

# 實驗六 中斷 ( Interrupt )

## 1. 學習重點

- 認識 8051 的中斷源以及與中斷相關之暫存器，並熟悉中斷流程。
- 分別用負緣觸發以及低準位觸發，兩種方式觸發中斷。

## 2. 材料清單

表 6-1、材料清單

器材名稱		數量
AT89S51		1
12MHz 石英震盪器		1
LED 二極體		8
按壓開關		3
電阻	1k $\Omega$	10
	10k $\Omega$	1
電容	20pF	2
	10 $\mu$ F	1

## 3. 元件原理

### 中斷 Interrupt 與輪詢 Polling

MCU 服務周邊裝置的方法有兩種。一種是中斷 ( Interrupt )，當中斷訊號產生時，主程式會暫停執行，並跳去執行中斷服務程式 ( ISR，Interrupt Service Routines )，若同時有很多中斷訊號產生，則會依照優先權決定順序；另一種是輪詢 ( Polling )，程式不斷詢問某個條件是否達成，當條件達成，就去執行使用者需求。

### 中斷之基本流程

中斷發生時 PC 值會被 push 進 stack 並會被設成相對應的中斷向量位址。當進入中斷服務程式後，依照應用情況，必要時會將主程式的資料 push 進 stack

( 例 如:PSW、ACC 等 )。若中斷服務程式中會使用到同樣的暫存器，也會在進入中斷服務程式後切換暫存器庫。結束中斷服務程式前，須將原本主程式的狀態從 stack 中 pop 出來。最後會執行 RETI 來結束中斷服務程，此時 return address 會被從 stack 中 pop 到 PC 並繼續執行中斷前的主程式。

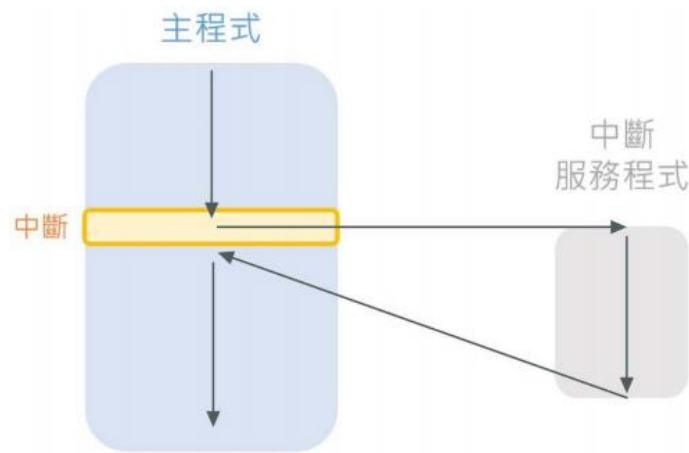


圖 6-1、中斷之基本流程

### 8051 的中斷源

8051 單晶片中提供 5 個中斷源，分別為：

- INTO：外部中斷，由 8051 單晶片 Pin 12 輸入。
- Timer0：計時/計數器中斷。
- INT1：外部中斷，由 8051 單晶片 Pin 13 輸入。
- Timer1：計時/計數器中斷。
- UART：串列埠中斷。

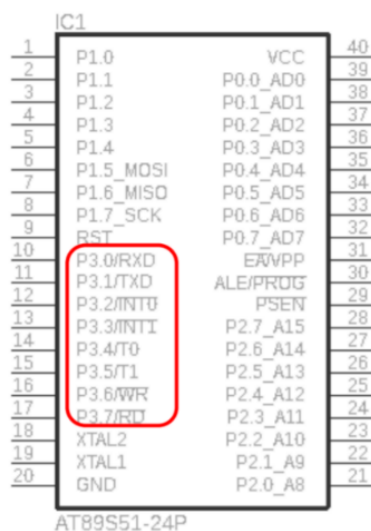


圖 6-2、8051 的中斷源

## 中斷向量

中斷源在 8051 中都有相對應的旗標，當中斷條件產生時，中斷源就會使其相對應的旗標值設定為 1。8051 的 CPU 會在每一個 machine cycle 檢查這些旗標的狀態，若系統允許相對的中斷源產生中斷，且該中斷相對應的旗標值亦為 1 時，則 CPU 會在執行完目前正在執行的指令後，將程式在記憶體中的位址存入堆疊中，並產生中斷服務副程式的呼叫，跳到該中斷所對應之中斷向量位址去執行，CPU 執行該中斷服務副程式，直到「RETI」指令後才結束中斷副程式，再從堆疊中取出先前存入的位址值繼續執行被中斷的程式。

表 6-2、8051 的中斷向量與中斷相關暫存器

中斷源	中斷向量	旗標	所屬暫存器
INT0	03H	IE0	TCON.1
Timer0	0BH	TF0	TCON.5
INT1	13H	IE1	TCON.3
Timer1	1BH	TF1	TCON.7
UART ( TXD )	23H	TI	SCON.1
UART ( RXD )	23H	RI	SCON.0

## 中斷致能暫存器 ( IE 暫存器 )

IE 暫存器為一個 8 位元的定址暫存器，其中各位元如下表格所示：

表 6-3、IE 暫存器

	7	6	5	4	3	2	1	0
IE	EA	-	ET2	ES	ET1	EX1	ET0	EX0

- IE.0：INT0 之中斷開關。  
EX0=1 時，啟用 INT0 中斷功能。  
EX0=0 時，停用 INT0 中斷功能。
- IE.1：TF0 之中斷開關。  
ET0=1 時，啟用 TF0 中斷功能。  
ET0=0 時，停用 TF0 中斷功能。
- IE.2：INT1 之中斷開關。  
EX1=1 時，啟用 INT1 中斷功能。  
EX1=0 時，停用 INT1 中斷功能。
- IE.3：TF1 之中斷開關。  
ET1=1 時，啟用 TF1 中斷功能。  
ET1=0 時，停用 TF1 中斷功能。
- IE.4：串列埠之中斷開關。  
ES=1 時，啟用串列埠中斷功能。  
ES=0 時，停用串列埠中斷功能。
- IE.5 ( 8052 才有 )：TF2 之中斷開關。  
ET2=1 時，啟用 TF2 中斷功能。  
ET2=0 時，停用 TF2 中斷功能。
- IE.6：保留。
- IE.7：中斷總開關。  
EA=1 時，啟用所有中斷功能。  
EA=0 時，停用所有中斷功能。

## 外部中斷

8051 的訊號的採樣方式可分為低準位觸發以及負緣觸發 ( 每一個 machine cycle 採樣一次 )。進入中斷服務程式後，IE0 ( TCON.1 ) 與 IE1 ( TCON.3 ) 的值會被 CPU 清除。

準位觸發 ( level-triggered )，又分為高準位觸發 ( high level-triggered ) 以及低準位觸發 ( low level-triggered )。

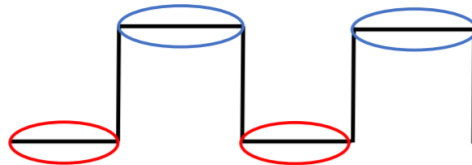


圖 6-3、level-triggered

邊緣觸發 ( edge-triggered )，又分為正緣觸發 ( rising edge-triggered ) 與負緣觸發 ( falling edge-triggered )。

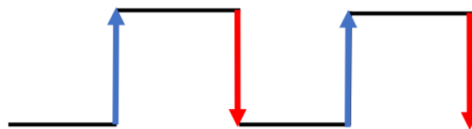


圖 6-4、edge-triggered

CPU 會透過 INT0、INT1 接腳來接受外部中斷需求。INT0 接腳 ( Pin12，與 Port3.2 共用 ) 接收到 low level 或 falling edge 訊號時，IE0 會被 CPU 自動設為 1，產生外部中斷；INT1 接腳 ( Pin13，與 Port3.3 共用 ) 接收到 low level 或 falling edge 訊號時，IE1 會被 CPU 自動設為 1，產生外部中斷。控制流程如下：

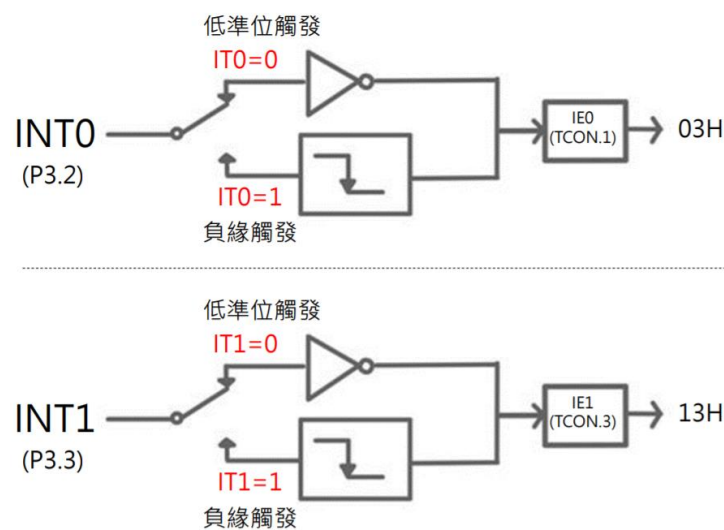


圖 6-5、外部中斷之控制流程

## 計時/計數器控制暫存器 ( TCON 暫存器 )

計時器控制為 CPU 將計數內部的時鐘脈波，提出中斷；計數器控制為 CPU 將計數外部的脈波，提出中斷。

TCON 暫存器為一個 8 位元的定址暫存器，其中各位元如下表格所示：

表 6-4、TCON 暫存器

	7	6	5	4	3	2	1	0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

- TCON.0：INT0 之信號種類。  
IT0=1 時，負緣觸發。  
IT0=0 時，低準位動作。
- TCON.1：INT0 之中斷旗標。  
INT0 中斷時，CPU 設定 IE0=1。  
INT0 中斷結束時，CPU 設定 IE0=0。
- TCON.2：INT1 之信號種類。  
IT1=1 時，負緣觸發。  
IT1=0 時，低準位動作。
- TCON.3：INT1 之中斷旗標。  
INT1 中斷時，CPU 設定 IE1=1。  
INT1 中斷結束時，CPU 設定 IE1=0。
- TCON.4 ~ 7：計時/計數器功能 ( 實驗七詳細說明 )。

### 串列埠中斷(UART)

CPU 透過 RXD、TXD 接腳接受或傳送中斷需求。RXD 接腳 ( Pin10，與 Port3.0 共用 )，其對應旗標為 RI ( Receive Interrupt )，接受中斷需求；TXD 接腳 ( Pin11，與 Port3.1 共用 )，其對應旗標為 TI ( Transfer Interrupt )，傳送中斷需求。

當其對應旗標 TI 或 RI 設定為 1 後，中斷服務程式將會執行。RXD 與 TXD 共用一個向量，ISR 必須自己判斷是 TI 或 RI 造成中斷。TI 變成 1 表示資料傳完；RI 變成 1 表示接收到資料，且 CPU 不會自動清除 TI 與 RI。



圖 6-6、UART 中斷

## 中斷優先等級暫存器 ( IP 暫存器 )

IP 暫存器為一個 8 位元的定址暫存器，其中各位元如下表格所示：

表 6-5、IP 暫存器

	7	6	5	4	3	2	1	0
IP	-	-	PT2	PS	PT1	PX1	PT0	PX0

- IP.0：INT0 之中斷優先等級。  
PX0=1 時，INT0 具高優先等級。  
PX0=0 時，INT0 不具高優先等級。
- IP.1：TF0 之中斷優先等級。  
PT0=1 時，TF0 具高優先等級。  
PT0=0 時，TF0 不具高優先等級。
- IP.2：INT1 之中斷優先等級。  
PX1=1 時，INT1 具高優先等級。  
PX1=0 時，INT1 不具高優先等級。
- IP.3：TF1 之中斷優先等級。  
PT1=1 時，TF1 具高優先等級。  
PT1=0 時，TF1 不具高優先等級。
- IP.4：串列埠之中斷優先等級。  
PS=1 時，串列埠具高優先等級。  
PS=0 時，串列埠不具高優先等級。
- IP.5 ( 8052 才有 )：TF2 之中斷優先等級。  
PT2=1 時，TF2 具高優先等級。  
PT2=0 時，TF2 不具高優先等級。
- IP.6、IP.7：保留。

Reset 後 IP 值均為 0，代表為相同的低層次優先權，若設定為 1 則會 變為高層次優先權。相同層次仍有預設優先順序：INT0>TF0>INT1>TF1>RI/TI。且 IP 暫存器為可位元定址 ( bit addressable )。

## 中斷的應用

### 設置中斷向量

8051 的中斷向量為上述表格之介紹，各中斷向量之間只有 8 個位址的記憶體空間，一般來說不會在這短短的 8 個位址中撰寫副程式，通常會以 JMP 之指令跳至特定的中斷副程式。

### 中斷的設定

- 開啟中斷開關 ( 即設定 IE 暫存器 )

範例：

要開啟總開關、INT0 開關可使用以下指令：

```
MOV IE, #10000001B
```

或使用以下指令：

```
SETB IE.7
```

```
SETB IE.0
```

若不清楚位置，也可使用直接指定開關名稱的方式使用以下指令：

```
SETB EA
```

```
SETB EX0
```

- 設定中斷優先等級 ( 即設定 IP 暫存器 )

範例：

要把 INT0 中斷的優先等級提高可使用以下指令設定中斷信號 ( 即設定 TCON 暫存器 )：

```
SETB PX1
```

- 設定中斷信號 ( 即設定 TCON 暫存器 )

範例：

要 INT0 採負緣觸發可使用以下指令：

```
SETB IT0
```



- 程式的預設堆疊指標是指向 07H 位置 ( 從 08H 開始存放資料 )，而 08H 為暫存器庫 1 ( 即 RB1 ) 的位置，為避免衝突，會將堆疊指標改至其他地方。

要把堆疊移到 30H 可使用以下指令：

跟其他副程式之差異為，其結尾要使用 *RETI*。

## 6. 軟體流程圖

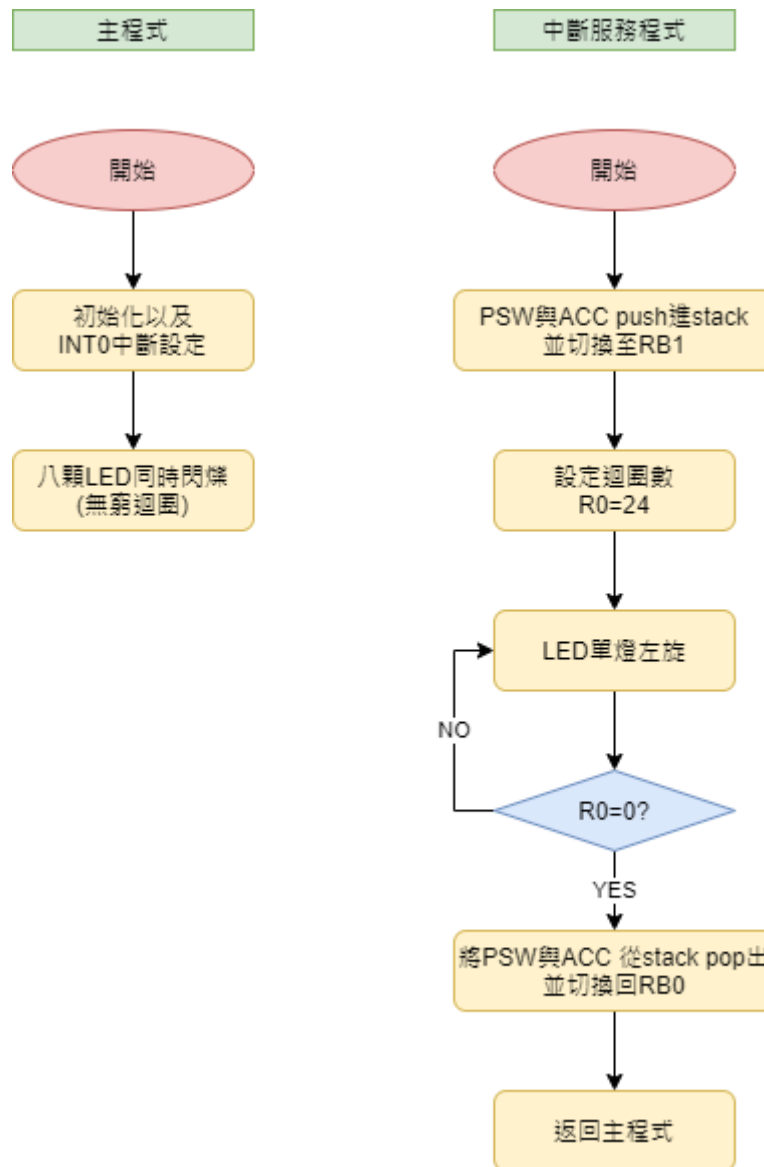


圖 6-8、實驗六基礎題參考軟體流程圖

## 7. 範例程式碼

```
1      ORG 00H                ; code start from 000h
2      SJMP MAIN              ; jump to MAIN
3      ORG 03H                ; vector address for INTO
4      SJMP INTO_ISR          ; jump to INTO_ISR
5      ORG 030H               ; after vector table space
6  MAIN:  MOV IE, #10000001B   ; enable EA and EX0
7         MOV SP, #30H        ; stack start from #30H
8         SETB ITO             ; falling edge-triggered
9         MOV A, #0000000B     ; set ACC as 0000000B
10  LOOP: MOV P2, A             ; P1 = A(LED output)
11         CALL DELAY          ; call delay function
12         CPL A               ; reverse A
13         SJMP LOOP           ; infinite loop
14  INTO_ISR: PUSH PSW          ; push PSW into stack
15         PUSH ACC            ; push ACC into stack
16         SETB RSO            ; switch to RB1
17         MOV A, #11111110B   ; set ACC as 11111110B
18         MOV R0, #24         ; loop counter = 24
19  ROTATE_L: MOV P2, A         ; P1 = A(LED output)
20         CALL DELAY          ; call delay function
21         RL A                ; rotate left
22         DJNZ R0, ROTATE_L    ; loop until R0 is 0
23         POP ACC             ; pop out ACC from stack
24         POP PSW             ; pop out PSW from stack
25         RETI                ; return from ISR
26  DELAY: MOV R7, #200
27  D1:    MOV R6, #250
28         DJNZ R6, $
29         DJNZ R7, D1
30         RET                 ; return
                END            ; end the code
```

## 8. 整理的題目，選擇/是非題