

# Lecture 10/03

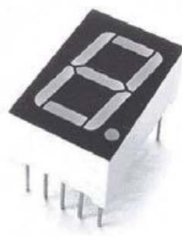
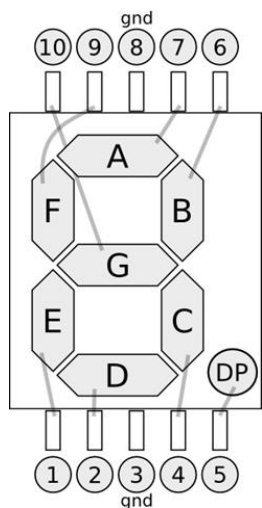
# Seven-Segment LED Display

FC Tien, YP Liu

Dept. of IE&M, Taipei Tech

# 7-Segment LED Display 七段顯示器

- 七段顯示器是以8個LED 排列組合而成，由順時鐘方向依序命名為: a, b, c, d, e, f, g 及 小數點 dp，另外上下各有一支COM 腳，以方便連結
- 七段顯示器有兩大類型（共陰極、共陽極 -- 並聯），往往依照功能需求來選擇所需要的七段顯示器，下面僅將對**共陰極**七段顯示器做介



名稱：七段LED

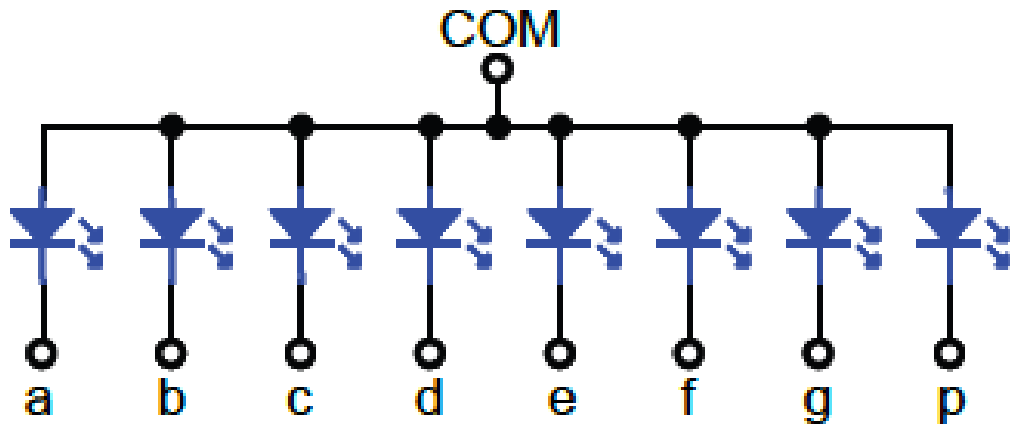
類型：類比元件

控制：將電壓轉為光能

極性：有正負之分，有接腳之別

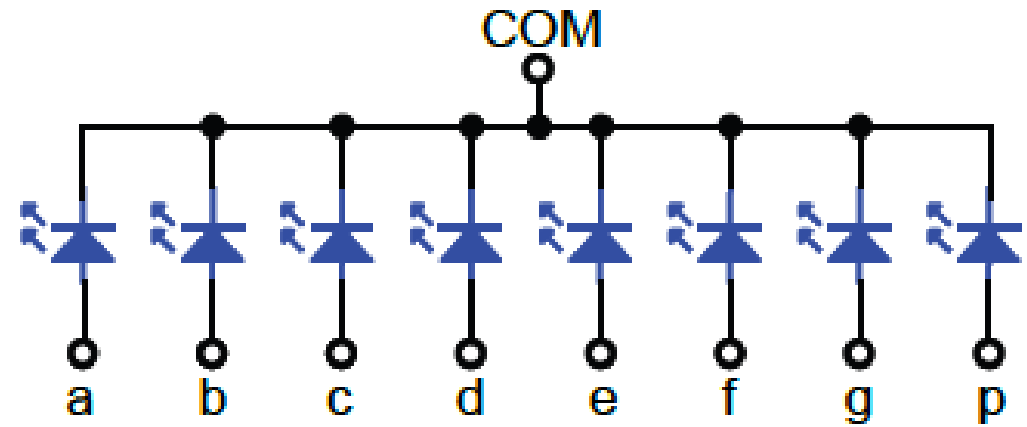
# 共陽極 vs. 共陰極

共陽極 及 共陰極的內部結構如下圖:



(a) 共陽極結構

COM 接5V + 電阻 and 其他接地

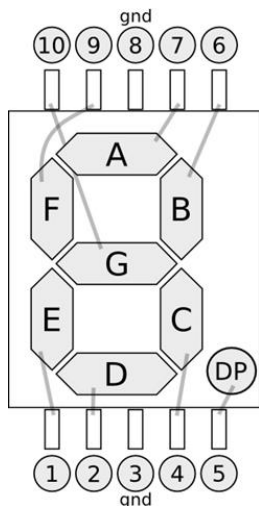


(b) 共陰極結構

COM 接Gnd + 電阻 and 其他皆5V

# 七段顯示器

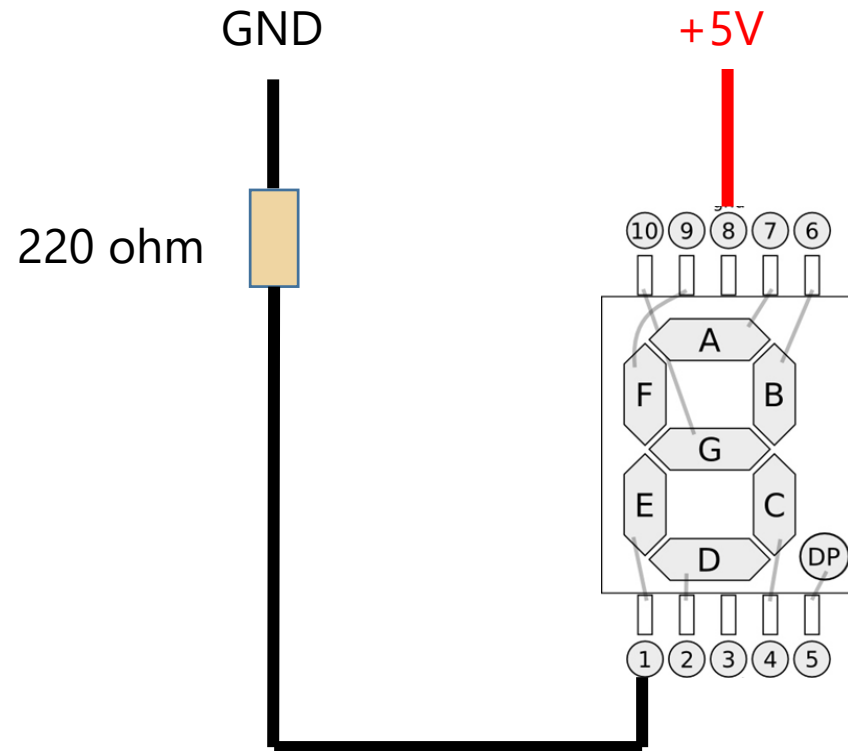
- 下面的圖分別為：七段顯示器正面圖、七段顯示器的數字顯示情形：



七段顯示器之顯示結果



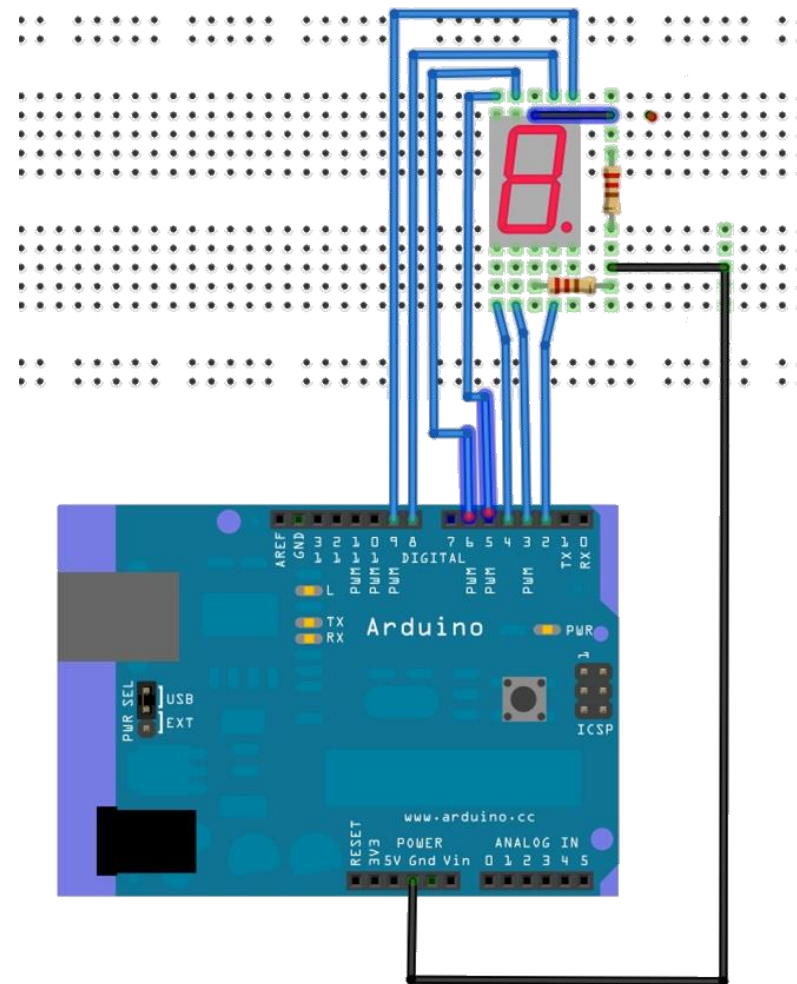
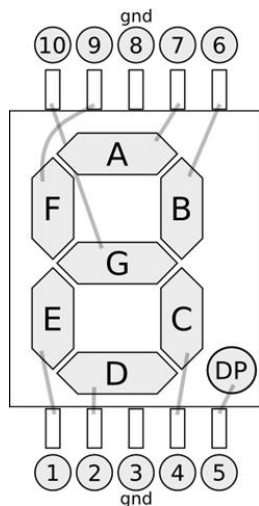
# 測試七段顯示器 (共陽極)



# 實驗 – Arduino 7段顯示器 (共陽極)

- 需求：顯示7，8，9。

Arduino Pin	七段顯示器 Pin
2	7 (A)
3	6 (B)
4	4 (C)
5	2 (D)
6	1 (E)
7	9 (F)
8	10 (G)
9	5 (DP)



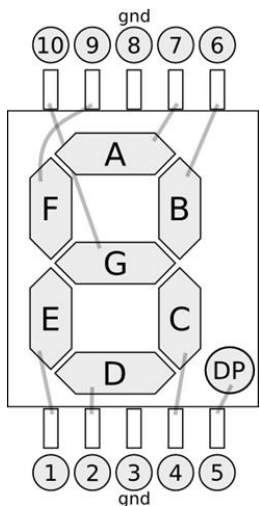
# 實驗 – Arduino 7段顯示器 (共陽極) 程式碼

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(2, OUTPUT);  
  pinMode(3, OUTPUT);  
  pinMode(4, OUTPUT);  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  pinMode(7, OUTPUT);  
  pinMode(8, OUTPUT);  
  pinMode(9, OUTPUT);  
  
  pinMode(10, OUTPUT);  
  
  digitalWrite(10, 1); //關閉7段LED-0  
}
```

```
void loop() {  
  //打開7段LED-0  
  digitalWrite(10, 0);  
  
  // 顯示數字 '9'  
  digitalWrite(2, 1);  
  digitalWrite(3, 1);  
  digitalWrite(4, 1);  
  digitalWrite(5, 0);  
  digitalWrite(6, 0);  
  digitalWrite(7, 1);  
  digitalWrite(8, 1);  
  delay(1000);  
  // 顯示數字 '8'  
  digitalWrite(2, 1);  
  digitalWrite(3, 1);  
  digitalWrite(4, 1);  
  digitalWrite(5, 1);  
  digitalWrite(6, 1);
```

```
  digitalWrite(7, 1);  
  digitalWrite(8, 1);  
  delay(1000);  
  
  //Turn Off 7-Segment LED 0  
  digitalWrite(10, 1);  
  
  // 暫停 2 秒鐘  
  delay(2000);  
};
```

# 實驗 – 7-Segment LED Display 查表程式



Arduino Pin	七段顯示器 Pin
2	7 (A)
3	6 (B)
4	4 (C)
5	2 (D)
6	1 (E)
7	9 (F)
8	10 (G)
9	5 (DP)

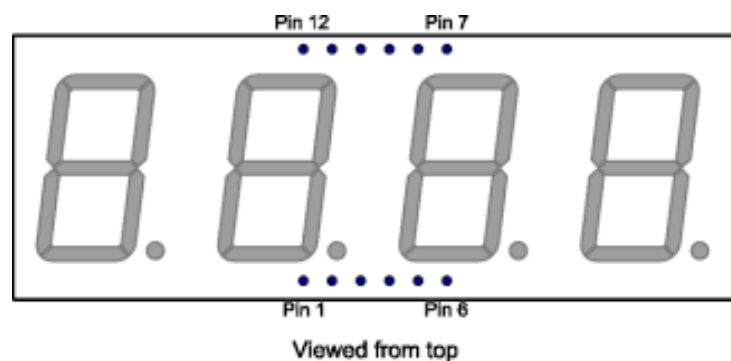
```
byte seven_seg_digits[10][7] =
{
    { 1,1,1,1,1,1,0 }, // = 0
    { 0,1,1,0,0,0,0 }, // = 1
    { 1,1,0,1,1,0,1 }, // = 2
    { 1,1,1,1,0,0,1 }, // = 3
    { 0,1,1,0,0,1,1 }, // = 4
    { 1,0,1,1,0,1,1 }, // = 5
    { 1,0,1,1,1,1,1 }, // = 6
    { 1,1,1,0,0,0,0 }, // = 7
    { 1,1,1,1,1,1,1 }, // = 8
    { 1,1,1,0,0,1,1 } // = 9
};
```

```
void sevenSegWrite(byte digit) {
    byte pin = 2;
    for (byte seg = 0; seg < 7; ++seg) {
        digitalWrite(pin,
            seven_seg_digits[digit][seg]);
        ++pin;
    }
}

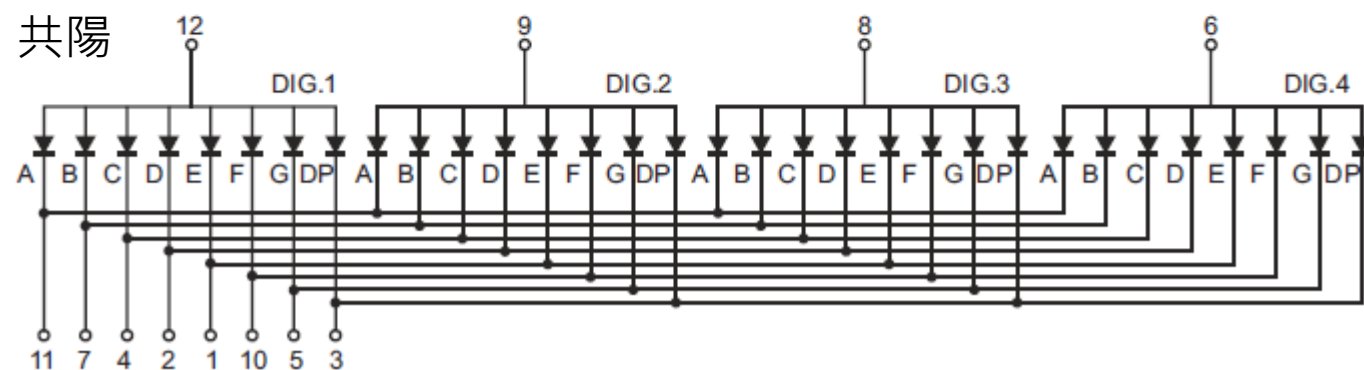
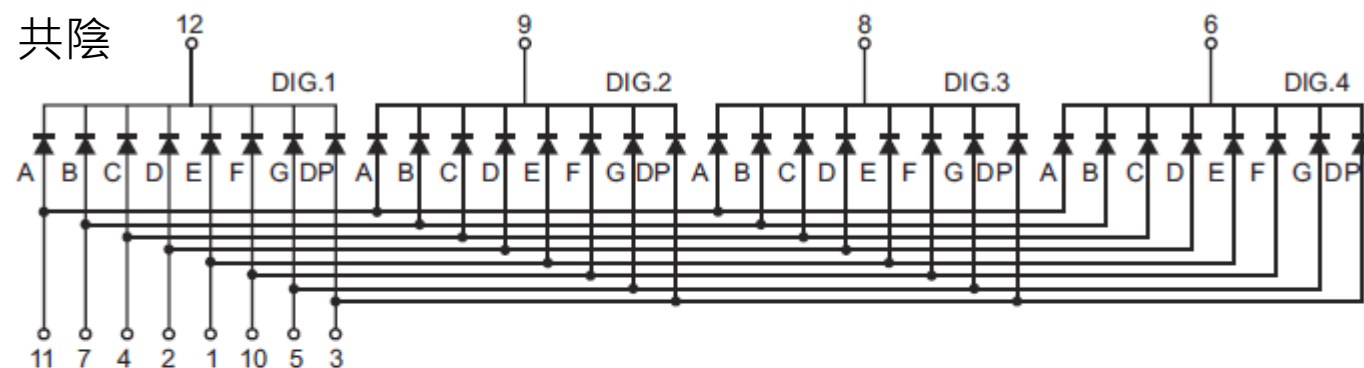
void loop() {
    sevenSegWrite(digit - 1);
    delay(1000);
}
```



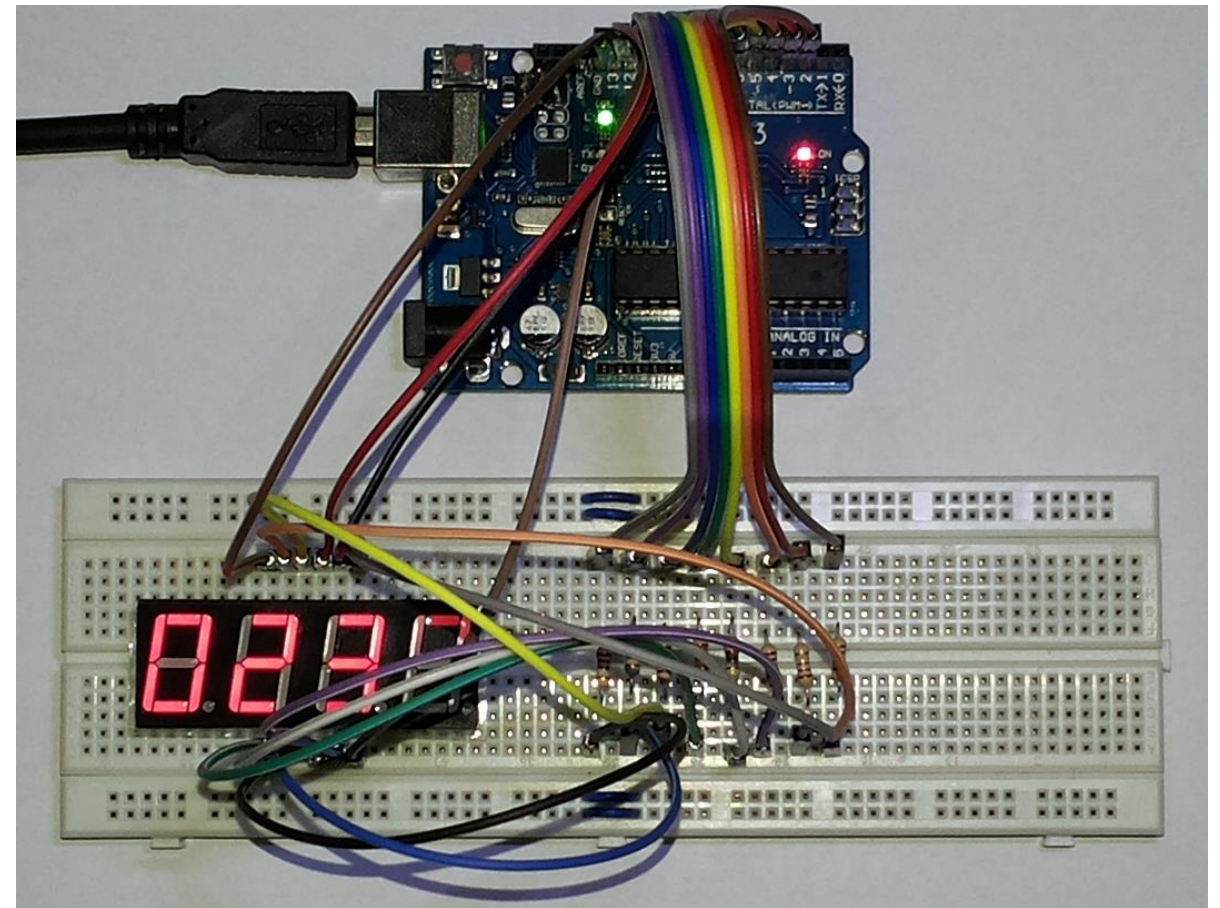
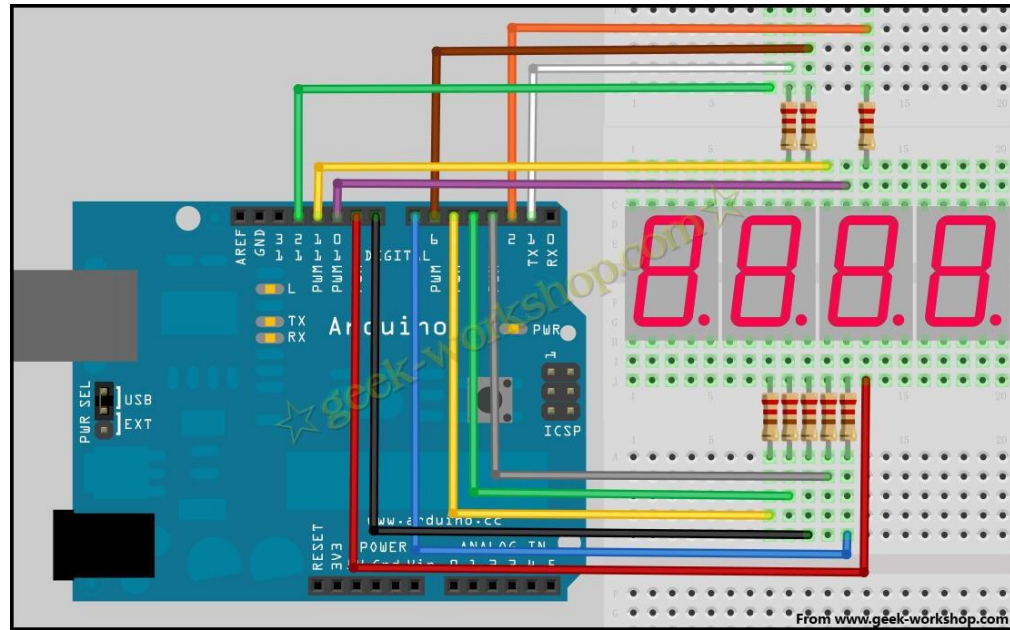
# 4位數 7段顯示器



## INTERNAL CIRCUIT DIAGRAM



# 4位數 7段顯示器



# 實驗 – 2位數 7段顯示器掃描程式

```
// 顯示2位數
void sevenSegShow(byte digit)
{
  byte high;
  byte low;

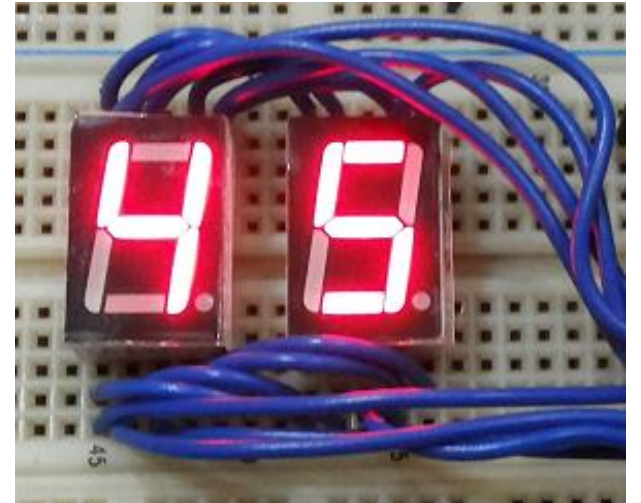
  high = digit / 10;
  low = digit % 10;

  digitalWrite(10, 0);
  digitalWrite(11, 1);
  sevenSegWrite(low);
  delay(10);

  digitalWrite(10, 1);
  digitalWrite(11, 0);
  sevenSegWrite(high);
  delay(10);
}
```

```
unsigned long startTime;
unsigned long duration;
void loop() {

  for (byte digit = 0; digit < 100; digit++) {
    startTime = millis();
    duration = 0;
    while (duration < 1000)
      {
        sevenSegShow(digit);
        duration = millis() - startTime;
      }
    //Serial.println(duration);
  }
}
```



# 實驗 – 7段顯示器 2位數顯示 掃描程式 計時器(Timer)

Arduino計時器 ( Timer ) 程式庫 <http://github.com/JChristensen/Timer>  
由Dr. Monk所開發，  
Jack Christensen修改

```
#include <Event.h>
#include <Timer.h>
```

```
Timer Tmr1;
Timer Tmr2;
byte Number;
```

```
void setup() {

    Tmr1.every(20, Scan7LED);
    Tmr2.every(1000, Tmr2Event);
    Number = 0;
}
```

```
void Scan7LED()
{
    sevenSegShow(Number);
}
```

```
void Tmr2Event()
{
    Number++;
}
```

```
void loop() {
    Tmr1.update();
    Tmr2.update();
}
```



# Homework

題目：如何改為4位數顯示？

