

類別：工程技術類

篇名：
亂數選號器

作者：
周書顯。臺北市立松山高級工農職業學校。電子科。三年級智班
楊竣棋。臺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

當看著老師拿出辛苦做的簽來給大家抽位子，同學也都一個個站起來去抽號碼牌，這種方法既辛苦又浪費時間，因此我就想做一個亂數選號器，可以節省時間也很公平，本電路功能如下：

- 一、製作一個可亂數程式。
- 二、旋鈕可以控制抽籤最大範圍。
- 三、按下小紅頭按鈕可執行亂數抽籤。
- 四、由 2 個 LED 7 段顯示器顯示抽籤結果。

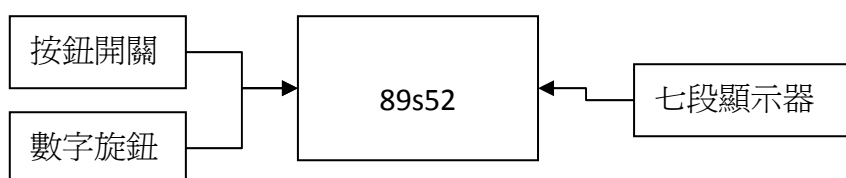
貳●正文

我們的專題爲了讓使用者可以控制選號範圍，因此加入了十進制對 BCD 碼的數字旋鈕開關，就可以讓使用者限定選號範圍，這樣就不會出現使用者不需要的號碼了，要不然使用者就會耗費掉很多不必要的時間。

一、電路圖

(一)硬體方塊圖

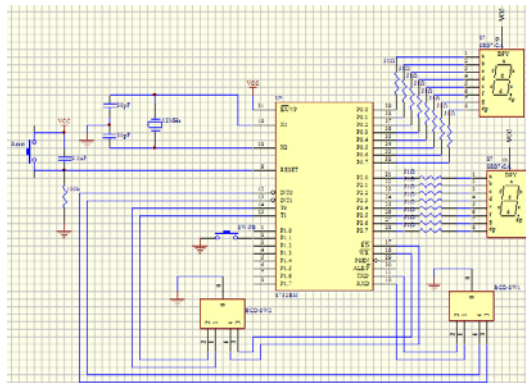
我們使用 89S52 來產生亂數選號，又用數字旋鈕來制定選號範圍，然後產生的數字輸出到七段顯示器，如圖一。



圖一 硬體方塊圖

亂數選號器

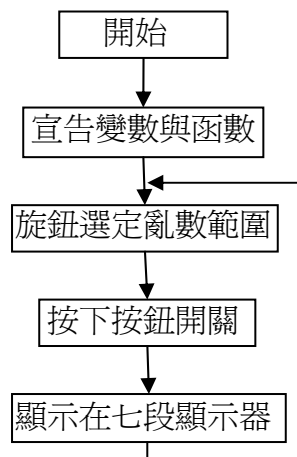
(二)電路圖



圖二 電路圖

二、軟體流程圖

設定完亂數選號範圍後，按下開關，系統會隨便挑出一組數字顯示在七段顯示器上，然後你可以在設定下一次選號範圍，再按下開關，就可以顯示出第二組數字，如圖三。



圖三 軟體流程圖

三、程式列表

```

//==宣告區=====
#include <reg51.h>//定義 8052 暫存器之標頭檔，P2-17~19
#define SW P3 //定義旋鈕接至 Port 3
sbit REDSW = P1^1; //宣告開關位置 Port 1 之 bit 1
#define SEG1 P0 //定義七節顯示器接至 Port 0
#define SEG2 P2 //定義七節顯示器接至 Port 2
  
```

亂數選號器

```
char code TAB[10]={0xc0,0xf9,0xa4,0xb0,0x99,          //數字 0-4
                  0x92,0x83,0xf8,0x80,0x98};         //數字 5-9
void debouncer(void);                                //宣告防彈跳函數
#include <stdlib.h>                                   //宣告亂數函數

//==主程式=====
main()        //主程式開始
{
  unsigned char A,F,I,H,L;                          //宣告變數
  REDSW=1;                                           //規劃輸入阜
  SEG1=TAB[H];                                       //輸出數字至七段顯示器
  SEG2=TAB[L];                                       //輸出數字至七段顯示器
  while(1)                                           //無窮迴圈，程式一直跑
  {
    srand(rand());                                  //亂數函數
    F=~P3>>4;                                       //將 Port 3 反向並向後移四位元
    A=F*10+(~P3 && 0x0F);                            //將 Port 3 改為二進制
    I = (rand()*100) % A + 1;                       //產生亂數
    L=I%10;                                          //定義個位數
    H=I/10;                                          //定義十位數

    if (REDSW==0)                                   //判斷 REDSW 是否按下
    { debouncer();                                   //呼叫防彈跳函數
      SEG1=TAB[H];                                   //輸出數字至七段顯示器
      SEG2=TAB[L];                                   //輸出數字至七段顯示器
      while(REDSW==0);                               //REDSW 是否按住？
      debouncer();                                   //呼叫防彈跳函數
    }                                                //if 敘述結束
  }                                                  //迴圈結束
}                                                    //主程式結束
//==副程式=====
void debouncer(void)                                //防彈跳函數開始
{ int i;                                           //宣告整數變數 i
  for(i=0;i<2400;i++);                             //計數 2400 次，延遲約 20ms
}                                                    //防彈跳函數結束
```

四、重要程式說明

(一) rand() 函數

在某些情況下(特別是設計遊戲時)，我們會需要使用到亂數，在 C 語言中提供了一個 rand() 函數(註一)可以讓我們取得一個亂數，其用法如下：

整數變數=rand();要使用 rand() 函數，必須 include 它的標頭檔 stdlib.h (stdlib 為 Standard Library, 標準函式庫)，上例的 rand() 將會傳回一個介於 0 到 RAND_MAX 之間的一個整數值，RAND_MAX 是 stdlib.h 定義的一個常數，其值在不同的系統有不同的大小。當然，這樣的變數對我們可能沒有什麼用，因此通常我們會搭配使用取餘數符號 % 來做運算。

(二) srand() 函數

當我們把上面的程式重複執行數次之後，會發現所得到的亂數是一樣的，這是因為它由上一個數值產生出下一個亂數，而一開始系統都是 0，所以每次產生出來的亂數才會一樣。因此我們可以用 srand() 函數改變一開始的亂數值(srand() 函數也是定義在 stdlib.h)，srand() 需要一個參數做為種子，以產生一個新的亂數數列。

(三) 限制亂數範圍程式

我們先使 F 轉為 4 個位元碼，然後再把 F 乘以十，做為十位數字，再加上十進制的個位數，之後再將丟到亂數公式內，此方法可以制定亂數公式的選擇範圍，最後再把得到的亂數 I 分別取十位碼 H 和個位碼 L，如圖四。

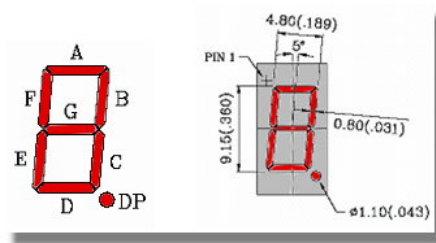
```
F=~P3>>4;
A=F*10+(~P3 && 0x0F);
I = (rand()*100) % A + 1 ;
L=I%10;
H=I/10;
```

圖四 限制亂數範圍

五、重要零件介紹

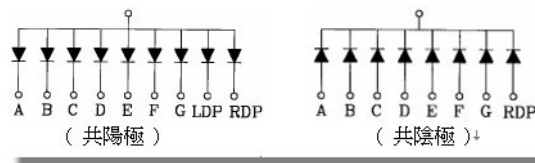
(一) 七段顯示器

七段顯示器(註二)，在許多產品或場合上經常可見。其內部結構是由八個發光二極體所組成，為七個筆畫與一個小數點，依順時針方向為 A、B、C、D、E、F、G 與 DP 等八組發光二極體之排列，可用以顯示 0~9 數字及英文數 A、b、C、d、E、F。目前常用的七段顯示器通常附有小數點，如此使其得以顯示阿拉伯數之小數點部份。七段顯示器的腳位和線路圖如下圖五所示(其第一支接腳位於俯視圖之左上角)。



圖五 七段顯示器尺寸圖

由於發光二極體只有在順向偏壓的時候才會發光。因此，七段顯示器依其結構不同的應用需求，區分為低電位動作與高電位動作的兩種型態的元件，另一種常見的說法則是共陽極(低電位動作)與共陰極(高電位動作)七段顯示器，如下圖六所示。



圖六 共陽極(低電位動作)與共陰極(高電位動作)

七段顯示器可用來顯示單一的十進制或十六進制的數字，它是由八個發光二極體所構成的(每一個二極體依位置不同而賦予不同的名稱，請參見圖五)。我們可以簡單的說，要產生數字，便是點亮特定資料的發光二極體。例如要產生數字「0」，須只點亮 A、B、C、D、E、F 等節段的發光二極體；要產生數字「5」，則須點亮 A、C、D、F、G 等節段發光二極體，以此

類推，參見圖七。因此，以共陽極七段顯示器而言，要產生數字「0」，必須控制 89S52 接腳接連至 A、B、C、D、E、F 等接腳呈現“低電位”，使電路形成通路狀態。表一則為共陽極七段顯示器顯示之數字編碼。

表一 共陽七段顯示器顯示之數字編碼

資料	DP	G	F	E	D	C	B	A	16 進制
0	1	1	0	0	0	0	0	0	C0
1	1	1	1	1	1	0	0	1	F9
2	1	0	1	0	0	1	0	0	A4
3	1	0	1	1	0	0	0	0	B0
4	1	0	0	1	1	0	0	1	99
5	1	0	0	1	0	0	1	0	92
6	1	0	0	0	0	0	1	0	82
7	1	1	1	1	1	0	0	0	F8
8	1	0	0	0	0	0	0	0	80



圖七 共陽七段顯示器顯示阿拉伯數字

(二) 十進制對 BCD 旋鈕開關

本專題使用此旋鈕，如圖八(A~F 不使用)，來控制亂數選擇範圍，此旋鈕為 BCD 碼旋鈕，所以我們使用兩個旋鈕，分別代表十位數及個位數，之後我們再把 BCD 碼轉為十進制碼後，才輸入亂數使用。

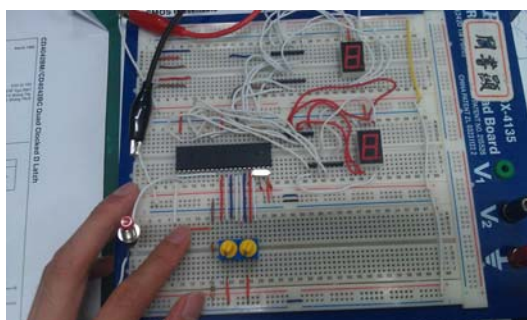


圖八 十進制對 BCD 碼旋鈕開關

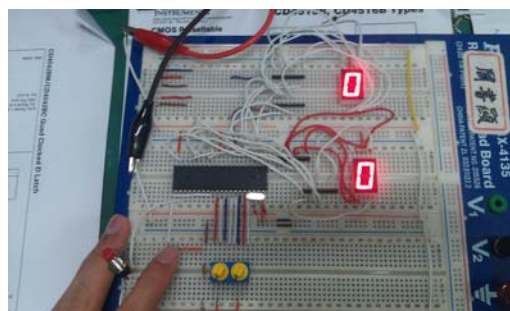
六、成品結果

先設定亂數選號的範圍，最大範圍可達1~99，如圖九。

接上電源，按下Rest鈕，使第九腳上的電壓降至低態，及恢復正常狀態（註三），使七段顯示器顯示初始值，如圖十。



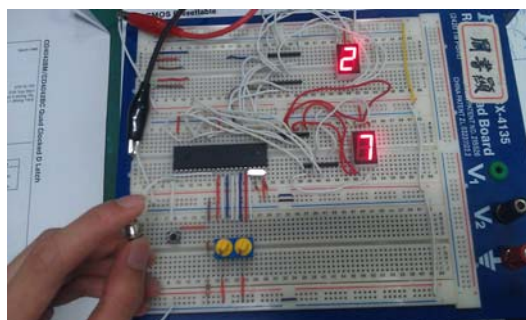
圖九 設定數字1~99



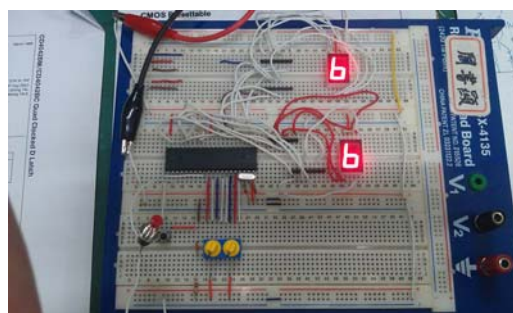
圖十 按下Rest鈕

按下開關，七段顯示器顯示亂數數值，上方為個位數，下方為十位數，如圖十一。

第二次按下開關，七段顯示器顯示出下一個亂數，上方為個位數，下方為十位數，如圖十二。

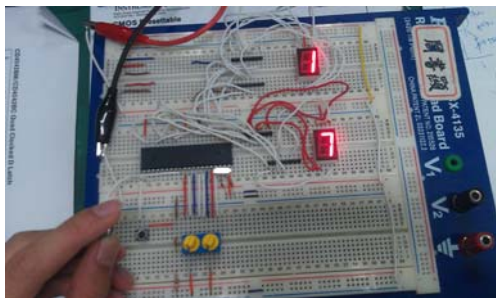


圖十一 按下開關圖



圖十二 第二次按下開關

第三次按下開關，七段顯示器顯示出第三個亂數，上方為個位數，下方為十位數，如圖十三。



圖十三 第三次按下開關

參●結論

一、問題解決

在打這個軟體程式時，所遇到最大的問題是亂數函數的部分，因為在我們學習單晶片時，是沒有學到亂數函數的部分，尤其我們還要可以控制選擇範圍，這又是一大難題，所以我們就想到使用十進制對BCD碼數字旋鈕來控制，那亂數就是使用rand()這個函數來產生。

二、未來展望

我們可以再發展出更大型、更多組的亂數選號，這樣就可以一次選出很多組號碼，就像樂透一樣，這樣就不用操作那麼多次，還有我們可以把亂數選號改為無線遙控，這樣就可以讓使用者無線操控抽籤機。

肆●引註資料

一、我心中的新西多摩。美麗 C 世界。rand() 函數。取自
<http://dhcp.tcgs.tc.edu.tw/c/p005.htm>

二、Altera FPGA/ASIC 多媒體雛型驗證平台。七段顯示器。取自
<http://www.starfpga.com/modules/tinyd3/>

三、張義和、王敏男、許宏昌、余春長(2009)。例說89S51-C語言(第三版)。
台北縣：新文京開發出版股份有限公司。