

# 實驗三 GPIO 輸入

## 1. 學習重點

- 熟悉組合語言中的指令。
- 了解 4\*4 鍵盤掃描原理。
- 了解 Bounce 及 Debounce。

## 2. 材料清單

表 3-1、實驗二基礎題所需材料清單

器材名稱		數量
AT89S51		1
12MHz 石英震盪器		1
按壓開關		1
4*4 鍵盤		1
單顆七段顯示器		1
電阻	1k $\Omega$	12
	10k $\Omega$	1
電容	20pF	2
	10 $\mu$ F	1

## 3. 元件原理

### 按壓按鈕 ( Tactile Switch )

按壓按鈕共有 4 支接腳 ( 圖 2 中標示①、②、③、④ )。在未按下按鈕的情況下，接腳①與接腳②為互通，接腳③與接腳④為互通。當按下按鈕時，中間接點接通，亦即 4 支接腳皆為互通。

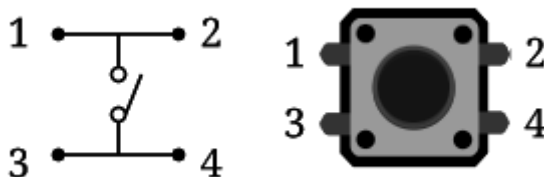


圖 3-1、按壓按鈕上視圖及內部電路示意圖

## 彈跳現象 ( Bounce )

在理想狀態下，若按鈕一邊接地，另一邊接 8051 輸入接腳。在沒有按下按鈕時，8051 輸入接腳收到高電位信號 ( High )。當按下按鈕時，兩邊接通，8051 輸入接腳收到低電位信號 ( Low )。放開按鈕，則回到高電位信號 ( High )。

實際上信號在趨於穩定前，會有雜訊使得 8051 接收到的信號忽高忽低，造成按壓信號的誤判，這種現象稱為彈跳現象 ( Bounce )，如圖 3-3。要避免此種現象，一般有兩種防彈跳 ( Debounce ) 方法，一種為硬體式，一種為軟體式，本實驗採用的是軟體式。

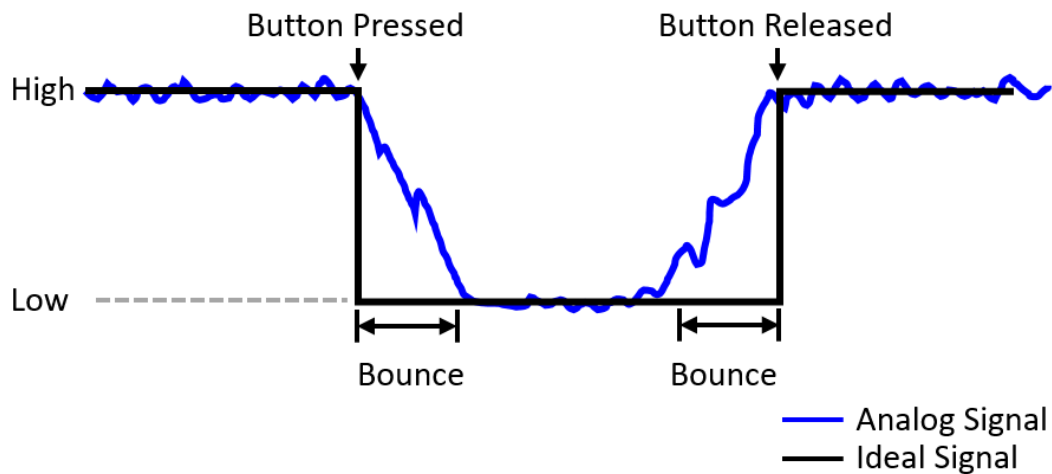


圖 3-2、按壓按鈕訊號圖

由上圖 3-3 的理想信號與類比信號的比較可知，在按下按鈕時以及放開按鈕時，會出現彈跳現象 ( Bounce ) 干擾判斷，彈跳部分大約為 10 ~ 20 毫秒。

## 軟體式防彈跳

彈跳現象發生在按下按鈕及放開按鈕的時候，只要利用延遲副程式，避免在彈跳現象發生時執行下一行程式碼，即可達到防彈跳的效果。以本實驗為例，在按下鍵盤數字時，按鈕會產生彈跳現象，此時執行大約 20 毫秒的延遲，等到信號穩定在 Low 的時候再讀取按鍵狀態，就不會因為雜訊而誤判使用者按了多次鍵盤。

## 掃描 4\*4 鍵盤

圖 3-4 (左) 為本實驗所使用的 4\*4 鍵盤樣式，由圖 3-4 (右) 可看出各接腳所對應的行向量與列向量。

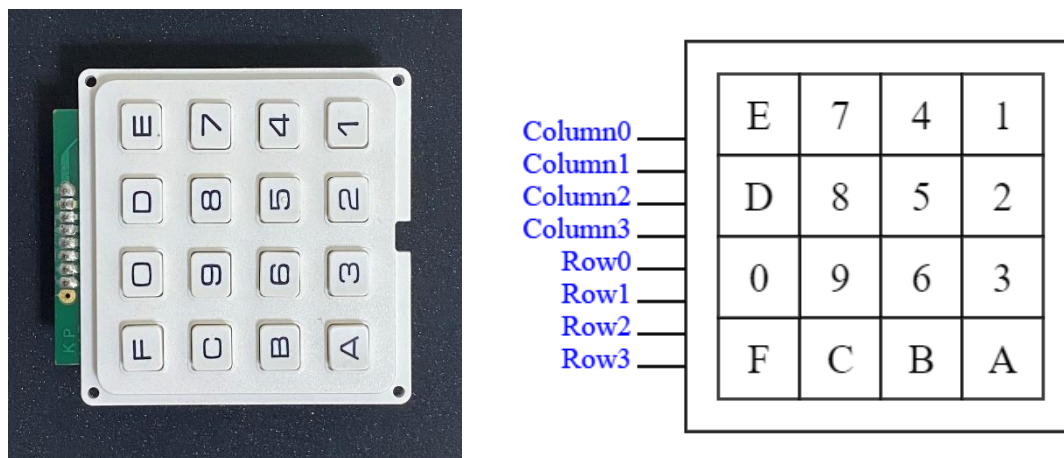


圖 3-3、4\*4 鍵盤樣式 (左) 及其腳位示意圖 (右)

鍵盤的掃描方式是從行 / 列送出掃描信號，由列 / 行讀取按鍵狀態。一次掃描一行 / 列的四個按鍵中哪一個被按下，若此四個按鍵都沒有按下，則掃描下一行 / 列的四個按鍵。掃描方式分為兩種，一種為「高態掃描」，共同接點連接到 GND；另一種為「低態掃描」，共同接點連接到 VCC。掃描過程中，若掃描到有按鍵被按下，則將按鍵解碼出來。本實驗使用計數方式解碼，亦即使按鍵上的數字就是計數所累積的數字。

## 高態掃描

將鍵盤的共同接點接上電阻後，再連接到 GND，使得輸出端 ( $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$ ) 保持在低電位 (圖 3-4)。掃描流程如下：

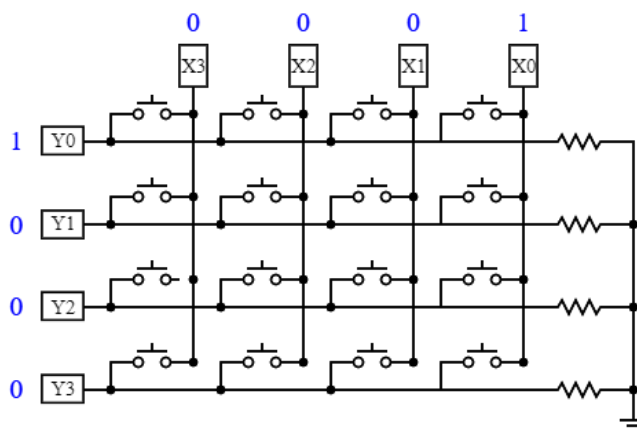


圖 3-4、高態掃描的結構

1. 將掃描信號 1000 輸入  $X_0$ 、 $X_1$ 、 $X_2$ 、 $X_3$ ，代表開始掃描  $X_0$  行：  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1000 (圖 3-4)，代表  $X_0$  行的  $Y_0$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0100，代表  $X_0$  行的  $Y_1$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0010，代表  $X_0$  行的  $Y_2$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0001，代表  $X_0$  行的  $Y_3$  列被按下。  
若  $X_0$  行都沒有被按下，則執行下一行的掃描。
2. 將掃描信號 0100 輸入  $X_0$ 、 $X_1$ 、 $X_2$ 、 $X_3$ ，代表開始掃描  $X_1$  行：  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1000，代表  $X_1$  行的  $Y_0$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0100，代表  $X_1$  行的  $Y_1$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0010，代表  $X_1$  行的  $Y_2$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0001，代表  $X_1$  行的  $Y_3$  列被按下。  
若  $X_1$  行都沒有被按下，則執行下一行的掃描。
3. 將掃描信號 0010 輸入  $X_0$ 、 $X_1$ 、 $X_2$ 、 $X_3$ ，代表開始掃描  $X_2$  行：  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1000，代表  $X_2$  行的  $Y_0$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0100，代表  $X_2$  行的  $Y_1$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0010，代表  $X_2$  行的  $Y_2$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0001，代表  $X_2$  行的  $Y_3$  列被按下。  
若  $X_2$  行都沒有被按下，則執行下一行的掃描。
4. 將掃描信號 0001 輸入  $X_0$ 、 $X_1$ 、 $X_2$ 、 $X_3$ ，代表開始掃描  $X_3$  行：  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1000，代表  $X_3$  行的  $Y_0$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0100，代表  $X_3$  行的  $Y_1$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0010，代表  $X_3$  行的  $Y_2$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0001，代表  $X_3$  行的  $Y_3$  列被按下。  
若  $X_3$  行都沒有被按下，則將掃描信號重新輸入為 1000，從步驟 1 重新開始掃描。

## 低態掃描

本實驗所使用的掃描方式。將鍵盤的共同接點接上電阻後，再連接到 VCC，使得輸出端 ( $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$ ) 保持在高電位 (圖 3-5)。掃描流程如下：

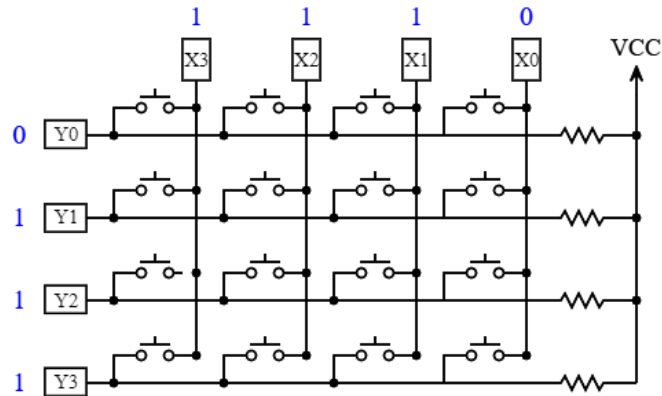


圖 3-5、低態掃描的結構

1. 將掃描信號 0111 輸入  $X_0$ 、 $X_1$ 、 $X_2$ 、 $X_3$ ，代表開始掃描  $X_0$  行：  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0111 (圖 3-5)，代表  $X_0$  行的  $Y_0$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1011，代表  $X_0$  行的  $Y_1$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1101，代表  $X_0$  行的  $Y_2$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1110，代表  $X_0$  行的  $Y_3$  列被按下。  
若  $X_0$  行都沒有被按下，則執行下一行的掃描。
2. 將掃描信號 1011 輸入  $X_0$ 、 $X_1$ 、 $X_2$ 、 $X_3$ ，代表開始掃描  $X_1$  行：  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0111，代表  $X_1$  行的  $Y_0$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1011，代表  $X_1$  行的  $Y_1$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1101，代表  $X_1$  行的  $Y_2$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1110，代表  $X_1$  行的  $Y_3$  列被按下。  
若  $X_1$  行都沒有被按下，則執行下一行的掃描。
3. 將掃描信號 1101 輸入  $X_0$ 、 $X_1$ 、 $X_2$ 、 $X_3$ ，代表開始掃描  $X_2$  行：  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0111，代表  $X_2$  行的  $Y_0$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1011，代表  $X_2$  行的  $Y_1$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1101，代表  $X_2$  行的  $Y_2$  列被按下。  
若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1110，代表  $X_2$  行的  $Y_3$  列被按下。  
若  $X_2$  行都沒有被按下，則執行下一行的掃描。

4. 將掃描信號 1110 輸入  $X_0$ 、 $X_1$ 、 $X_2$ 、 $X_3$ ，代表開始掃描  $X_3$  行：
- 若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 0111，代表  $X_3$  行的  $Y_0$  列被按下。
  - 若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1011，代表  $X_3$  行的  $Y_1$  列被按下。
  - 若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1101，代表  $X_3$  行的  $Y_2$  列被按下。
  - 若  $Y_0$ 、 $Y_1$ 、 $Y_2$ 、 $Y_3$  輸出為 1110，代表  $X_3$  行的  $Y_3$  列被按下。
- 若  $X_3$  行都沒有被按下，則將掃描信號重新輸入為 0111，從步驟 1 重新開始掃描。

## 4. 實驗內容

掃描讀取 4\*4 鍵盤之按壓輸入訊號 ( 1~9 )。當按下鍵盤上數字 1~9 時，單顆七段顯示器上顯示該數字。本實驗使用的單顆七段顯示器為共陽極。

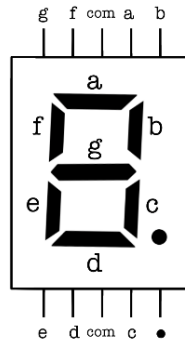


圖 3-6、單顆七段顯示器腳位圖 ( 共陽極 )

## 5. 實驗電路圖

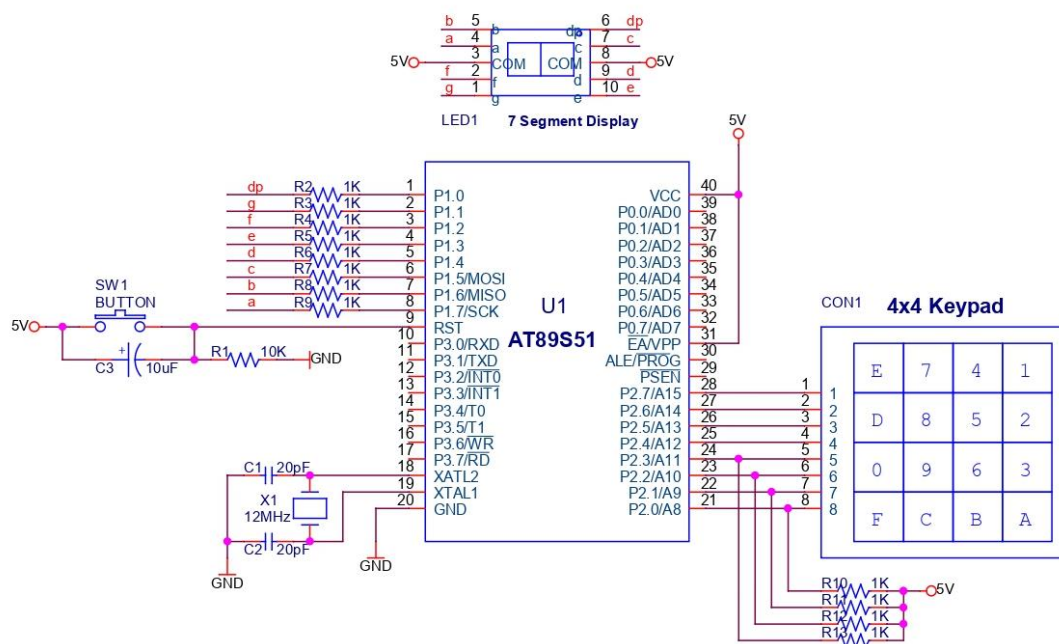


圖 3-1、實驗二基礎題參考電路圖

## 6. 軟體流程圖

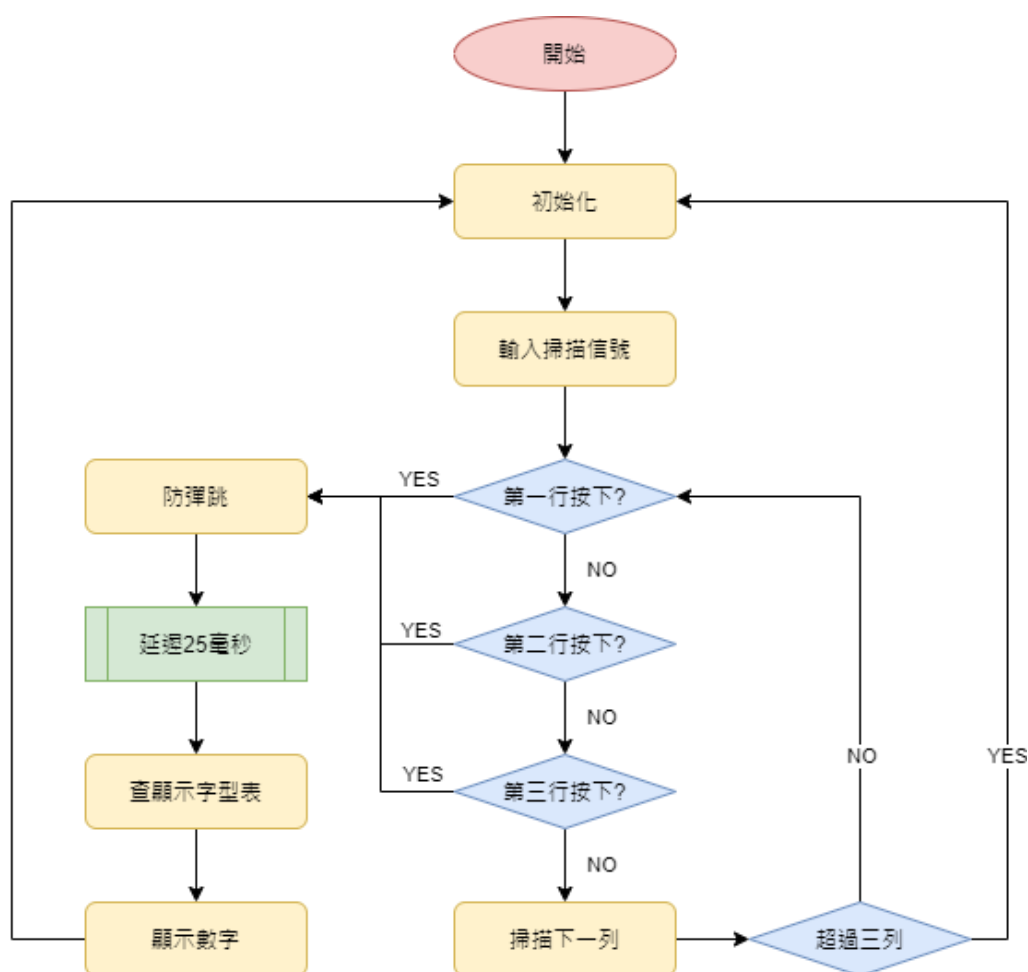


圖 3-2、實驗二基礎題參考軟體流程圖

## 7. 範例程式碼

1	<code>ORG 0</code>	<code>; start from 0000H</code>
2	<code>MOV DPTR, #TABLE</code>	<code>; DPTR point to TABLE</code>
3	<code>START: MOV R0, #0</code>	<code>; initialize typed signal</code>
4	<code>MOV R1, #3</code>	<code>; set 3 rows</code>
5	<code>MOV R2, #11110111B</code>	<code>; scan first row</code>
6	<code>SCAN: MOV A, R2</code>	<code>; move R2 to A</code>
7	<code>MOV P2, A</code>	<code>; input scanned signal</code>
8	<code>JNB P2.7, KEYIN</code>	<code>; scan column 1</code>
9	<code>INC R0</code>	<code>; add #001H into R0</code>
10	<code>JNB P2.6, KEYIN</code>	<code>; scan column 2</code>
11	<code>INC R0</code>	<code>; add #001H into R0</code>
12	<code>JNB P2.5, KEYIN</code>	<code>; scan column 3</code>
13	<code>INC R0</code>	<code>; add #001H into R0</code>
14	<code>RR A</code>	<code>; next row</code>
15	<code>MOV R2, A</code>	<code>; move A to R2</code>



```

16          DJNZ R1, SCAN          ; scan 3 rows
17          SJMP START            ; jump to scan keypad again
18  KEYIN:  CALL DEBOUNCE         ; call debounce function
19          MOV A, R0             ; get typed signal
20          MOVC A, @A+DPTR       ; get signal from TABLE
21          MOV P1, A             ; output displayed signal
22          SJMP START            ; jump to scan keypad again
23  DEBOUNCE: MOV R4, #50         ; move 50 times into R4
24  DE_500_us: MOV R3, #250      ; move 250 times into R3
25          DJNZ R3, $            ; jump to itself R3 times
26          DJNZ R4, DE_500_us    ; delay 25ms
27          RET                   ; return to main code
28  TABLE: DB 10011111B         ; display "1"
29          DB 00100101B         ; display "2"
30          DB 00001101B         ; display "3"
31          DB 10011001B         ; display "4"
32          DB 01001001B         ; display "5"
33          DB 01000001B         ; display "6"
34          DB 00011011B         ; display "7"
35          DB 00000001B         ; display "8"
36          DB 00001001B         ; display "9"
37          END

```

## 8. 整理的題目，選擇/是非題

- ( ) 將本實驗範例程式碼中的防彈跳延遲副程式刪掉對實驗結果沒有太大影響。
- ( ) 在實驗電路中，若鍵盤僅連接 8051 的輸入埠，不接向 VCC，對此範例程式沒有影響。