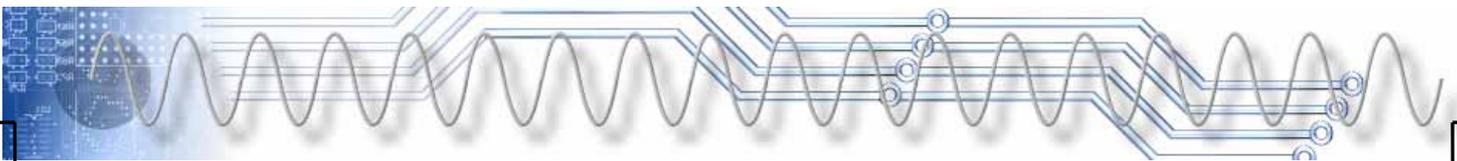
1 相驅動實驗 (ch9-1.asm)

● 操作

1. 依功能需求與電路結構撰寫程式，然後將該程式組譯與連結，以產生*.HEX 檔。
2. 請利用 AVSIM51 之類的模擬軟體，模擬其功能。若有非預期的狀況，則檢視原始程式，看看哪裡出問題？並將它記錄在實驗報告裡。
3. 請按圖 23 連接線路，再使用實體模擬器，載入該程式(*.HEX)，以模擬該電路的動作。若有非預期的狀況，則檢視線路的連接狀況，看看哪裡出問題？並將它記錄在實驗報告裡。
4. 若實體模擬功能正常，請將程式燒錄到 89C51，再把該 89C51 放入實體電路，以取代剛才的實體模擬器，然後直接送電，看看是否正常？
5. 撰寫實驗報告。

● 思考一下

1. 當 TIMES 參數為 10 時，每步之間的時間差距約 0.5 秒，則步進馬達轉一圈(200 步)應該是 100 秒，請實際量測，是否相符？
2. 試著增加 TIMES 參數，看看是否能使步進馬達的速度變慢？再試著減少 TIMES 參數，看看是否能使步進馬達的速度變快？而找出多快時，步進馬達就不能正常轉動？
3. 請修改程式，採 2 相激磁方式驅動步進馬達？
4. 若要從 P2 輸出到步進馬達，要如何修改程式？



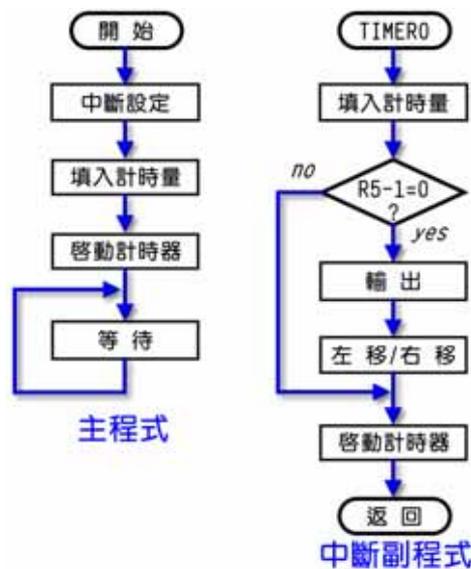
9-3-2 以計時器產生驅動信號實例演練

● 功能說明

同 9-3-1 節的電路(圖 23)，改以計時器的方式，替代延遲副程式。

● 參考程式

若要以計時器的方式產生時間延遲，則在主程式裡設定所有計時器中斷的設定，並啟動計時器後，就讓程式在原地打轉，主要的動作是在中斷副程式裡操作。對於 MODE 1 而言，每次中斷最多只能延遲 0.065 秒，太短了！所以，在此只設定為-50000，以產生 0.05 秒的中斷，再利用 R5 做為計次裝置，每中斷 R5 次，才改變輸出的信號。如此就能以改變 R5 的次數，以控制步進馬達的速度。



```

;利用 TIMER 0, MODE 1 計時(0.05 秒 XR5), 以產生驅動信號
;由 P1 輸出
;#11H 為 1 相驅動, #33H 為 2 相驅動
;速度為 1/(0.05XTIMES) 步/秒
TIMES      EQU    10          ;重複次數
PHASE      EQU    33H        ;驅動方式
COUNT     EQU    -50000     ;計時量
OUT        REG    P1         ;輸出埠
;=====
                ORG    0          ;開始
                JMP    START      ;跳至 START
                ORG    0BH        ;TIMER0 中斷向量
                JMP    TIMER0     ;跳至 TIMER0 中斷副程式
  
```

```

START:    MOV    IE, #10000010B ;中斷致能
          MOV    TMOD, #1      ;設定模式
          MOV    TH0, #>COUNT ;填入計時量
          MOV    TL0, #<COUNT ;填入計時量
          MOV    A, #PHASE     ;設定驅動方式
          SETB   TR0           ;啓動 TIMER0
          MOV    R5, #TIMES    ;指定重複數
          JMP    $             ;停滯

;=====
TIMER0:   CLR    TR0           ;關閉 TIMER0
          MOV    TH0, #>COUNT ;填入計時量
          MOV    TL0, #<COUNT ;填入計時量
          DJNZ   R5, AGAIN     ;時間到了嗎?
          MOV    R5, #TIMES    ;重新指定重複次數
          MOV    OUT, A        ;驅動喇叭
          RL     A             ;左移
AGAIN:    SETB   TR0           ;啓動 TIMER0
          RETI   ;返回

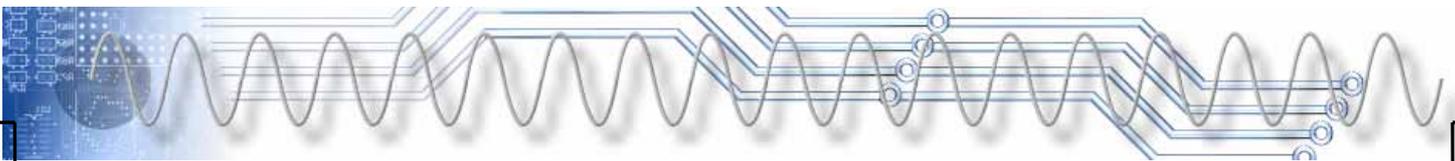
;=====
          END

```

2 相驅動實驗 (ch9-2.asm)

操作

1. 依功能需求與電路結構撰寫程式，然後將該程式組譯與連結，以產生*.HEX 檔。
2. 請按圖 23 連接線路，再使用實體模擬器，載入該程式(*.HEX)，以模擬該電路的動作。若有非預期的狀況，則檢視線路的連接狀況，看看哪裡出問題？並將它記錄在實驗報告裡。
3. 若實體模擬功能正常，請將程式燒錄到 89C51，再把該 89C51 放入實體電路，以取代剛才的實體模擬器，然後直接送電，看看是否正常？
4. 撰寫實驗報告。



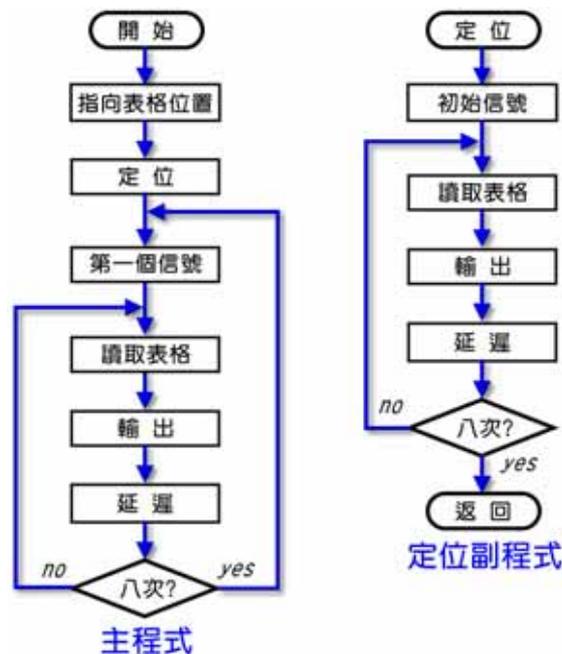
9-3-3 1-2 相驅動實例演練

● 功能說明

同 9-3-1 節的電路(圖 23)，但採 1-2 相激磁方式輸出。

● 參考程式

1-2 相激磁信號是夾雜著 1 相激磁信號與 2 相激磁信號，很難以前述之左移或右移方式產生。在此將以查表法，將這八個激磁信號放入表格之中，再逐一取出、輸出即可。



```

;由 P1 輸出
;利用查表法，以產生 1-2 相驅動信號
;速度為 1/(0.05xTIMES) 步/秒
TIMES      EQU    10          ;重複次數
OUT        REG    P1         ;輸出埠
;=====
START:     ORG     0
           MOV    DPTR, #TABLE ;指向表格位置
           CALL   POSITION      ;呼叫定位副程式
           MOV    R4, #8       ;八個驅動信號
           MOV    R3, #0       ;信號初始位置
LOOP:      MOV    A, R3        ;
           MOVC   A, @A+DPTR   ;取出驅動信號
           MOV    OUT, A       ;輸出
           MOV    R5, #TIMES   ;指定重複次數
  
```

```

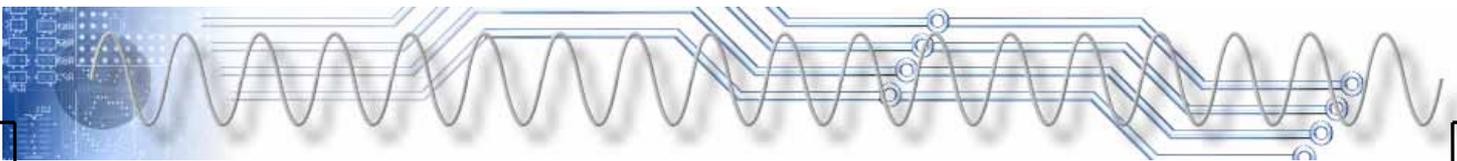
CALL    DELAY           ;呼叫延遲副程式
INC     R3              ;指向下個位置
DINZ   R4, LOOP        ;是否已輸出八個驅動信號
JMP    START           ;跳至 START 形成一個迴路
;=====延遲副程式==R5x0.05 秒=====
DELAY:   MOV    R7, #100
D1:     MOV    R6, #250
        DJNZ   R6, $
        DJNZ   R7, D1
        DJNZ   R5, DELAY
        RET
;=====定位副程式=====
POSITION: MOV    R4, #8           ;八個驅動信號
        MOV    R3, #0           ;信號初始位置
P_1:    MOV    A, R3             ;
        MOVC   A, @A+DPTR      ;取出驅動信號
        MOV    OUT, A          ;輸出
        MOV    R5, #TIMES      ;指定重複次數
        CALL   DELAY          ;呼叫延馳副程式
        INC    R3              ;指向下個位置
        DINZ   R4, P_1        ;是否已輸出八個驅動信號
        RET
;=====
TABLE:   DB    1, 3, 2, 6
        DB    4, 12, 8, 9
;=====
END

```

1-2 相驅動實驗 (ch9-3.asm)

● 操作

1. 依功能需求與電路結構撰寫程式，然後將該程式組譯與連結，以產生*.HEX 檔。
2. 請按圖 23 連接線路，再使用實體模擬器，載入該程式(*.HEX)，以模擬該電路的動作。若有非預期的狀況，則檢視線路的連接狀況，看看哪裡出問題？並將它記錄在實驗報告裡。
3. 若實體模擬功能正常，請將程式燒錄到 89C51，再把該 89C51 放入實體電路，以取代剛才的實體模擬器，然後直接送電，看看是否正常？
4. 撰寫實驗報告。



● 思考一下

1. 如何將本實驗裡的步進馬達，由正轉變成反轉？
2. 在本實驗裡，步進馬達轉一圈需要多少時間？

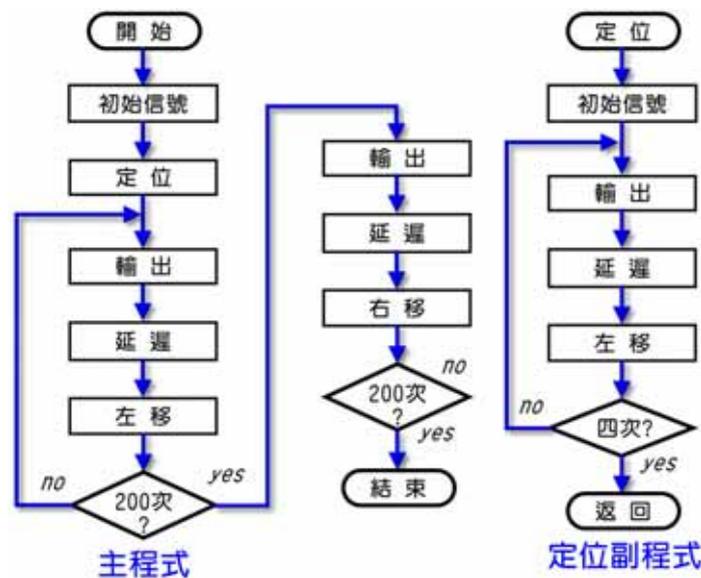
9-3-4 方向控制實例演練

● 功能說明

同 9-3-1 節的電路(圖 23)，而在此將先讓步進馬達正轉一圈(200步)，再反轉一圈。

● 參考程式

依功能需求與電路結構得知，最簡單的方式還是採用延遲副程式的方式，先以左移方式輸出 200 個信號，再以右移方式輸出 200 個信號。同樣地，激磁方式也可在程式開始的「PHASE EQU 11H」虛擬指令中指定，若要改採 2 相激磁，則改為「PHASE EQU 22H」即可。



```

;利用 DELAY 副程式(0.05 秒 XR5)，產生驅動信號
;由 P1 輸出，正轉 200 步(1 圈)，再反轉 200 步(1 圈)，
;#11H 為 1 相驅動，#33H 為 2 相驅動
;速度為 1/(0.05XTIMES) 步/秒
STEPS      EQU    200      ;步數設定
TIMES      EQU    10      ;時間延遲次數
PHASE      EQU    11H     ;激磁方式
  
```

```

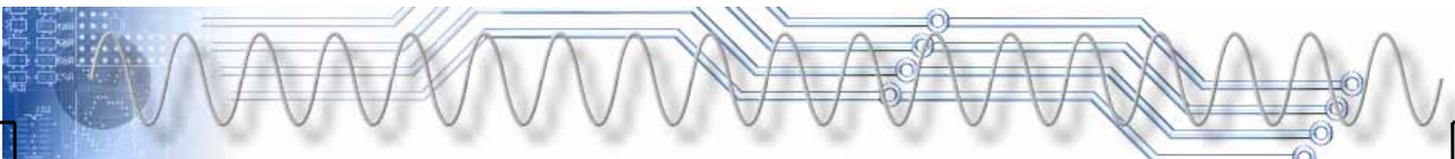
OUT          REG    P1          ;指定輸出埠
;=====
                ORG    0          ;程式開始位置
                CALL  POSITION     ;呼叫定位副程式
START:        MOV    A, #PHASE   ;指定驅動信號
;=====正轉=====
                MOV    R4, #STEPS ;指定正轉步數
RL_1:         MOV    OUT, A      ;輸出驅動信號
                MOV    R5, #TIMES ;指定重複數
                CALL  DELAY       ;呼叫延遲副程式
                RL     A          ;下一個驅動信號
                DJNZ  R4, RL_1    ;是否已 200 步?
;=====反轉=====
                MOV    R4, #STEPS ;指定正轉步數
RR_1:         MOV    OUT, A      ;輸出驅動信號
                MOV    R5, #TIMES ;指定重複數
                CALL  DELAY       ;呼叫延遲副程式
                RR     A          ;下一個驅動信號
                DJNZ  R4, RR_1    ;是否已 200 步?
;=====延遲副程式==R5x0.05 秒=====
DELAY:        MOV    R7, #100
D1:           MOV    R6, #250
                DJNZ  R6, $
                DJNZ  R7, D1
                DJNZ  R5, DELAY
                RET
;=====定位副程式=====
POSITION:
                MOV    30H, #4    ;四個驅動信號
                MOV    A, #PHASE  ;指定驅動信號
P_1:          MOV    OUT, A      ;輸出驅動信號
                MOV    R5, #TIMES ;指定重複數
                CALL  DELAY       ;呼叫延遲副程式
                RL     A          ;下一個驅動信號
                DJNZ  30H, P_1    ;跳至 R_1 形成迴圈
                RET
;=====
                END

```

方向控制實驗 (ch9-4.asm)

● 操作

1. 依功能需求與電路結構撰寫程式，然後將該程式組譯與連結，以產生*.HEX 檔。
2. 請按圖 23 連接線路，再使用實體模擬器，載入該程式(*.HEX)，



以模擬該電路的動作。若有非預期的狀況，則檢視線路的連接狀況，看看哪裡出問題？並將它記錄在實驗報告裡。

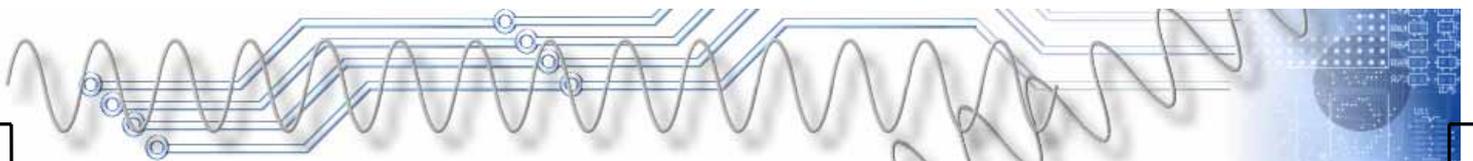
3. 若實體模擬功能正常，請將程式燒錄到 89C51，再把該 89C51 放入實體電路，以取代剛才的實體模擬器，然後直接送電，看看是否正常？
4. 撰寫實驗報告。

9-4

即時練習

在本章裡介紹步進馬達的原理、步進馬達的驅動電路，並探討如何在 8051 裡產生驅動步進馬達的信號。在此請試著回答下列問題，以確認對於此部分的認識程度。

1. 試說 2 相 5 線式步進馬達與 2 相 6 線式步進馬達之異同？
2. 某步進馬達之轉子齒間距為 14.4 度，則其步進角度為多少？
3. 某 200 步之步進馬達，若以 1 相驅動，則每個驅動信號，將產生多少角度的位移？若改以 1-2 相驅動，則每個驅動信號，將產生多少角度的位移？
4. 同一個步進馬達，使用 1 相驅動與 2 相驅動，有何差別？
5. 若要進行精確的位置或角度控制，在使用步進馬達之前，必須進行定位或歸零，這個動作是如何進行的？
6. 試問 UNL2003 系列驅動 IC，每顆 IC 提供多少個反相驅動器？而每個反相驅動器最大能吸取多大電流？
7. 試簡述 FT5754 步進馬達驅動 IC 的內部結構？
8. 請畫出以 FT5754 驅動步進馬達的電路？
9. 若要使用達靈頓電晶體來驅動步進馬達，其每一極電路為何？
10. 若程式所產生的驅動信號太快，步進馬達來不及反應，將會怎樣？



心得筆記

Handwriting practice area consisting of 18 horizontal lines, each starting with a blue dot on the left side.

