

投稿類別：工程技術類

篇名：
創意點矩陣

作者：
顏士傑。台北市立松山高級工農職業學校。電子科三年級智班
蔡忠義。台北市立松山高級工農職業學校。電子科三年級智班

指導老師：
郭盈顯老師

壹●前言

燈在日常生活中與我們息息相關，但是燈總是只有亮滅兩種變化，每天看著這些死板板的燈，原本有朝氣的人都會變得死氣沉沉的，於是我們便想到，如果我們讓這些燈活起來，是否會讓燈多點朝氣，讓這些燈成為獨一無二有價值的裝飾，於是我們想到用我們二年級下學期所學到的點矩陣來做這個實驗，因為總比拿一顆一顆 LED 燈接電路要方便得多，既省空間又一體成型，使電路不會散得亂七八糟的，我們將接好的點矩陣電路搭配上 89S52 和自己設計的程式，讓他能夠以不同的方式運作，不再像以前一樣只有照明一種模式，而是多一項點綴環境的功能。

貳●正文

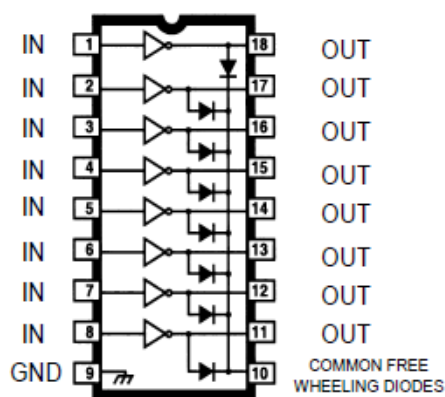
一、相關材料介紹

(一) ULN2803A

1.簡介

2803 的內部共有提供八組達靈頓電路，可以控制 0.5A，50V 以下的負載電路，輸入端控制準位使用 TTL5V 即可運作，可以運用於繼電器、紅外線 LED 發射器、步進馬達等等。

2.腳位圖，如圖一



圖一 2803 腳位圖

3.腳位介紹

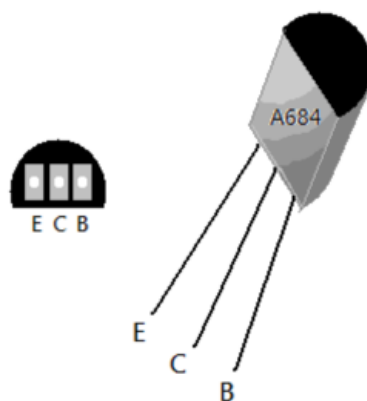
第一到第八腳是八組的輸入端，第十一到第十八腳則是八組的輸出端，第九腳是接地腳，而第十腳則是提供二極體迴流路徑，保護內部達靈電路，不受電感性負載反向電動勢擊穿，當負載接+24V 時，第十腳就接到+24V，若是接+10V，那就將第十腳接到+10V，這隻接腳並不是真正的電源輸入接腳，此顆 IC 不需額外電源提供。

(二) A684 電晶體

1.簡介

A684 是一顆 PNP 電晶體，電晶體的工作模式分成四種順向工作區、反向工作區、飽和區和截止區，它的 I_c 電流可以達 1A，我們將它用於飽和區及截止區，也就是開關的作用。

2.A684 腳位圖，如圖二



圖二 A684 腳位圖及外觀

二、實際製作

(一) 功能說明

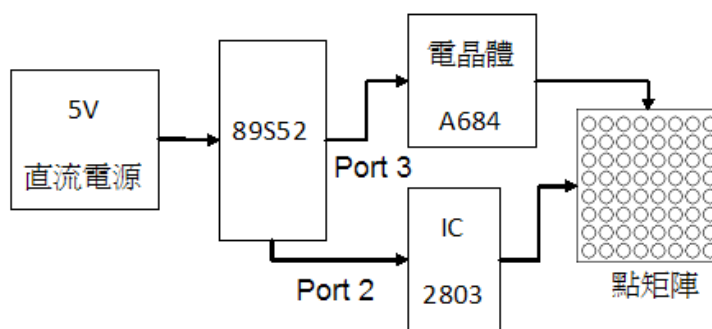
將點矩陣電路接妥後，利用 89S52 來使點矩陣能夠照我們所寫的程式跑動，跑動方式以隨機的方式控制橫向輸入端選擇不同的線路，再控制直向輸入端，使被選到線路的內部 LED 燈能逐一跑完。

(二) 材料表，如表一

表一 材料表

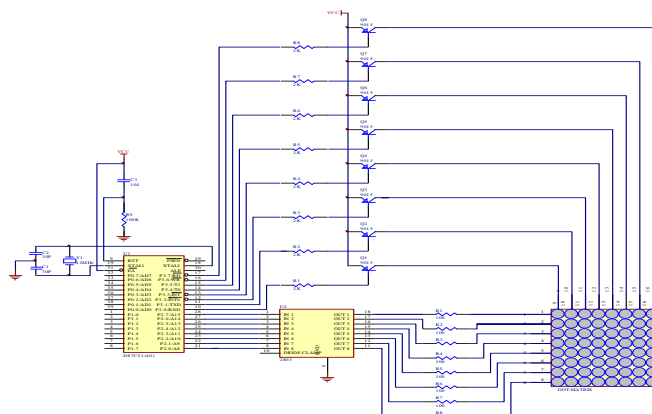
項次	名稱	規格	單位	數量
01	電阻	1.5KΩ	顆	8
02	電晶體	A684	顆	8
03	點矩陣	8X8	顆	1
04	積體電路	89S52	個	1
05	積體電路	2803	個	1
06	矽二極體	1N4001	顆	8
07	陶瓷電容	30 pF	顆	2
08	電解質電容	1μ F	顆	1
09	石英振盪器	12 MHz	個	1
10	麵包板		片	1
11	裸銅線			若干
12	單芯線			若干

(三) 電路方塊圖，如圖三



圖三 電路方塊圖

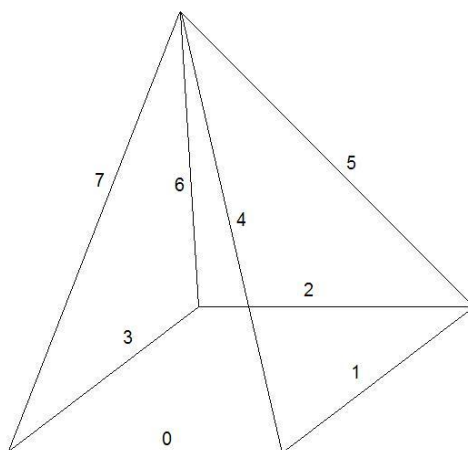
(四) 硬體電路圖，如圖四



圖四 電路圖

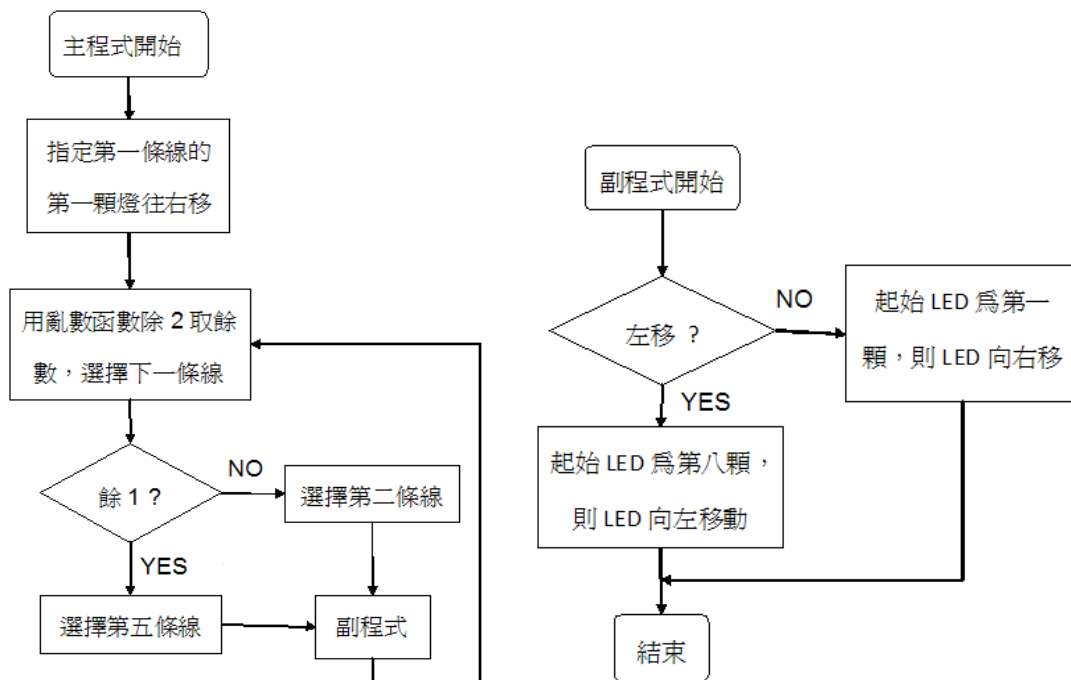
(五) 程式簡介

本次實驗程式，是以一四角錐之立體圖形為思考邏輯的方式所寫的，並將此程式以點矩陣呈現，四角錐如圖五。



圖五 四角錐編號圖

(六) 軟體流程圖



圖六 軟體流程圖

(七) 程式流程說明

1.宣告

宣告 $D=0,L=0,DATA=1,ESC=0,A,i,j$ 為短整數，宣告副程式 `void runled()`以及 `void delay(int)`延遲函數為整數，設 Port2 為點控制，Port3 為線控制。

2.主程式

設定初始值，一開始為 L0 開始跑，跑到最後一顆 LED 燈時，開始以亂數除以 2 取餘數，若餘 0 則往 L1 跑，若餘 1 則往 L4 跑，後面以此類推。

3 副程式.

若 LED 一開始亮在第一顆，則讓它跑向第八顆，若 LED 一開始亮第八顆，則讓它跑向第一顆，每顆燈約延遲 50ms。

4.延遲函數

將內迴圈 J 變數設為 60，約延遲 0.5ms，則將外迴圈 X 變數乘以 0.5ms 就是總共延遲的時間了。

(八) 部分重要程式

```
//===== 宣告變數與函數 =====  
#include <reg51.h>  
#include <stdlib.h>  
#define ROWPP2 // 輸出列接至 P2  
#define COLP P3 // 輸出行接至 P3  
unsigned short D=0,L=0,DATA=1,ESC=0,A,i,j; // 宣告函數  
void delay(int); // 宣告延遲函數  
void runled(); // 宣告 runled 函數  
//===== 主 程 式 =====  
main() // 主程式開始  
{  
    runled(); // 執行副程式  
    while (1) // 無窮盡迴圈  
    {
```

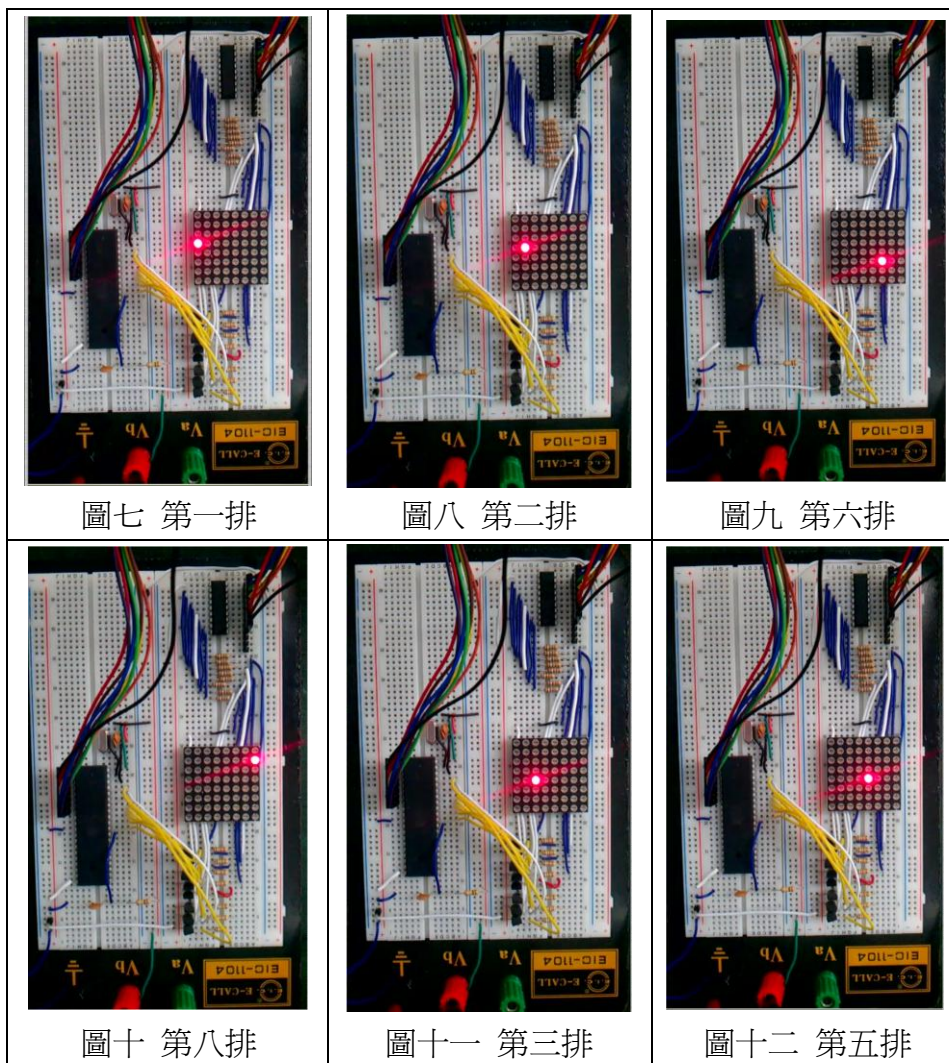
創意點矩陣

```
if (L==0&&D==0) // 如果 L 為第一條，方向向右移
{
    A=rand()%2; // 以亂數除 2 取餘數
    if (A==0) // 如果餘數為 0
    {
        L=1; // 選擇第二條
        D=0; // 方向向右移
        DATA=1; // 從第一顆 LED 開始跑
        ESC=0; // ESC 變數為 0
        runled(); // 執行副程式
    }
    else // 否則
    {
        L=4; // 選擇第五條
        D=0; // 方向向右移
        DATA=1; // 從第一顆 LED 開始跑
        ESC=0; // ESC 變數為 0
        runled(); // 執行副程式
    }
}
else if(L==1&&D==0) // 否則如果 L 為第二條，方向向
右移
{
    A=rand()%2; // 以亂數除 2 取餘數
    if (A==0) // 如果餘數為 0
    {
        L=2; // 選擇第三條
        D=0; // 方向向右移
        DATA=1; // 從第一顆 LED 開始跑
        ESC=0; // ESC 變數為 0
        runled(); // 執行副程式
    }
    else // 否則
    {
        L=5; // 選擇第六條
        D=0; // 方向向右移
        DATA=1; // 從第一顆 LED 開始跑
        ESC=0; // ESC 變數為 0
    }
}
```

創意點矩陣

```
runled(); // 執行副程式
}
}
//===== 副程式 =====
void runled()
{
while(ESC==0) // 無窮盡迴圈
{
P2=1<<L; // 設定 Port2 為控制線
P3=~DATA; // 設定 Port3 為控制點
delay(100); // 每顆燈延遲 0.05 秒
if(D==0) // 如果變數 D=0
{
DATA=DATA>>1; // LED 向右移
if (DATA==256) // 若 LED 跑超過第八顆
{ESC=1; // 變數 ESC=1，跳出副程式
DATA=0; // LED 關閉
}
}
else // 否則
{
DATA=DATA<<1; // LED 向左移
if (DATA==0) // 若 LED 跑超過第一顆
{ESC=1; // 變數 ESC=1，跳出副程式
DATA=256; // LED 關閉
}
}
COLP=L; // Port2 為控制線
ROWP=DATA; // Port3 為控制點
}
}
//===== 延遲函數 =====
void delay(int x)
{ int i,j; // 宣告變數
for (i=0;i<x;i++) // 外迴圈 xms
for (j=0;j<60;j++);// 內迴圈 0.5ms
} // 延遲函數結束
```


(九) 功能顯示圖



(十) 實做成果影片網址

http://www.youtube.com/watch?v=gIFYgvssP_s

參●結論

這次的實驗我們運用了 8X8LED 燈點矩陣的模式，搭配著 89S52 這顆 IC，程式則是用 C 語言所寫的，我們對於 C 語言都非常的陌生，在很多語法以及程式邏輯我們都不太了解，不了解的地方我們只能問老師以及查課本。

在這過程不斷的除錯，不斷的改程式，費盡千辛萬苦終於把程式給寫出來，經過偵錯後也沒有問題，於是我們把程式燒進了 89S52，再將 89S52 接到我們事先接好的點矩陣電路，燈它亮了，但看來看去好像跟我們所設計得不太一樣，沒錯，它的確跑錯了，我們觀察它亮的方式，點矩陣會跑，但是它是不亮的燈在跑，

而不是亮的燈在跑，並且每次跑到最後一顆都不會亮，我們檢查了電路又檢查了程式到底哪裡有錯，我檢查到了程式在設定 89S52 的 Port 腳時設錯了，並且還檢查到了一個很重要的地方。

由於 unsigned char 資料型態的範圍是 0~255，所以第八顆 LED 燈都不會亮，因為第八顆燈的範圍是 2^7 ，所以每當到了第八顆燈時超過了 255，因此燈就不會亮了，我們將 unsigned char 資料型態，改成 unsigned short 資料型態讓它的範圍變大到 0~65535，將這些問題都改過後，重新燒錄程式，問題就終於解決了，點矩陣它也依照我們所寫的方式去跑了。

對於這次的實驗我們覺得其實 C 語言很有趣，它的寫法跟 Visual Basic 不一樣，它的彈性和應用變化都比較大，所以我們如何寫出更多種的變化，如何讓我們的程式變得更簡短，這幾點的確值得我們深入研究，這次我們所寫的程式奇時是以一個四角錐的思維去寫得，所以硬體部分我們下次想把平面的點矩陣變成立體的形狀讓燈可以立體的跑，我想這絕對會比這次更有趣。

肆●引注資料

- 一、張義和(2009)。例說 89S51-C 語言(三版)。台北市：新文京開發。
- 二、孫宗瀛、黃金定、劉慶樹(1995)。常用數位 IC 資料手冊。金華科技圖書。
- 三、奇摩知識家。2011/10/22。取自
<http://tw.knowledge.yahoo.com/question/question?qid=1010081308019>
- 四、奇摩知識家。2011/10/22。取自
<http://sun.cis.scu.edu.tw/~lab/knowledge/t.htm>