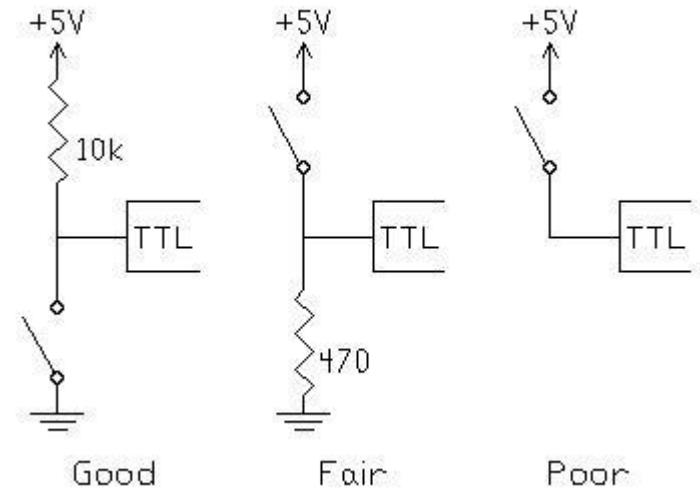


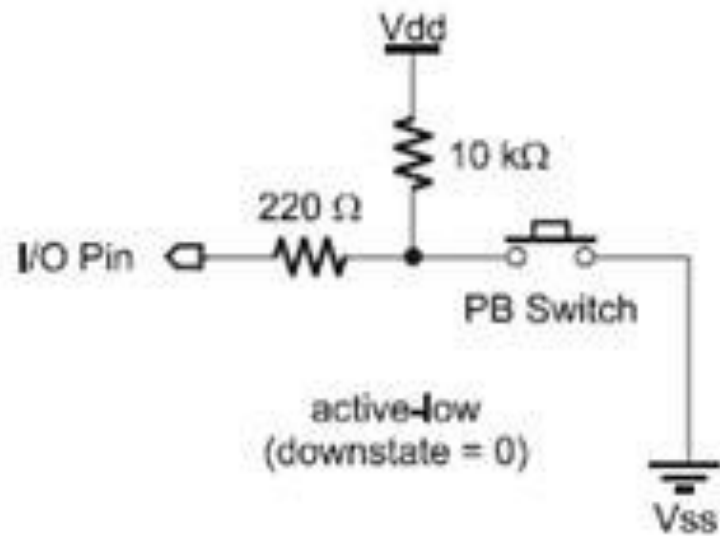
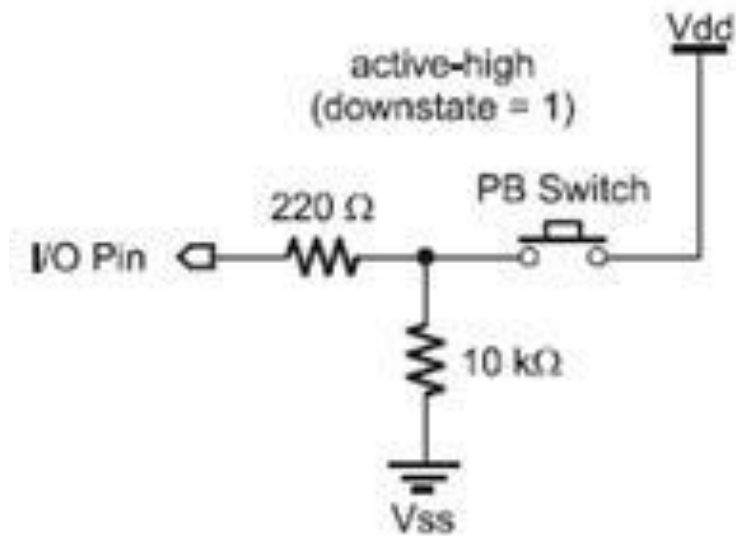
# 8051 輸入/輸出埠 LED與按鈕

2012.10.26修訂

# 8051和開關電路(input)

- 在good電路，開關open時，電流為 $5V/10k=0.5mA$ ，輸入高電位。開關close時，輸入低電位。
- 在fair電路，開關open時，電流 $1.6mA$ 通過的電阻(470)不能太大，否則電壓降可能超過 $0.9V$ 。當開關close時，輸入高電位，電流為 $5V/470=10.6mA$ 太大。
- 在poor電路，開關close時，輸入高電位。但是開關open時，輸入為浮動，通常為1。

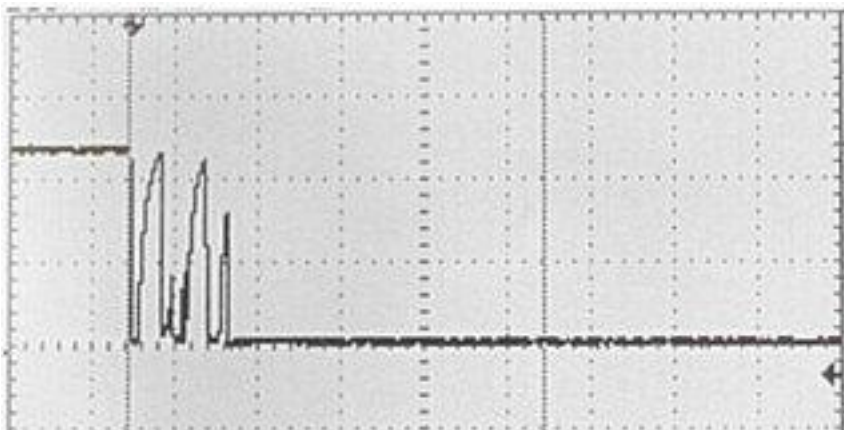




- 左圖鍵盤未按下時，輸入為0V；按下後，輸入為5V。
- 右圖鍵盤未按下時，輸入為5V；按下後，輸入為0V。

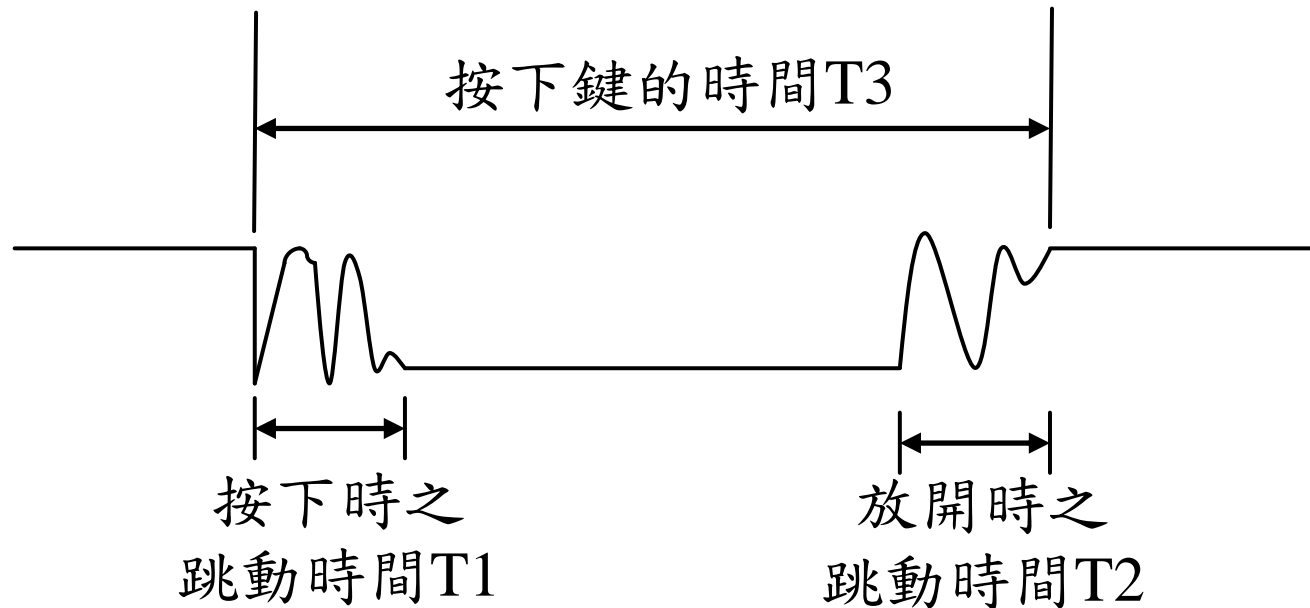
# 按鍵

- 按鍵是機械裝置，按壓後，在穩定之前，內部連結會在幾毫秒間來回彈跳。
- 在消除彈跳的這段時間裡，low和high的訊號都會偵測到，造成誤判。
- 為了避免這種情況發生，通常會加入約20ms的延遲，再開始判讀按鍵值。

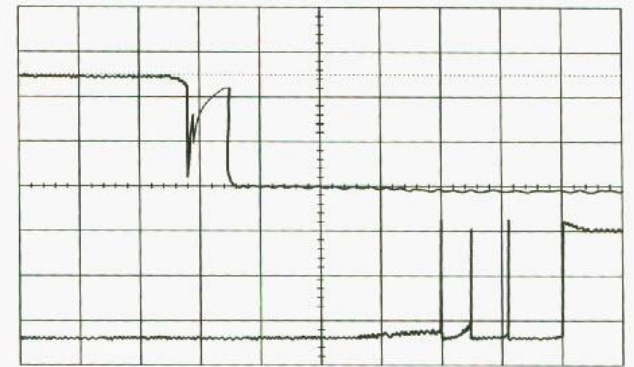
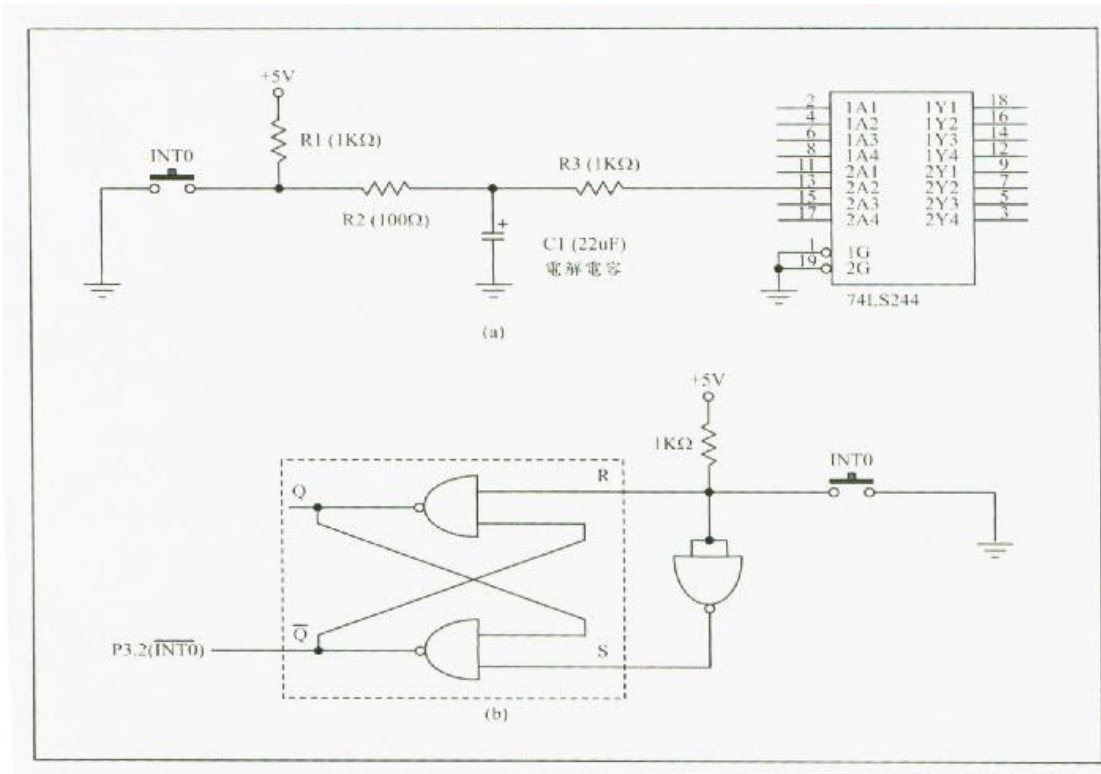


# 按鍵彈跳現象

- 硬體濾波：利用硬體濾波器產生較平滑的電壓變化。
- 軟體延遲：當軟體偵測到有鍵盤被按時，利用程式等待一段時間，讓按鍵之接點開關穩定後，再去讀取按鍵值。

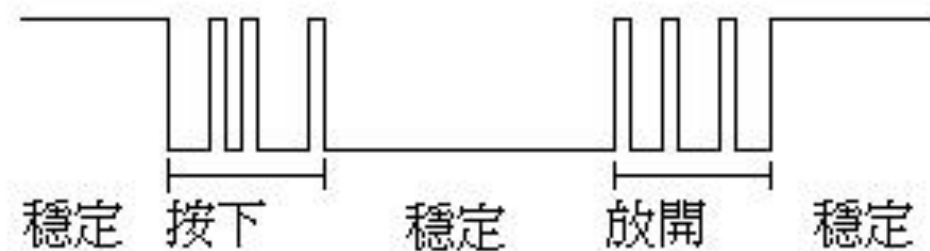


# 按鍵



由於按鍵是機械式動作，每次按下或放開時，可能因彈跳現象產生多次的信號，期間大約10ms。左圖使用硬體電路來處理彈跳現象。

# 軟體處理



```
while (key);           //如果鍵盤未按下，會一直重複執行
delay20ms;             // 延遲20ms
if(key){               //檢查鍵盤是否還是按下
    keypressed=0;      //只是雜訊，不是按下鍵盤
} else {
    keypressed=1;      //設定鍵盤旗標或呼叫鍵盤處理程式
    while(!key);       //等待鍵盤鬆開
    delay20ms;         //延遲20ms，
}
```

# 軟體處理

- 一般人的按鍵速度至多是10次/秒，亦即一次按鍵時間是100 ms，按下的時間可估算為50 ms。若取樣信號的週期為8 ms計，則可以取樣到6次。
- 所謂彈跳，就是按鍵正處於『ON』與『OFF』位置之間來回的振動，持續時間大約是10 ms，通常會在20ms 內結束而趨於穩定。
- 若設定每 4 ms 檢查一次按鍵狀態，如果連續 4 次的狀態都相同，則確認為新狀態，亦即相同狀態持續達16 ms 以上才作確認。



# 範例4\_5 8051 I/O： 計算彈跳的次數

/\*

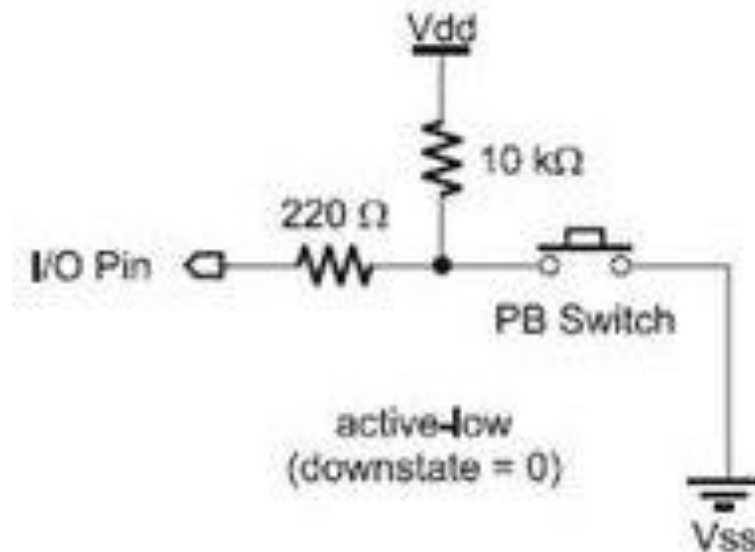
鍵盤：連接到P1^0

LED：連接到P2

功能：

1. 鍵盤按下時，計算彈跳的次數

\*/



# 範例 8051 IO： 計算彈跳的次數

```
18 sbit key = P1^0;
19
20 main()
21 {
22
23     unsigned char ci=0;
24
25     P2=~ci;
26
27     while(1){
28
29         while (key);    //如果鍵盤未按下，會一直重複執行
30
31         ci++;
32
33         while (!key);   //如果鍵盤未鬆開，會一直重複執行
34
35         P2=~ci;
36     }
37 }
```

## 範例4-3 8051 I/O： 按鍵輸入, LED閃爍

/\*

鍵盤：連接到P1^0

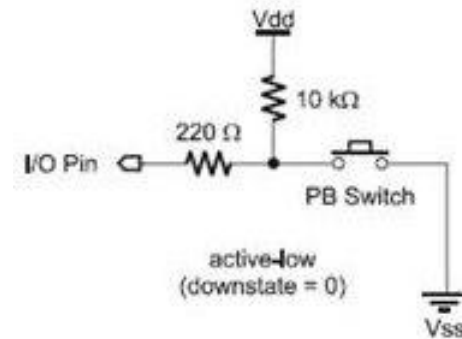
LED：連接到P2

功能：

1. 鍵盤按下時，LED 1-3-5-7變亮，其餘LED不亮
2. 再按一次時，LED 2-4-6-8變亮，其餘LED不亮
3. 再按一次時，反覆執行1，2
4. 使用軟體處理按鍵彈跳現象：

當偵測到低電位時，延遲20ms，若還是低電位，表示按鍵動作，否則為雜訊。

\*/



## 範例4-3 8051 IO： 按鍵輸入，LED閃爍

```
14 #include <STC12C5410AD.H>
15
16 #define FOSC 11059200L           //設定石英震盪器頻率
17 #define T20ms (65536-FOSC/12/50) //20ms, 1s=50*20ms
18 #define T20msH T20ms/256        //high byte
19 #define T20msL T20ms%256        //low byte
20
21 sbit key = P1^0;
22
23 main()
24 {
25     unsigned char keypressed;
```

## 範例4-3 8051 IO： 按鍵輸入, LED閃爍

```
27 P2=0x55;
28 TMOD = 0x01;           //set timer0 as model (16-bit)
29 //TL0 = T20msL;        //initial timer0 low byte
30 //TH0 = T20msH;        //initial timer0 high byte
31 //TR0 = 1;             //timer0 start running
32 //ET0 = 1;             //enable timer0 interrupt
33 //EA = 1;              //open global interrupt switch
34 while(1){
```

## 範例4-3 8051 I/O： 按鍵輸入，LED閃爍

```

36 while (key);           //如果鍵盤未按下，會一直重複執行
37
38 TL0 = T20msL;          //reload timer0 low byte
39 TH0 = T20msH;          //reload timer0 high byte
40 TR0 = 1;               //timer0 start running,
41 while( !TF0);          //延遲20ms
42 TR0=0;                 //timer0 stop running
43 TF0=0;                 //reset TF0
44
45 if(key){               //檢查鍵盤是否還是按下
46     keypressed=0;      //只是雜訊，不是按下鍵盤
47 } else {
48     keypressed=1;      //設定鍵盤旗標或呼叫鍵盤處理程式
49     P2=~P2;            //反相
50
51     while( !key);      //等待鍵盤鬆開

```

## 範例4-3 8051 I/O： 按鍵輸入, LED閃爍

```
53 TL0 = T20msL; //reload timer0 low byte
54 TH0 = T20msH; //reload timer0 high byte
55 TR0 = 1; //timer0 start running,
56 while(!TF0); //延遲20ms
57 TR0=0; //timer0 stop running
58 TF0=0; //reset TF0
59 }
60 }
61 }
```



# 範例4-4 8051 IO： 按鍵輸入, LED閃爍

/\*

程式：4\_4ButtonLed

鍵盤：連接到P1^0

LED：連接到P2

功能：

1. 鍵盤按下時，LED 1-3-5-7變亮，其餘LED不亮
2. 再按一次時，LED 2-4-6-8變亮，其餘LED不亮
3. 再按一次時，反覆執行1，2
4. 使用軟體處理按鍵彈跳現象：

若設定每 4 ms 檢查一次按鍵狀態，如果連續6次的狀態都是低電位，則確認為按鍵按下，亦即相同狀態持續達20-24ms 以上才作確認；否則為雜訊。確認按鍵後，略過25次(延遲100mA)，再處理下一次按鍵。

\*/



## 範例4-4 8051 IO： 按鍵輸入，LED閃爍

```
16 #include <STC12C5410AD.H>
17
18 #define FOSC 11059200L           //石英振盪器頻率
19 #define T4ms (65536-FOSC/12/250) //4ms, 1s=250*4ms
20 #define T4msH T4ms/256          //high byte
21 #define T4msL T4ms%256          //low byte
22
23 sbit key = P1^0;                 //按鍵
24 unsigned char count=0;           //紀錄已經按下幾次
25 unsigned char keypressed=0;      //紀錄是否按下
```

## 範例4-4 8051 IO： 按鍵輸入, LED閃爍

```
28  main()  
29  {  
30  
31      P2=0x55;           //01010101  
32      TMOD = 0x01;       //set timer0 as model (16-bit)  
33      TL0 = T4msL;       //initial timer0 low byte  
34      TH0 = T4msH;       //initial timer0 high byte  
35      TR0 = 1;           //timer0 start running  
36      //ET0 = 1;         //enable timer0 interrupt  
37      //EA = 1;          //open global interrupt switch  
38  
39      while(1){ I
```

## 範例4-4 8051 IO： 按鍵輸入，LED閃爍

```
41 while(!TF0);           //延遲4ms
42 TL0 = T4msL;           //initial timer0 low byte
43 TH0 = T4msH;           //initial timer0 high byte
44 TF0=0;                 //reset TF0
45
46 if(keypressed){        //鍵盤按下，延遲80ms
47     keypressed--;
48 } else {
49     if (key){           //鍵盤按下？
50         count=0;        //鍵盤未按下
51     } else {           //鍵盤按下
52         if(count){      //鍵盤是否第一次低電位
53             count--;
54             if(!count){  //鍵盤連續4次低電位
55                 P2=~P2;  //P2反相
56                 keypressed=20; //鍵盤按下，延遲80ms
57                 count=0;   //reset keyflag
58             }
59         } else {        //鍵盤第一次低電位
60             count=3;     //set counter，還要3次
61         }
62     }
```

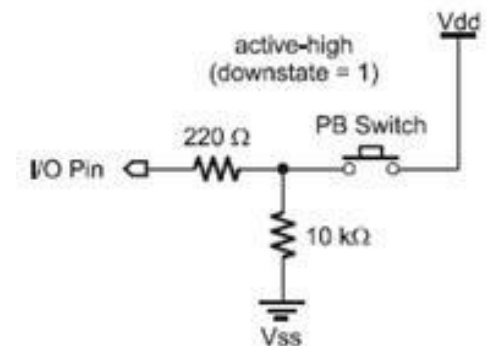
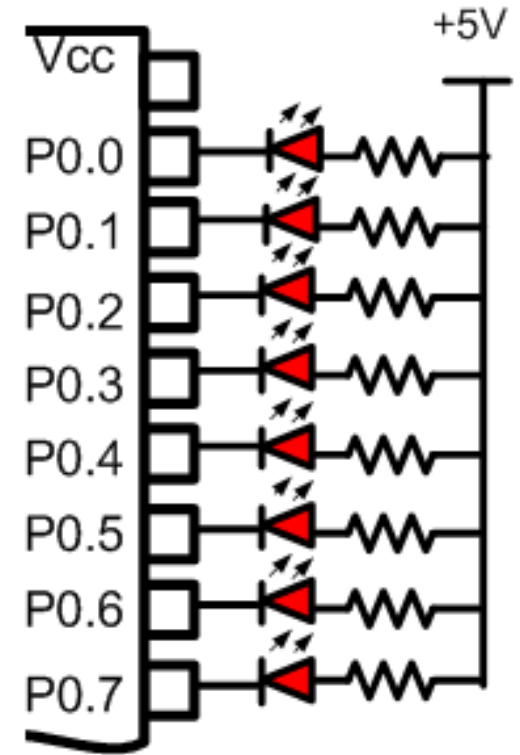
## 範例4-4 8051 IO： 按鍵輸入, LED閃爍

```
46 if(keypressed){ //鍵盤按下，延遲80ms
47     keypressed--;
48 } else {
49     if (key){ //鍵盤按下？
50         count=0; //鍵盤未按下
51     } else { //鍵盤按下
52         if(count){ //鍵盤是否第一次低電位
53             count--;
54             if(!count){ //鍵盤連續4次低電位
55                 P2=~P2; //P2反相
56                 keypressed=20; //鍵盤按下，延遲80ms
57                 count=0; //reset keyflag
58             }
59         } else { //鍵盤第一次低電位
60             count=3; //set Counter，還要3次
61         }
62     }
63 }
64
65 }
66 }
```



## 範例4-5

- 將LED連接到PORT2，當P2輸出0時，LED為亮；當P2輸出1時，LED為暗。
- 鍵盤：連接到P1<sup>0</sup>
- 功能：
  1. LED依序0, 1, 2, 3, 4, 5, 6, 7, 一次亮一個；間隔一秒，反覆執行
  2. 鍵盤按下時，LED按相反順序，一次亮一個；間隔一秒，反覆執行

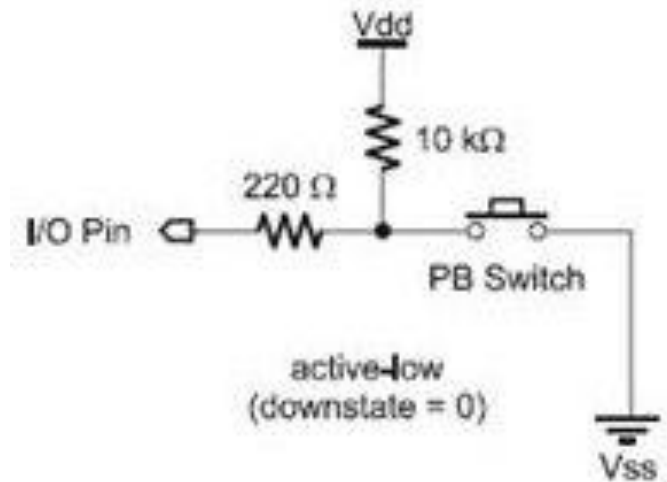
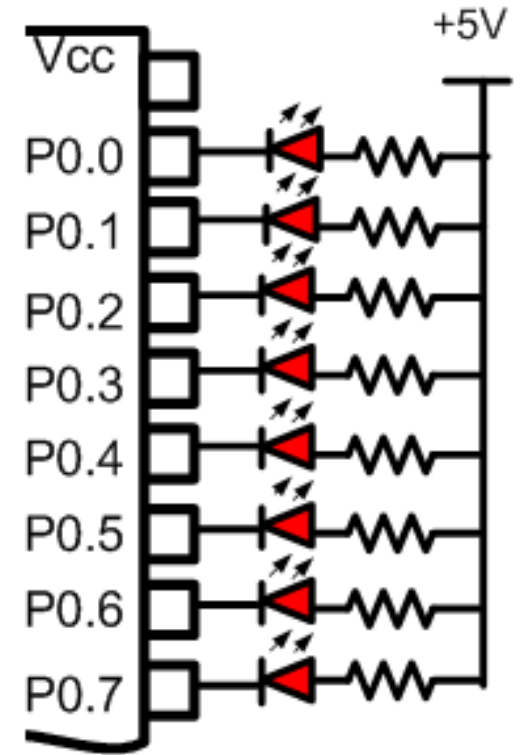


# 範例4-5

- LED依序0, 1, 2, 3, 4, 5, 6, 7, 一次亮一個；間隔一段時間，反覆執行。
- 問題：
- 若LED間隔1秒，使用for迴圈延遲時間，何時去檢查鍵盤的輸入？
- 在鍵盤輸入需要時間處理彈跳現象，會不會影響LED的計時？
- 解決方法：
- 使用定時器和中斷，不要依賴迴圈計時或者由CPU檢查計時器的旗標。

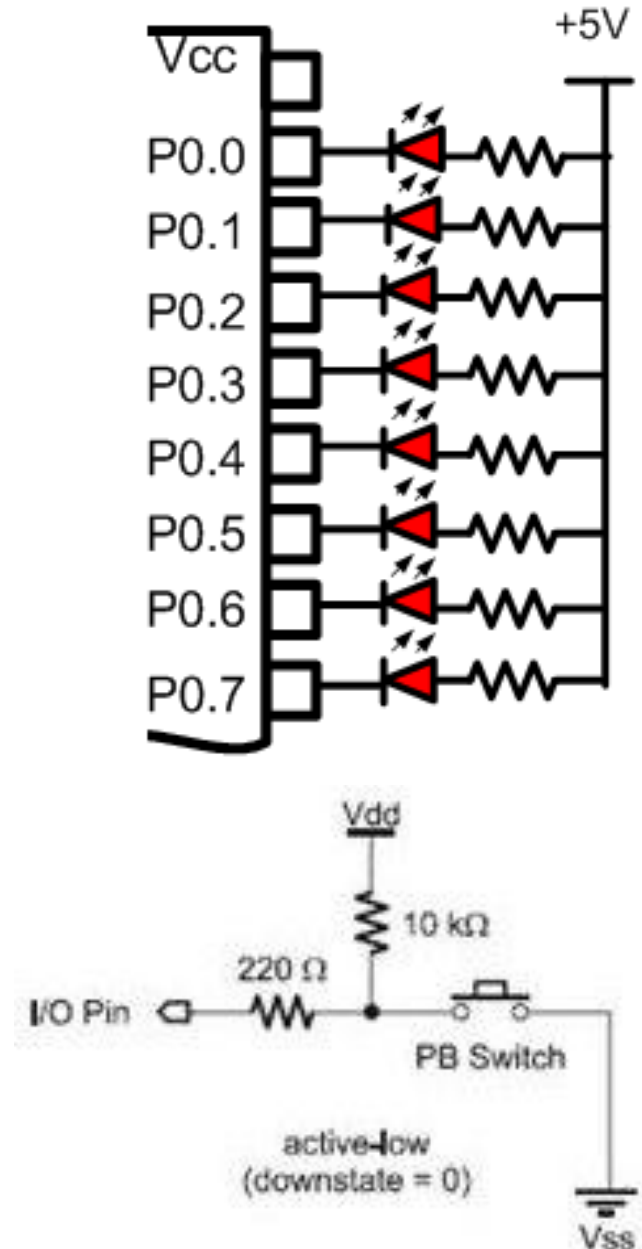
## 作業4-3

- 將LED連接到PORT2，當P2輸出0時，LED為亮；當P2輸出1時，LED為暗。
- 鍵盤：連接到P1<sup>0</sup>
- 功能：
  1. 鍵盤每按一次，LED依序0, 1, 2, 3, 4, 5, 6, 7，一次亮一個LED，其餘不亮；
  2. 到達7後，反覆執行1
  3. 使用軟體處理按鍵彈跳現象：  
當偵測到低電位時，延遲20ms，若還是低電位，表示按鍵動作否則為雜訊。



## 作業4-4

- 將LED連接到PORT2，當P2輸出0時，LED為亮；當P2輸出1時，LED為暗。
- 鍵盤：連接到P1<sup>0</sup>
- 功能：
  1. 鍵盤每按一次，LED依序0, 1, 2, 3, 4, 5, 6, 7, 間隔一秒一次亮一個；
  2. 到達7後，LED依序7, 6, 5, 4, 3, 2, 1, 0, 間隔一秒一次亮一個；
  3. 反覆執行1和2





~ END ~