投稿類別：資訊類

篇名：使用 Visual C# 設計七段顯示器之數字辨認系統

作者：
呂育瑋。市立楊梅高中。資訊二甲

指導老師：簡樹桐 老師

壹、前言

一、研究動機

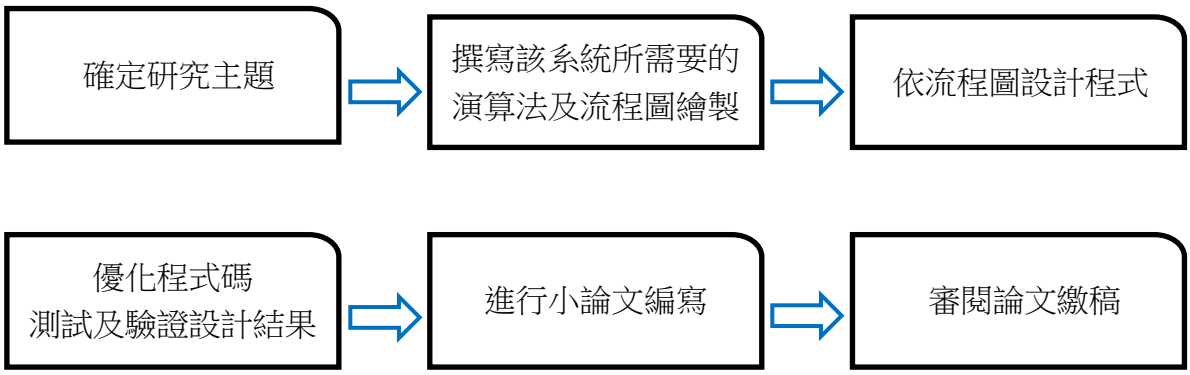
為了提升程式設計的能力，在試解工科技藝競賽「電腦軟體設計」歷屆試題過程中，其中有一題為「設計辨認數字系統」，原題意功能為利用亂數點亮七段顯示器的各段 LED，且判斷七段顯示器所點亮的各段組合為數字或非數字。考慮為了讓系統更加具有完整性，在老師的建議下，增加自動判斷的功能設計，而所謂數字與非數字的判斷依據，就依照常用 BCD 解碼演算法。

在高二上的「數位邏輯實習」課程中有做過七段顯示器的實習操作，配合這次解題功能要求需完成判斷是否為數字的程式，於是就開始針對這題目所要求的功能撰寫程式。其主要的想法為想透過程式設計的方式模擬實現部分硬體動作，這就是我撰寫這篇小論文的動機。

二、研究方法

（一）利用　Visual Studio C# Windows Form　應用程式開發設計。
（二）熟悉　Visual Studio C#　的程式編寫環境。
（三）了解七段顯示器的編碼與解碼原則。
（四）button 為內建的按鈕元件，按下後會觸發當中所寫的程式。
（五）rectangleShape 為外掛的繪圖元件，可繪出矩形樣式。

三、研究流程

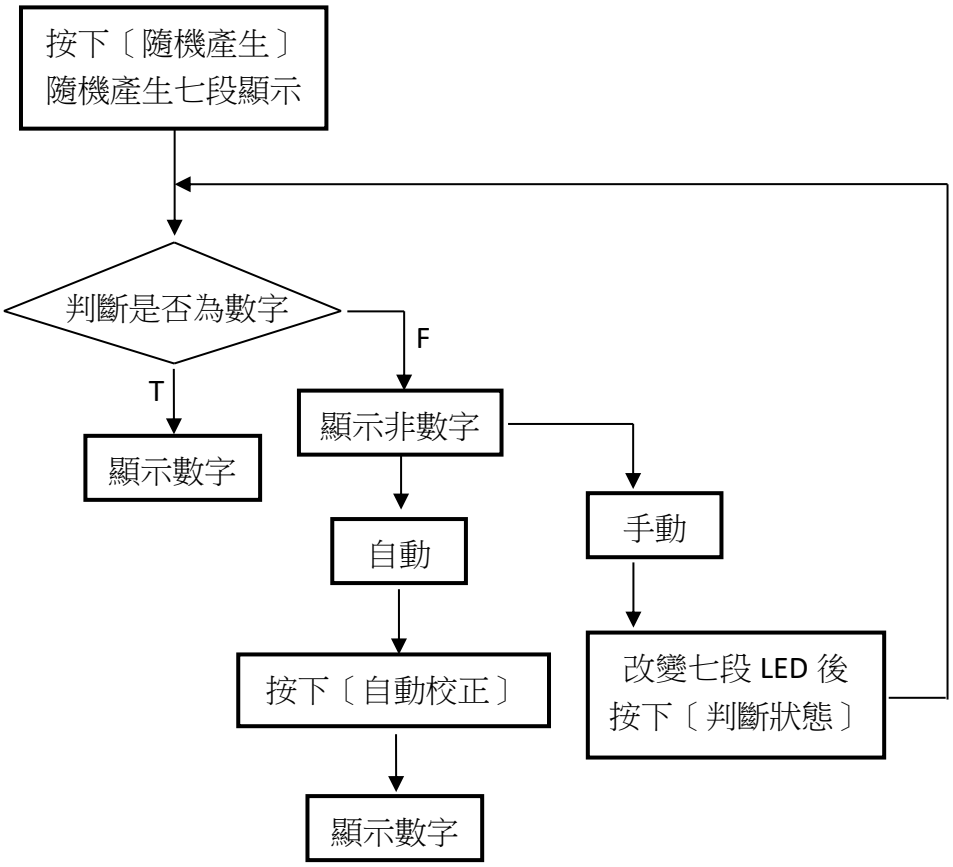


貳、正文

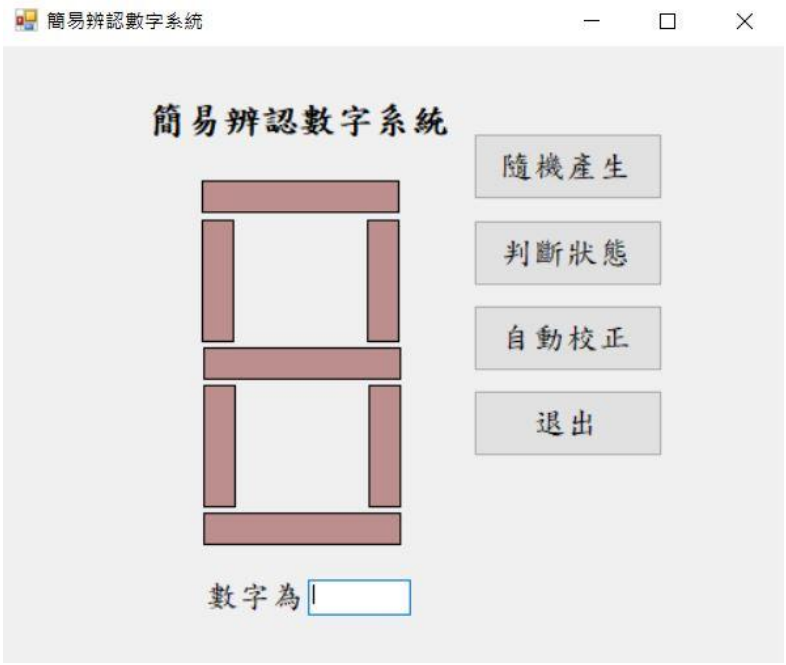
一、辨認數字系統介紹與問題解析

本系統為設計一個判斷七段顯示器顯示是否為數字的視窗軟體，七段顯示器的每一段 LED 都具有 ON 和 OFF 功能（亮或不亮）。視窗中的〔隨機產生〕功能讓七段隨機顯示，在系統自動判斷下如果是數字則顯示數字，不是則顯示非數字。視窗中的〔判斷狀態〕功能為使用者自行點擊七段顯示器的 LED 後，判斷現在七段顯示器是數字或非數字。視窗中的〔自動校正〕功能為自動將七段顯示器非數字狀態改為數字。視窗中的〔退出〕功能為退出系統。

要利用軟體程式模擬上述的功能，首先要排出七段顯示器 LED 的圖案、4 個按鈕和 1 個文字輸出方塊，排完後要寫出可以讓七段顯示器產出亂數、點擊七段顯示器 LED 可變色、自動更正為數字狀態以及判斷數字非數字的程式，因為按鈕都要判斷數字或非數字，所以把判斷式寫在副程式裡較合適，系統執行流程圖如圖（一），圖形化介面如圖（二）。



圖（一）數字辨認系統架構自繪圖



圖（二）圖形化介面自製圖

二、系統方塊圖各功能介紹

（一）判斷狀態　副程式介紹

```
void judge()  {
int total = 0;
if (rectangleShape1.FillColor == Color.Red) total += 64;
if (rectangleShape2.FillColor == Color.Red) total += 32;
//如上依序寫法
if (rectangleShape7.FillColor == Color.Red) total += 1;
      int[] ans = { 0, 1, 1, 2, 3, 4, 5, 6, 6, 7, 8, 9, 9 };
      int[] y = { 126, 48, 6, 109, 121, 51, 91, 95, 31, 112, 127, 123, 115 };
      for (int i = 0; i < 13; i++)  {
            if (total == y[i])  {
                  textBox1.Text = "" + ans[i];
                  break;  }
            else
                  textBox1.Text = "非數字";  }
}
```

因為按鈕都會用到判斷數字的程式，所以將這段程式放在 void judge()中加以應用。total 的起始值是 0，total += 64 的寫法等於 total = total + 64，本系統用的

演算法為七段顯示器以 2 進制表示，rectangleShape1 為最高位元，rectangleShape7 為最低位元，依權重來加權，total 所加的數也跟著改變。範例：二進制 7 位元若要顯示數字 1，則 2 進制表示為 0110000，rectangleShape2 以及 rectangleShape3 為 ON，其他為 OFF，則 total = 48。在按鈕的程式下呼叫副程式 judge()，就可判斷是否為數字了。

（二）七段顯示 LED 變色　副程式介紹

```
void r(int x, int sw)  {
   string y = "rectangleShape" + x;
   RectangleShape led =
   shapeContainer1.Shapes.OfType<RectangleShape>().FirstOrDefault(o =>
   o.Name == y);
   switch (sw) {
        case 0: led.FillColor = Color. RosyBrown; break;
        case 1: led.FillColor = Color.Red; break;     }
}
```

將這段程式放入 void r(int x, int sw)，因為在編寫程式中常用 rectangleShape1.FillColor = Color.Red 的用法，所以將這一行程式化為副程式的用法變成 r(1,1)，就可以將指定的 LED 變色，範例：r(1,0)是將第一個 LED 變成淺紅色（如同真實 LED 狀態為 OFF）、r(5,1)是將第五個 LED 變成亮紅色（如同真實 LED 狀態為 ON）。

（三）七段顯示 LED 變色　共用介紹

剛開始我用 button（即為按鈕）當作七段顯示器的 LED，後來發現使用 rectangleShape 當 LED 會比較合適，為了要把 button 上的預設字清除、改變顏色，就會花費許多時間。rectangleShape 方便許多，預設值就完全符合所要求的。總共有 7 個 LED 的程式碼要寫，如下是 rectangleShape1_Click 的程式：

```
if (rectangleShape1.FillColor == Color. RosyBrown)
    r(1,1);   //使用副程式
    else
        r(1,0);
```

若每個 LED 都這樣寫，將多出多達 42 行程式碼，因此我採用另一種方法，將每一個 LED 的事件都設為 rectangleShape1_Click，並在這個方格的程式裡寫：

```
RectangleShape rec = (RectangleShape)sender;
if (rec.FillColor == Color. RosyBrown)
    rec.FillColor = Color.Red;
    else
        rec.FillColor = Color. RosyBrown;
```

這麼一來，不必 7 個 LED 都寫相同的判斷式就能有相同的功能了。但這裡不能使用像是 r(1,0)的副程式，因為副程式中指定了第幾個 LED 做變化，而這裡是用類似代稱的方式讓 LED 變色。

（四）〔隨機產生〕功能介紹

```
Random ran = new Random();
if (ran.Next(0, 2) == 1) r(1, 1);     //使用副程式
    else r(1, 0);
```

若要讓 1 個 LED 隨機顯示 ON 或 OFF，必須要有個隨機產生數字，並設成為當亂數為某數時，此 LED 顯示 ON。我將每個 LED 的 ON 跟 OFF 的機率一致，這樣比較有較大的機率出現數字的樣式。其中 ran.Next(0, 2)表示產生 0~1 的亂數，當亂數值為 1 時，第一個 LED 顯示亮紅色；為 0 時，顯示淺紅色。再來是系統自動判別是否為數字，那麼在程式末端加上 judge()即可。

（五）〔判斷狀態〕功能介紹

```
judge();
```

〔判斷狀態〕的功能就是副程式 judge()，這麼在這個按鍵底下只須寫上一行，就可以優化重複程式碼了。

（六）〔自動校正〕功能介紹

與〔隨機產生〕、〔判斷狀態〕按鈕不同的是，〔自動校正〕按鍵必須將目前七段顯示器非數字狀態轉變為數字，如同除錯的功能，那麼必須在判斷數字副程式中間加上以下程式：

```
if (total > 6 && total < 31) {
    total = 31;
```

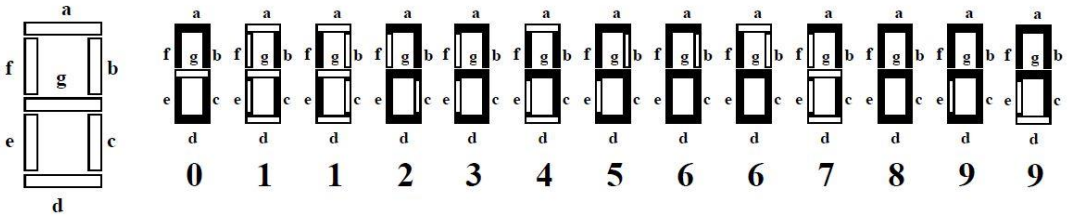
r(1, 0); r(2, 0); r(3, 1); r(4, 1); r(5, 1); r(6, 1); r(7, 1); }

非數字的值不會在 y[] 陣列中，因此必須把非數字的 total 改為在 y[] 陣列中的數字，並把七段顯示器的 LED 狀態改變成對應的數字，之後判斷是否在 y[] 陣列中時，就是在找更正後的七段顯示器狀態，是哪一個數字需要被顯示出來。

三、數位數字辨認系統實驗過程及結果呈現

（一）判斷數字的設計流程

1. 首先要知道判斷式數字的樣式並算出其 2 進制所表示的數字，再由 2 進制轉 10 進制後計算，如圖（三）所示。



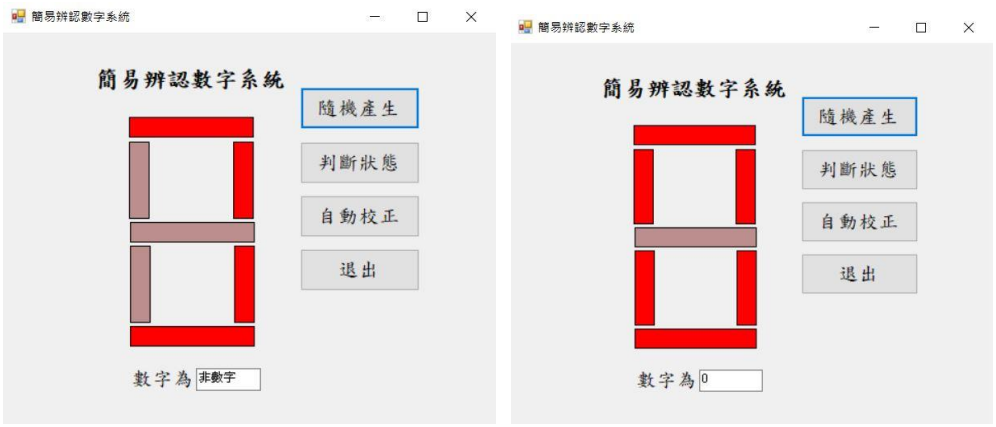
圖（三）數字樣式

2. 本程式使用兩個陣列，ans[] 為輸出的數字，y[] 陣列為對應 ans[] 陣列的 10 進制數字，算法以圖（三）中的 a 為最高位元(MSB)，權值為 64；b 的權值為 32 …… f 為 2；g 為 1。

3. 在讀取各個 LED 是否 ON 時，用 total 來計算此 LED 的權值，如此一來便可得知經由累加後的 10 進制數，再一一比較 total 是否與 y[] 陣列的數字符合，如果有符合，就顯示對應 ans[] 陣列的數字，並使用 break 來停止比較。

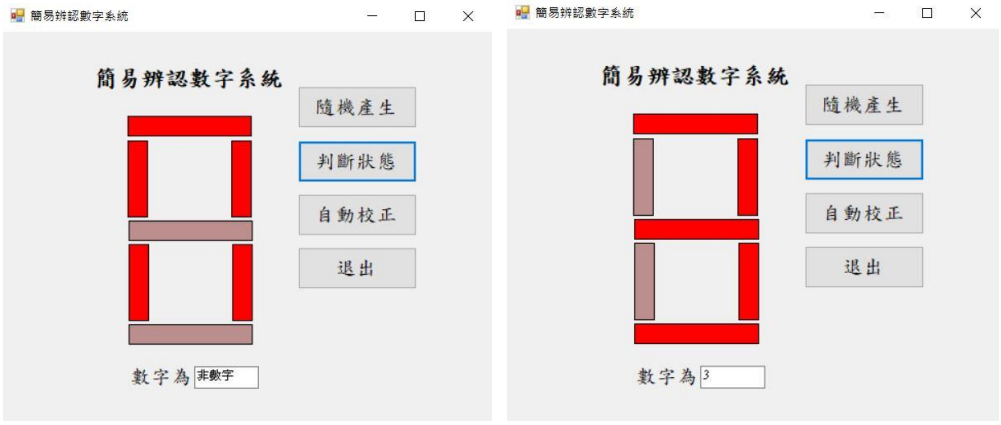
（二）實驗結果呈現

1. 在〔隨機產生〕按鈕中，隨機顯示 LED 狀態並自動判斷是否為數字，如圖（四）所示。



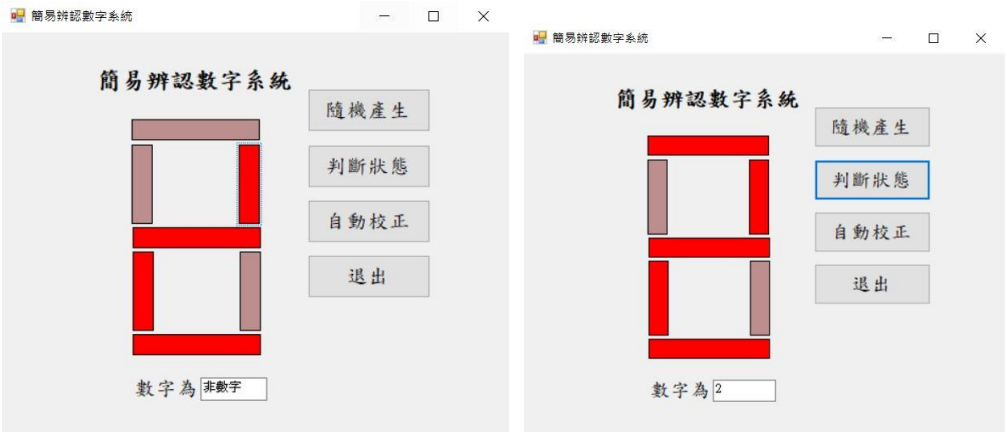
圖（四）〔隨機產生〕按下後自動判斷非數字和判斷為數字

2. 在〔判斷狀態〕按鈕中，為按下按鈕判斷是否為數字，如圖（五）所示。



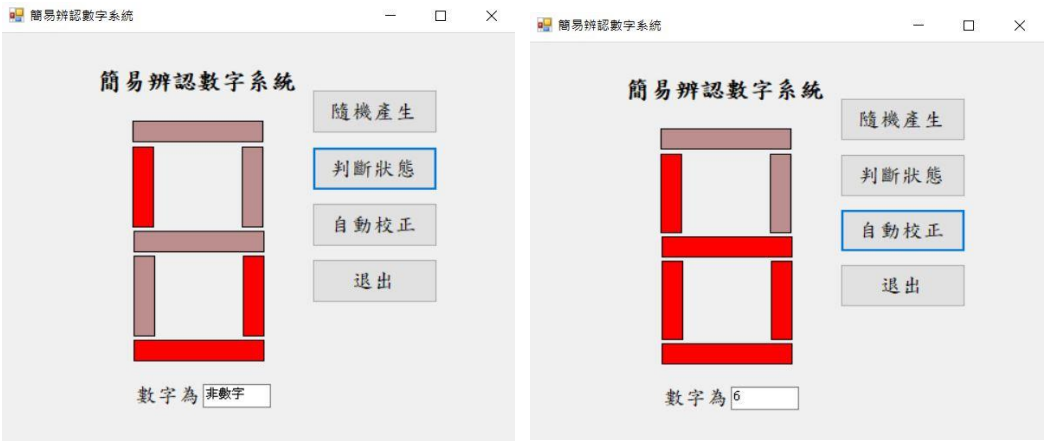
圖（五）〔判斷狀態〕按下後判斷非數字和判斷為數字

3. 而七段顯示 LED 也具有手動校正的功能，點擊後變色，如圖（六）所示。



圖（六）點擊七段顯示 LED 手動校正

4. 在〔自動校正〕按鈕中，將非數字狀態為數字狀態，如圖（七）所示。



圖（七）〔自動校正〕按下後將非數字自動更正為數字

參、結論

一、心得

在設計這個程式，雖然接觸過 Dev C 環境，但用 Dev C 圖形化界面的處理較不方便，換成 VC#環境後，圖形化介面較容易實現，將 Dev C 所學套用在 VC#中。相較之下，VC#程式架構也明確許多，需要適應的是兩者之間使用函式庫方式的不同，也能做出相同的功能。

撰寫程式從無到有，從測試功能屢次失敗到最後的成功，且需有耐心地進行邏輯演算、程序推演及除錯。失敗乃為成功之母，因為失敗才知道哪裡需要改進，過程中培養挫折感、上進心、解決問題的能力和程式邏輯的運算，成就感使得無數次失敗的滋味化為甜美，這就是我學習寫程式的主要原因。

二、學習成果

（一）從本次撰寫小論文過程中，學習到利用 Visual Studio C#中 Form 應用程式圖形的程式設計。

（二）在寫程式的過程中，了解到只有在 button 按下的事件觸發，對應的事件程式才會動作。

（三）原本使用 button 元件來模擬七段顯示器的 LED，後來使用 rectangleShape 元件，發現比 button 元件方便許多。

（四）將七段顯示器每個 LED 的觸發事件設為同一個 LED 的觸發事件下，並在此

LED 下作出改變，便可使每個 LED 觸發動作相同，也使用到副程式，優化重複程式碼。

（五）將七段顯示器的 ON 與 OFF 狀態的程式規劃成副程式，縮短其他需要使用相同功能的地方，方便閱讀。

三、未來方向

本次利用數位邏輯七段顯示器解碼的應用，做出模擬真實七段顯示器狀態的程式，在目前高二課程中，教授到 Arduino 微控制板晶片控制實習、CPLD 邏輯設計等科目，套用這些所學的專業知識，設計出更多程式，也會嘗試更多的題目。用軟體來模擬硬體功能，可給初學者模擬真實電路的狀態，先了解工作原理，優點是減少接線、邏輯閘製作和 IC 的耗材等，但缺點是模擬功能不盡然與硬體功能相符。

未來是 AI 與大數據的時代，隨著人口老化、少子化等問題，AI 機器人也會大量生產，其背後需要大量的程式設計的人才。為了要讓自己更具職場競爭力，提升自己編寫程式的能力更顯重要，多元的接觸各種工具製作程式，多方面接觸程式設計，對於未來的幫助也更多。

肆、引註資料

註一、蕭柱惠（2015）。**數位邏輯實習**。新北市：台科大圖書

註二、蕭柱惠（2015）。**數位邏輯**。新北市：台科大圖書

註三、黃建庭（2018）。**程式設計—使用 Visual C# 2017**。新北市：全華圖書

註四、勁樺科技。**入門首選 C 語言程式設計**。新北市：台科大圖書

註五、簡易數字辨認系統題目。**全國高級中等學校技藝映賽資訊平台/歷屆考題**。

取自 https://reurl.cc/p3W64