

投稿類別：工程技術類

篇名：
計秒器及其原理探討

作者：
王得安。台北市立高級松山工農職業學校。電子科三年級智班
羅章晨。台北市立高級松山工農職業學校。電子科三年級智班

指導老師：
郭盈顯老師

壹●前言

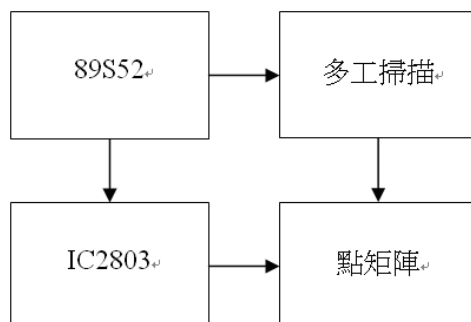
在馬路上總是能夠輕易的看到小綠人，在典禮上總是能夠輕易的看到跑馬燈，甚至連球場上的記分板、倒數計時的螢幕都是點矩陣做出來的，他們能夠依照程式跑出應該執行的文字或是圖案，非常容易引人注目，然而這些美麗的畫面，卻是由眾多個LED組成的，如此的圖案不禁令人感到好奇，究竟是如何將多數個LED變成會動的文字或者是美麗圖案。以下是本組的研究目的：

- 一、利用電晶體的掃描讓點矩陣顯示出圖案。
- 二、利用程式燒錄89S52製作一個計秒器。

貳●正文

本專題是由89S52當控制元件，配合ULN2803及PNP型的2SA684電晶體製成一個計秒器，顯示在點矩陣上。

一、電路方塊圖



圖一 電路方塊圖

電路方塊圖說明：

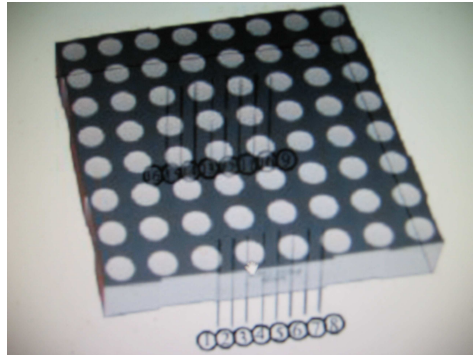
利用89S52讓2SA684對點矩陣的每一行進行掃描，再使IC2803讓點矩陣亮出字型。

二、重要零件介紹

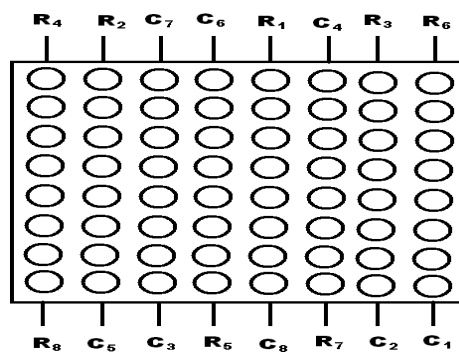
(一)點矩陣

1、外觀、構造及腳位

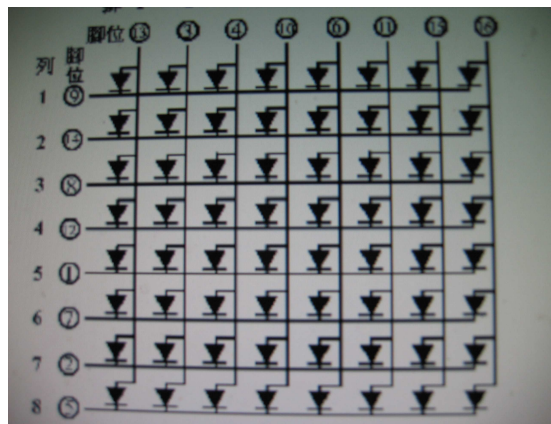
計秒器及其原理探討



圖二 點矩陣實體圖



圖三 點矩陣腳位



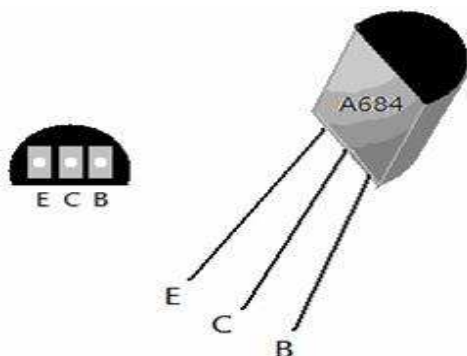
圖四 點矩陣內部構造

共陽8x8矩陣是控制輸入接腳為低電位發亮，共陽接腳接高電位，共陰8x8矩陣是控制輸入接腳為高電位發亮（註一）。共陽LED可用PNP電晶體驅動，共陰LED可用NPN電晶體驅動。

(二)2SA684

我們採用共陽的點矩陣，故推動晶體採用PNP型的2SA684電晶體當推動晶體。

1、符號



圖五 2SA684的頂視圖及外觀

2、功用

「電晶體的用途很多，除了做信號放大之外，也可以當作一個電子開關」（註三），大部分是利用電晶體的C腳與E腳之間的導通與否。

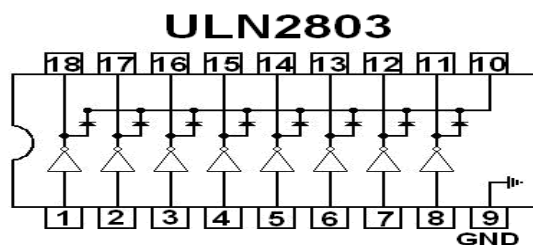
表一 電晶體的開關作用

	銻	矽
CE短路	$V_{BE} > 0.3V$	$V_{BE} > 0.7V$
CE開路	$V_{BE} < 0.3V$ 或 逆向偏壓時	$V_{BE} < 0.7V$ 或 逆向偏壓時

(三)ULN2803

本專題採用ULN2803推動點矩陣中LED的導通與否，此電路為高態輸出。

1、內部構造

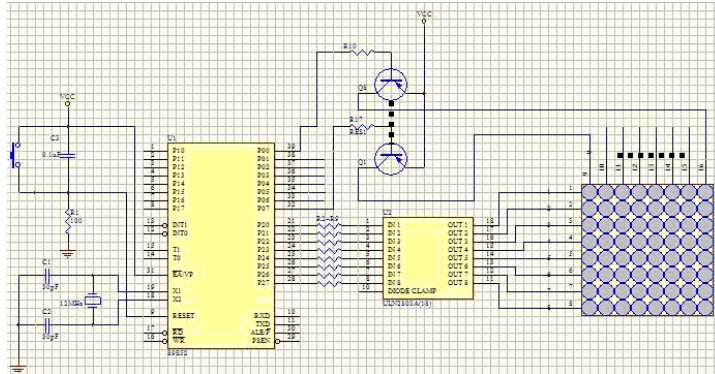


圖六 ULN2803內部結構

2、工作原理

ULN2803的原理是把輸入1利用反閘使它輸出變成0，或1變成0，而反閘是一只單輸入閘，其輸出恆為輸入的補數，即 $y = \bar{x}$ 。

三、電路圖

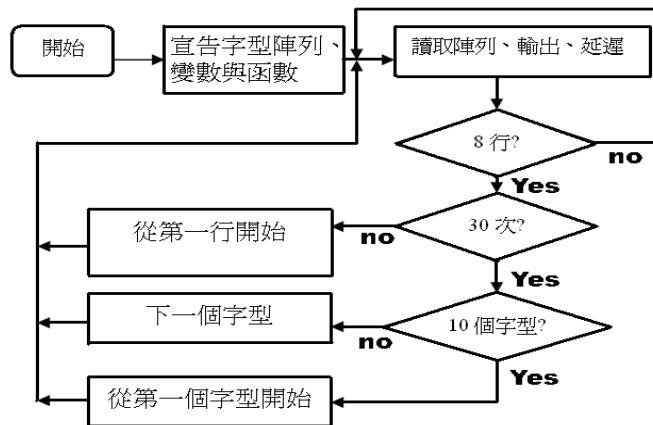


圖七 電路圖

電路圖說明：

89S52控制八個A684對點矩陣的行掃描，在此採用低態掃描，一行掃描完成後，再把低態信號轉到鄰近的其他行。再由ULN2803推動點矩陣中LED的輸出。

四、程式流程說明圖



圖八 程式流程說明圖

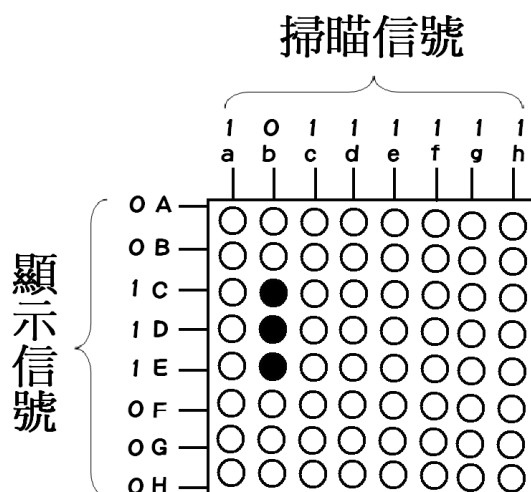
(一)流程說明

一開始程式宣告字型陣列、函數以及變數，開始掃描每一行，掃描全部八行之後，重複掃描30次，再輸出圖形。

(二)演算法

點矩陣以0為例子，多工掃描每2mS掃描一行。若掃描信號為「10111111」，則該行(第2行)裡的LED的陰極皆為高態；若顯示信號為「00111000」可能會亮，所以該行第3顆、第4顆、第5顆LED為低態。兩組信號交集，所以第2行裡的第3顆、第4顆、第5顆LED因順向導通而發亮，如圖九所示。

因為每掃描一行需要2mS，故掃完8行共16mS，剛好符合人類的視覺暫留(16mS)，這也就是我們為何能夠看見一個又一個字型出現的原因。



圖九 演算法範例

五、程式碼

```
#define ROWP    P2 // 輸出列接至P2
#define COLP    P0 // 掃描行接至P0
#define repeat 30 // 掃描30週，約2m*8*30=0.48秒
unsigned char code disp[][8]= // ===== 字 型 =====
{
    {0x00, 0x1c, 0x22, 0x41, 0x41, 0x22, 0x1c, 0x00}, // 0
    {0x00, 0x40, 0x44, 0x7e, 0x7f, 0x40, 0x40, 0x00}, // 1
    {0x00, 0x00, 0x66, 0x51, 0x49, 0x46, 0x00, 0x00}, // 2
    {0x00, 0x00, 0x22, 0x41, 0x49, 0x36, 0x00, 0x00}, // 3
    {0x00, 0x10, 0x1c, 0x13, 0x7c, 0x7c, 0x10, 0x00}, // 4
    {0x00, 0x00, 0x27, 0x45, 0x45, 0x45, 0x39, 0x00}, // 5

```

```

    {0x00, 0x00, 0x3e, 0x49, 0x49, 0x32, 0x00, 0x00}, // 6
    {0x00, 0x03, 0x01, 0x71, 0x79, 0x07, 0x03, 0x00}, // 7
    {0x00, 0x00, 0x36, 0x49, 0x49, 0x36, 0x00, 0x00}, // 8
    {0x00, 0x00, 0x26, 0x49, 0x49, 0x3e, 0x00, 0x00}}; // 9
void delaylms(int); // 宣告延遲函數
//===== 主 程 式 =====
main() // 主程式開始
{
    unsigned char i,j,k,scan; // 宣告變數
    while (1) // 無窮盡迴圈
    for (i=0;i<10;i++) // 字型0~9
        for (k=0;k<repeat;k++) // 重複執行repeat次
        {
            scan=1; // 初始掃描信號
            for (j=0;j<8;j++) // 掃描8行
            {
                ROWP=0x00; // 關閉LED
                COLP=~scan; // 輸出掃描信號
                ROWP=disp[i][j]; // 輸出顯示信號
                delaylms(2); // 延遲2ms
                scan<<=1; // 下一個掃描信號
            } // 掃描8行(j迴圈)結束
        } // 執行repeat次(k迴圈)結束
} // 主程式結束
//===== 延遲函數 =====
void delaylms(int x)
{
    int i,j; // 宣告變數
    for (i=0;i<x;i++) // 外迴圈 xms
        for (j=0;j<120;j++); // 內迴圈 1ms
} // 延遲函數結束

```

重要程式分析：

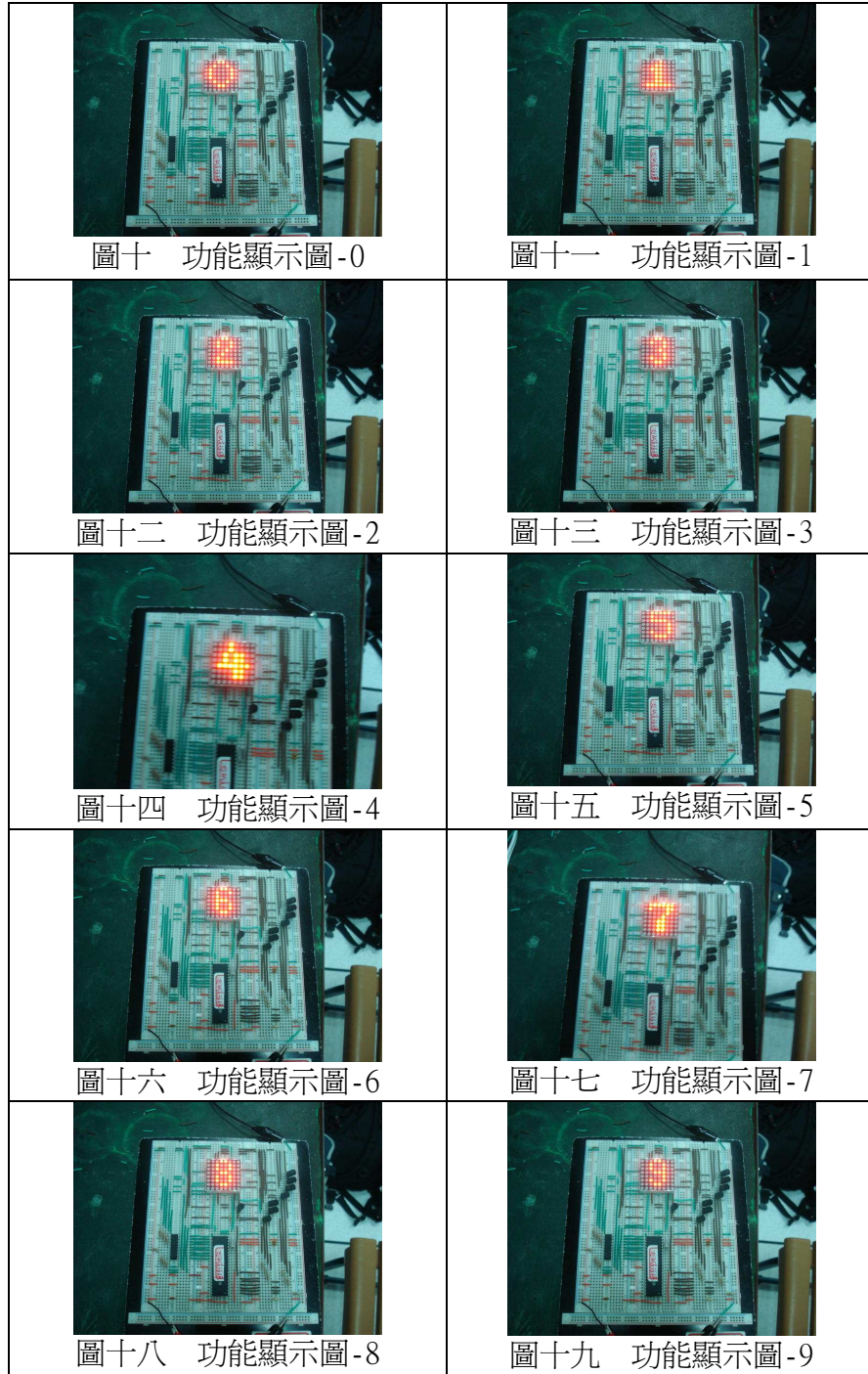
```

scan=1; // 初始掃描信號
ROWP=0x00; // 關閉LED
COLP=~scan; // 輸出掃描信號
ROWP=disp[i][j]; // 輸出顯示信號

```

一開始ROWP輸入為0x00經A684使點矩陣不亮，COLP開始有做掃描動作，所以此程式屬於低態掃描。ROWP作信號輸出。

六、功能顯示圖



參●結論

一開始思考元件以及線路的佈線該如何擺於麵包板就已經讓我們下不了手，花了一段時間才安排好位置，電路真的不是普通的複雜，但是最後還是成功啦！那種成就感是在實習課裡得不到的！因為第一次接好測試時是失敗的，

一想到如此複雜的電路，又要 Debug……，實在是心有餘而力不足啊！還好我們擁有一些基本的檢修能力：電晶體的 EBC 是否有接錯、89S52 的 PORT 有沒有與程式中不符或是 IC 是否接反了……，最後才發現原來因為電路過於複雜，有幾條單芯線短路了！檢修完畢，再次測試，才完成了這個實驗！

成功了一個 8X8 的點矩陣實驗，當然學習是沒有終點的，下次有機會能夠再自己製作一個電路的話，我們打算再做一個 16X16 的點矩陣呢！雖然電路會更複雜，但是我們已經早有準備了，不畏艱難才能夠得到更多的經驗，學習到更多的技術。

肆●引註資料

- 一、奇摩知識+。2011/10/20。
<http://tw.knowledge.yahoo.com/question/question?qid=1508010108014>
- 二、台灣晶技。2011/10/20。
http://www.txc.com.tw/tw/d_support/01.html#02
- 三、陳茂璋、鄧明發、郭盈顯(2000)。**基礎電子實習[II]**。臺北：知行出版社。