

國立臺北科技大學

學生實習心得報告

學期：110 學年度第 2 學期

科別：智慧自動化控制科

班級：三年級

學號：1082B0004、1082B0006

姓名：劉予安、林宏信

課程名稱：機電整合與實習

授課老師：李政宏博士

1 單元名稱：七段顯示器

本次的主題是七段顯示器，七段顯示器本身只是七個 LED 燈排成數字 8 的形狀，透過不同的燈光組合來呈現不同的數字。然而由於有 7 個 LED 燈就會佔用 7 個接腳，Arduino 上的接腳數量有限，就無法控制多個燈光。因此七段顯示器多半會搭配 74HC595 電晶體，它是 serial in - parallel out 的移位暫存器，即可只使用 clock、latch、serial in 三支腳位，輸出 8 位元的資料，連結到四個七段顯示器上，在使用 enable 控制哪幾個要亮起。

然而我們的開發板上只有一顆 74HC595，因此只能夠同時儲存 8 位元的資料，因此不可能同時讓兩個七段顯示器亮兩個不同的數字，所以實際上是使用輪閃的方式，運用視覺暫留來製造同時亮的效果。

本次的實作題目是設計透過按鈕切換的兩個模式，第一個是控制四個七段顯示器從左到右照 a、b、c...g 的順序逐一亮起，第二則是控制 4 個 LED 的霹靂燈。

2 實習設備與材料

項次	品名	數量
1	Arduino	1
2	七段顯示器	4
3	74HC595	1

3 實習步驟

一開始是先實作控制四個七段顯示器的函式，我透過一個四個元素的陣列，儲存四個七段顯示器分別要亮起的二進制編碼 (8 bits)。show 函式會將四個七段顯示器輪流寫入資料、點亮 1ms、熄滅、換下一個七段顯示器。74HC595 透過 shiftOut 函式來寫入串列資料，在打開 latch 後，將 digits[i] 裡面的二進制編碼，先做 binary NOT 後，每個 clock 會推入一位元到 DATA pin 中，一共推入八個位元。此函式即可讓四個七段顯示器顯示任意的四個數字，但必須持續呼叫才會亮，因此主函式必須採用非阻塞寫法，所有的函式也不能有 delay。

```
1 void show(byte digits[4]) {
2     for (int i = 0; i < 4; i++) {
3         digitalWrite(LATCH, LOW);
4         shiftOut(DATA, CLK, MSBFIRST, ~digits[i]); // 低態觸發
5         digitalWrite(LATCH, HIGH);
6         digitalWrite(scan[3 - i], LOW); // 點亮一毫秒
7         delay(1);
8         digitalWrite(scan[3 - i], HIGH);
9     }
10 }
```

再來是模式一的函式，此程式會每 200ms 將現在指定的數字往左移位並補一，即可做出燈光逐一亮起的效果。當一個七段顯示器的所有 LED 皆亮起時，則將 p 移至下一個數字。當四個數字都已被點亮，則重置所有 LED。

```
1 void mode_display() {
2     static byte digits[4]; // 所有數字的狀態
3     static byte p = 0; // 現在要改變的數字
4     static unsigned long checkpoint = 0;
5     show(digits);
6     if (millis() - checkpoint > 200) {
7         if (p == 4) {
8             // 所有 LED 皆點亮，重設所有值
9             p = 0;
10            for (int i = 0; i < 4; i++) digits[i] = 0;
11        }
12        digits[p] <= 1; // 向左移位
13        digits[p] += 1; // 補一
14        if (digits[p] == 0xFF) {
15            // if current digit is full, change to the next one
16            p++;
17        }
18        checkpoint = millis();
19    }
20 }
```

至於模式二則是一個相當簡單的輪閃程式，我用硬體接線來實作老師要求的特殊順序：1、3、2、4。

```
1 void mode_led() {
2     static unsigned long checkpoint;
3     static int c = 0;
4     if (millis() - checkpoint > 200) {
5         for (int i = 0; i < 4; i++) {
6             digitalWrite(LED[i], i != c%4);
7         }
8         c++;
9         checkpoint = millis();
10    }
11 }
```

其餘模式切換與主函式都與上一次報告類似，便不加贅述。

4 問題與討論

這次由於 74HC595 只有一顆的緣故，是使用輪閃的方式來實作，然而過程中可以觀察到，眼睛雖然看不出有閃，但是亮度跟持續亮著有差別，若今天要控制的是十六個七段顯示器，輪閃造成的亮度損失可能會相當嚴重。因此我想更好的做法，應為每個七段顯示器搭配一個 74HC585，再透過多工器來將 clock、latch、serial in 接到每個 74HC595 上，這樣即可做到輪流更新、同時亮起，在沒有更新數值的時候也不需時時更新 74HC595，更重要的，不會有亮度減低的不良影響。

5 心得與建議

這次學習到了如何操作七段顯示器及 74HC595，一開始沒有搞清楚串列輸入、輪閃等等的性質，遇到了一些奇怪的錯誤，在一次次嘗試中才慢慢理解實際運作原理。在幫助同學的過程中發現，大部分人都是卡在「輪閃」這個點上，我自己猜想是因為不了解 shiftOut 這個陌生指令究竟做了什麼，加上有多層的迴圈，邏輯確實是複雜了些。也許需要讓大家更了解程式各段的架構、邏輯究竟代表什麼，當然也需要搭配每個人的實作，從錯誤中去理解運作原理，但不能只是「程式能動就好」，而必須能解釋程式為什麼能動，才是真正的學會。

6 完整程式碼

```
1 const unsigned int scan[] = {5, 6, 7, 8};
2 const unsigned int LED[] = {9, 13, 2, 3};
3 const unsigned int BUTTON[] = {A0, A1};
4 #define DATA 10
5 #define LATCH 11
6 #define CLK 12
7
8 void setup() {
9     for (int i = 0; i < 4; i++) {
10         pinMode(scan[i], OUTPUT);
11         digitalWrite(scan[i], HIGH);
12     }
13     for (int i = 0; i < 4; i++) {
14         pinMode(LED[i], OUTPUT);
15         digitalWrite(LED[i], HIGH);
16     }
17     for (int i = 0; i < 2; i++) {
18         pinMode(BUTTON[i], INPUT);
19     }
20     pinMode(DATA, OUTPUT);
21     pinMode(LATCH, OUTPUT);
22     pinMode(CLK, OUTPUT);
23 }
24
25 int get_mode() {
26     static int last = 0;
27     for (int i = 0; i < 2; i++) {
28         if (!digitalRead(BUTTON[i])) {
29             last = i+1;
30             return last;
31         }
32     }
33     return last;
34 }
35
36 void show(byte digits[4]) {
37     for (int i = 0; i < 4; i++) {
38         digitalWrite(LATCH, LOW);
39         shiftOut(DATA, CLK, MSBFIRST, ~digits[i]); // 低態觸發
40         digitalWrite(LATCH, HIGH);
41         digitalWrite(scan[3 - i], LOW); // 點亮一毫秒
42         delay(1);
43         digitalWrite(scan[3 - i], HIGH);
44     }
45 }
46
47 void mode_display() {
48     static byte digits[4]; // 所有數字的狀態
49     static byte p = 0; // 現在要改變的數字
50     static unsigned long checkpoint = 0;
51     show(digits);
52     if (millis() - checkpoint > 200) {
53         if (p == 4) {
54             // 所有 LED 皆點亮，重設所有值
55             p = 0;
56             for (int i = 0; i < 4; i++) digits[i] = 0;
57         }
58         digits[p] <= 1; // 向左移位
59     }
60 }
```

```
59     digits[p] += 1;    // 補一
60     if (digits[p] == 0xFF) {
61         // if current digit is full, change to the next one
62         p++;
63     }
64     checkpoint = millis();
65 }
66 }
67
68 void mode_led() {
69     static unsigned long checkpoint;
70     static int c = 0;
71     if (millis() - checkpoint > 200) {
72         for (int i = 0; i < 4; i++) {
73             digitalWrite(LED[i], i != c%4);
74         }
75         c++;
76         checkpoint = millis();
77     }
78 }
79
80 void loop() {
81     switch (get_mode()) {
82     case 1:
83         mode_display();
84         break;
85
86     case 2:
87         mode_led();
88         break;
89     }
90 }
```
