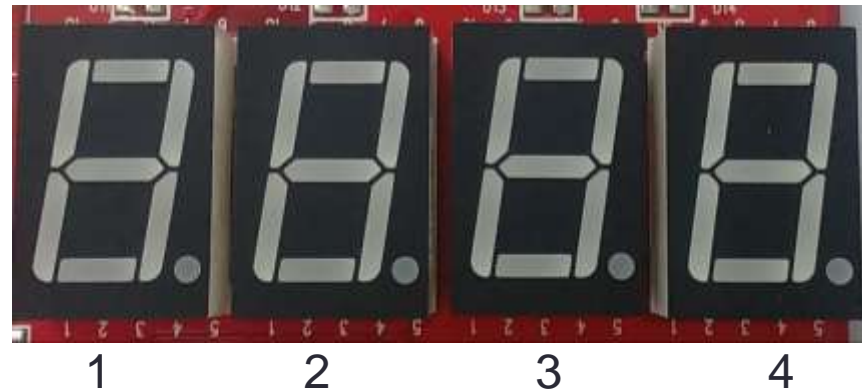


# 七段顯示器

---

# 接腳參數



接腳名稱 及 所在之bit位置	說明
E_GPC 4	第一顆七段顯示器 (最右邊)
E_GPC 5	第二顆七段顯示器
E_GPC 6	第三顆七段顯示器
E_GPC 7	第四顆七段顯示器 (最左邊)

Enable : 1

Disable : 0

同一時間，只能有有一個七段顯示器被選到(enable)

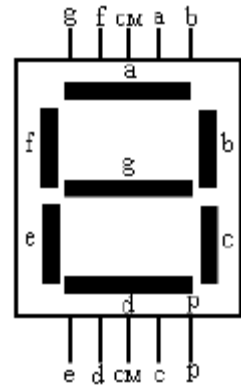
# 接腳參數

E\_GPE負責控制七段顯示器中每個位置的暗或亮

亮: 0

暗: 1

四個七段顯示器共用這些資料接腳



數字0~5

	g	e	d	b	a	f	dp	c
	7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	1	0
1	1	1	1	0	1	1	1	0
2	0	0	0	0	0	1	1	1
3	0	1	0	0	0	1	1	0
4	0	1	1	0	1	0	1	0
5	0	1	0	1	0	0	1	0

Bit位置

# 控制

- 暫存器
  - E\_GPC , E\_GPE
- 暫存器之存取
  - 指定某一暫存器
    - `tGPIO_C = (GPIO_T *)((uint32_t)GPIOA + (2*0x40));`
    - `tGPIO_E = (GPIO_T *)((uint32_t)GPIOA + (4*0x40));`
  - 於該暫存器中存取資料
    - `tGPIO_C->DOUT = 0xff10;`

# 舉例1

- 兩個數字切換顯示，在同一個七段顯示器

```
tGPIO_C->DOUT = 0xff10; //最右邊的七段顯示器
while(1)
{
    tGPIO_E->DOUT = 0x52;           //輸出圖形5
    DrvSYS_Delay(500000);

    tGPIO_E->DOUT = 0x6a;           //輸出圖形4
    DrvSYS_Delay(500000);
}
```

## 舉例2

- 顯示的資料，可用array存起來，再使用

```
int pattern[6]={0x82,0xee,0x07,0x46,0x6a,0x52};  
int index =0;
```

```
tGPIO_C->DOUT = 0xff10; //最右邊的七段顯示器
```

```
while(1)  
{  
    tGPIO_E->DOUT = pattern[index]; //從pattern中叫出所要顯示的圖形  
    DrvSYS_Delay(500000);  
    index = (index+1) % 6;  
}
```

## 舉例3 – 殘影問題

- 在2個七段顯示器上顯示 圖形6 和 5 相互交換顯示，  
要在第2個七段顯示器顯示圖形6時,由於資料還是0x52  
，所以在第2個七段顯示器有可能會看到圖形5

```
while(1)
{
    tGPIO_C->DOUT = 0xff10; //最右邊的七段顯示器
    tGPIO_E->DOUT = 0x52; //圖形5
    DrvSYS_Delay(500000);

    tGPIO_C->DOUT = 0xff20; //從右數第二個七段顯示器
    tGPIO_E->DOUT = 0x12; //圖形6
    DrvSYS_Delay(500000);
}
```

## 舉例3 – 解決殘影問題

- 先清除七段顯示器資料，在寫入要顯示的資料，來解決殘影

```
while(1) {  
    tGPIO_C->DOUT = 0x0000; //清空七段顯示器  
    tGPIO_E->DOUT = 0x52; //圖形5  
    tGPIO_C->DOUT = 0xff10; //最右邊的七段顯示器  
    DrvSYS_Delay(500000);  
  
    tGPIO_C->DOUT = 0x0000; //清空七段顯示器  
    tGPIO_E->DOUT = 0x12; //圖形6  
    tGPIO_C->DOUT = 0xff20; //從右數第二個七段顯示器  
    DrvSYS_Delay(500000);  
}
```